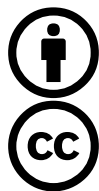


THE OPEN LOGIC TEXT

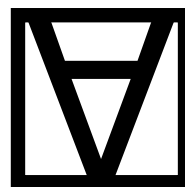
Debug Build (brazilian / pt)

Projeto Open Logic

Revision: 51c2271 (main)
2026-07-12



The Open Logic Text do
Projeto Open Logic está
licenciado sob a Licença
Creative Commons Atri-
buição 4.0 Internacional.



Sumário

I	Teoria Ingênua dos Conjuntos	20
1	Conjuntos	20
1.1	Extensionalidade	20
1.2	Subconjuntos e conjuntos potência	22
1.3	Alguns Conjuntos Importantes	23
1.4	Unões e Interseções	24
1.5	Pares, Tuplas, Produtos Cartesianos	27
1.6	Paradoxo de Russell	29
2	Relações	30
2.1	Relações como Conjuntos	30
2.2	Reflexões Filosóficas	32
2.3	Propriedades Especiais das Relações	33
2.4	Relações de Equivalência	34
2.5	Ordens	35
2.6	Grafos	38
2.7	Árvores	39
2.8	Operações sobre Relações	41
3	Funções	42
3.1	Noções Básicas	42
3.2	Tipos de Funções	44
3.3	Funções como Relações	46
3.4	Inversas de Funções	47
3.5	Composição de Funções	50
3.6	Funções Parciais	51
4	O Tamanho dos Conjuntos	52
4.1	Introdução	52
4.2	Enumerações e Conjuntos Enumeráveis	52
4.3	Método do Zigue-Zague de Cantor	57
4.4	Funções de Pareamento e Códigos	58
4.5	Uma Função de Emparelhamento Alternativa	60
4.6	Conjuntos Não enumeráveis	61
4.7	Redução	64

4.8	Equinumerosidade	66
4.9	Conjuntos de Tamanhos Diferentes e o Teorema de Cantor	68
4.10	A Noção de Tamanho e Schröder-Bernstein	70
4.11	Enumerações e Conjuntos Enumeráveis	70
4.12	Conjuntos Não Enumeráveis	72
4.13	Redução	74
5	Aritmetização	76
5.1	De $\mathbb{N}$ para $\mathbb{Z}$	76
5.2	De $\mathbb{Z}$ para $\mathbb{Q}$	78
5.3	A Reta Real	80
5.4	De $\mathbb{Q}$ para $\mathbb{R}$	81
5.5	Algumas Reflexões Filosóficas	83
5.6	Anéis e Corpos Ordenados	84
5.7	Apêndice: os Reais como Sequências de Cauchy	88
6	Conjuntos Infinitos	91
6.1	Hotel de Hilbert	92
6.2	Álgebras de Dedekind	92
6.3	Indução Aritmética	94
6.4	“Prova” de Dedekind	95
6.5	Apêndice: Provando Schröder-Bernstein	97
II	Propositional Logic	99
7	Syntax and Semantics	99
7.1	Introduction	99
7.2	Propositional Fórmulas	101
7.3	Preliminaries	102
7.4	Formation Sequences	104
7.5	Valorações and Satisfaction	106
7.6	Semantic Notions	108
8	Sistemas de Derivação	109
8.1	Introdução	109
8.2	O Cálculo de Sequentes	111
8.3	Dedução Natural	112
8.4	Tableaux	113
8.5	Dedução Axiomática	115
9	O Cálculo de Sequentes	116
9.1	Regras e Derivações	116
9.2	Regras Proposicionais	117
9.3	Regras Estruturais	118
9.4	Derivações	119

SUMÁRIO

9.5	Exemplos de Derivações	121
9.6	Noções da Teoria da Prova	125
9.7	Derivabilidade e Consistência	127
9.8	Derivabilidade e os Conectivos Proposicionais	129
9.9	Correção	130
10	Dedução Natural	134
10.1	Regras e Derivações	135
10.2	Regras Proposicionais	135
10.3	Derivações	137
10.4	Exemplos de Derivações	138
10.5	Noções da Teoria da Prova	143
10.6	Derivabilidade e Consistência	145
10.7	Derivabilidade e os Conectivos Proposicionais	146
10.8	Correção	148
11	Tableaux	151
11.1	Rules and Tableaux	152
11.2	Propositional Rules	152
11.3	Tableaux	154
11.4	Examples of Tableaux	155
11.5	Proof-Theoretic Notions	160
11.6	Derivabilidade and Consistency	162
11.7	Derivabilidade and the Propositional Connectives	164
11.8	Soundness	166
12	Axiomatic Derivações	169
12.1	Rules and Derivações	169
12.2	Axioms and Rules for the Propositional Connectives	171
12.3	Examples of Derivações	171
12.4	Proof-Theoretic Notions	173
12.5	The Deduction Theorem	175
12.6	Derivabilidade and Consistency	176
12.7	Derivabilidade and the Propositional Connectives	177
12.8	Soundness	178
13	O Teorema da Completude	179
13.1	Introdução	179
13.2	Esboço da Prova	180
13.3	Conjuntos Consistentes Completos de Sentenças	182
13.4	Lema de Lindenbaum	183
13.5	Construção de um Modelo	184
13.6	O Teorema da Completude	185
13.7	O Teorema da Compacidade	186
13.8	Uma Prova Direta do Teorema da Compacidade	187

III Lógica de Primeira Ordem	189
14 Introdução à Lógica de Primeira Ordem	189
14.1 Lógica de Primeira Ordem	189
14.2 Sintaxe	191
14.3 Fórmulas	192
14.4 Satisfação	193
14.5 Sentenças	194
14.6 Noções Semânticas	195
14.7 Substituição	196
14.8 Modelos e Teorias	197
14.9 Correção e Completude	198
15 Sintaxe da Lógica de Primeira Ordem	199
15.1 Introdução	199
15.2 Linguagens de Primeira Ordem	200
15.3 Termos e Fórmulas	202
15.4 Leitura Única	205
15.5 Operador principal de uma Fórmula	207
15.6 Subfórmulas	208
15.7 Sequências de Formação	210
15.8 Variáveis Livres e Sentenças	214
15.9 Substituição	215
16 Semântica da Lógica de Primeira Ordem	217
16.1 Introdução	217
16.2 Estruturas para Linguagens de Primeira Ordem	218
16.3 Estruturas Cobertas para Linguagens de Primeira Ordem	219
16.4 Satisfação de a Fórmula em a Estrutura	220
16.5 Atribuições de Variáveis	225
16.6 Extensionalidade	230
16.7 Noções Semânticas	231
17 Teorias e Seus Modelos	233
17.1 Introdução	233
17.2 Expressando Propriedades de Estruturas	235
17.3 Exemplos de Teorias de Primeira Ordem	236
17.4 Expressando Relações em a Estrutura	239
17.5 A Teoria dos Conjuntos	240
17.6 Expressando o Tamanho de Estruturas	243
18 Sistemas de Derivação	244
18.1 Introdução	245
18.2 O Cálculo de Sequentes	246
18.3 Dedução Natural	247
18.4 Tableaux	248

SUMÁRIO

18.5	Dedução Axiomática	250
19	O Cálculo de Sequentes	251
19.1	Regras e Derivações	252
19.2	Regras Proposicionais	253
19.3	Regras de Quantificadores	253
19.4	Regras Estruturais	255
19.5	Derivações	256
19.6	Exemplos de Derivações	257
19.7	Derivações com Quantificadores	262
19.8	Noções da Teoria da Prova	263
19.9	Derivabilidade e Consistência	265
19.10	Derivabilidade e os Conectivos Proposicionais	267
19.11	Derivabilidade e os Quantificadores	268
19.12	Correção	269
19.13	Derivações com Predicado de identidade	274
19.14	Correção com Predicado de identidade	275
20	Dedução Natural	275
20.1	Regras e Derivações	276
20.2	Regras Proposicionais	276
20.3	Regras de Quantificadores	278
20.4	Derivações	279
20.5	Exemplos de Derivações	281
20.6	Derivações com Quantificadores	285
20.7	Noções da Teoria da Prova	289
20.8	Derivabilidade e Consistência	291
20.9	Derivabilidade e os Conectivos Proposicionais	293
20.10	Derivabilidade e os Quantificadores	294
20.11	Correção	295
20.12	Derivações com Predicado de identidade	299
20.13	Correção com Predicado de identidade	301
21	Tableaux	301
21.1	Rules and Tableaux	302
21.2	Propositional Rules	302
21.3	Quantifier Rules	303
21.4	Tableaux	305
21.5	Examples of Tableaux	306
21.6	Tableaux with Quantifiers	311
21.7	Proof-Theoretic Notions	315
21.8	Derivabilidade and Consistency	317
21.9	Derivabilidade and the Propositional Connectives	319
21.10	Derivabilidade and the Quantifiers	321
21.11	Soundness	322
21.12	Tableaux with Predicado de identidade	325

21.13	Soundness with Predicado de identidade	326
22	Axiomatic Derivações	327
22.1	Rules and Derivações	327
22.2	Axioms and Rules for the Propositional Connectives	329
22.3	Axioms and Rules for Quantifiers	329
22.4	Examples of Derivações	330
22.5	Derivações with Quantifiers	332
22.6	Proof-Theoretic Notions	333
22.7	The Deduction Theorem	334
22.8	The Deduction Theorem with Quantifiers	336
22.9	Derivabilidade and Consistency	337
22.10	Derivabilidade and the Propositional Connectives	338
22.11	Derivabilidade and the Quantifiers	339
22.12	Soundness	339
22.13	Derivações with Predicado de identidade	341
23	O Teorema da Completude	341
23.1	Introdução	342
23.2	Esboço da Prova	343
23.3	Conjuntos Consistentes Completos de Sentenças	345
23.4	Expansão de Henkin	347
23.5	Lema de Lindenbaum	349
23.6	Construção de um Modelo	350
23.7	Identidade	353
23.8	O Teorema da Completude	355
23.9	O Teorema da Compacidade	356
23.10	Uma Prova Direta do Teorema da Compacidade	358
23.11	O Teorema de Löwenheim–Skolem	360
24	Além da lógica de primeira ordem	361
24.1	Visão Geral	361
24.2	Lógica de Múltiplas Ordenações	362
24.3	Lógica de Segunda Ordem	363
24.4	Lógica de Ordem Superior	368
24.5	Lógica Intuicionista	370
24.6	Lógicas Modais	374
24.7	Outras Lógicas	375
IV	Model Theory	377
25	Basics of Model Theory	377
25.1	Reducts and Expansions	377
25.2	Subestruturas	378
25.3	Overspill	379

SUMÁRIO

25.4	Isomorphic Structures	379
25.5	The Theory of a Estrutura	381
25.6	Partial Isomorphisms	382
25.7	Dense Linear Orders	384
26	Models of Arithmetic	386
26.1	Introduction	386
26.2	Standard Models of Arithmetic	387
26.3	Non-Standard Models	389
26.4	Models of Q	391
26.5	Models of PA	393
26.6	Computable Models of Arithmetic	396
27	The Interpolation Theorem	398
27.1	Introduction	398
27.2	Separation of Sentenças	398
27.3	Craig's Interpolation Theorem	400
27.4	The Definability Theorem	402
28	Lindström's Theorem	404
28.1	Introduction	404
28.2	Abstract Logics	405
28.3	Compactness and Löwenheim–Skolem Properties	407
28.4	Lindström's Theorem	408
V	Computability	410
29	Recursive Functions	410
29.1	Introduction	410
29.2	Primitive Recursion	411
29.3	Composition	413
29.4	Primitive Recursion Functions	415
29.5	Primitive Recursion Notations	417
29.6	Primitive Recursive Functions are Computable	418
29.7	Examples of Primitive Recursive Functions	419
29.8	Primitive Recursive Relations	422
29.9	Bounded Minimization	424
29.10	Primes	425
29.11	Sequences	426
29.12	Trees	429
29.13	Other Recursions	430
29.14	Non-Primitive Recursive Functions	431
29.15	Partial Recursive Functions	432
29.16	The Normal Form Theorem	434
29.17	The Halting Problem	435

29.18	General Recursive Functions	436
30	Computability Theory	437
30.1	Introduction	437
30.2	Coding Computations	438
30.3	The Normal Form Theorem	439
30.4	The s - m - n Theorem	440
30.5	The Universal Partial Computable Function	441
30.6	No Universal Computable Function	442
30.7	The Halting Problem	442
30.8	Comparison with Russell's Paradox	444
30.9	Computable Sets	445
30.10	Computably Enumerable Sets	445
30.11	Definitions of C. E. Sets	446
30.12	There Are Non-Computable Sets	448
30.13	Union and Intersection of C.E. Sets	449
30.14	Computably Enumerable Sets not Closed under Complement	450
30.15	Reducibility	451
30.16	Properties of Reducibility	452
30.17	Complete Computably Enumerable Sets	454
30.18	An Example of Reducibility	455
30.19	Totality is Undecidable	456
30.20	Rice's Theorem	456
30.21	The Fixed-Point Theorem	458
30.22	Applying the Fixed-Point Theorem	462
30.23	Defining Functions using Self-Reference	463
VI	Máquinas de Turing	464
31	Computações de Máquinas de Turing	464
31.1	Introdução	464
31.2	Representando Máquinas de Turing	466
31.3	Máquinas de Turing	470
31.4	Configurações e Computações	471
31.5	Representação Unária de Números	473
31.6	Estados de Parada	476
31.7	Máquinas Disciplinadas	478
31.8	Combinando Máquinas de Turing	479
31.9	Variantes de Máquinas de Turing	481
31.10	A Tese de Church–Turing	483
32	Indecidibilidade	484
32.1	Introdução	485
32.2	Enumerando Máquinas de Turing	486
32.3	Máquinas de Turing Universais	489

SUMÁRIO

32.4	O Problema da Parada	491
32.5	O Problema da Decisão	493
32.6	Representando Máquinas de Turing	494
32.7	Verificando a Representação	497
32.8	O Problema da Decisão é Insolúvel	502
32.9	Teorema de Trakhtenbrot	503
VII Incompleteness		507
33	Introduction to Incompleteness	507
33.1	Historical Background	507
33.2	Definitions	511
33.3	Overview of Incompleteness Results	516
33.4	Undecidability and Incompleteness	517
34	Arithmetization of Syntax	519
34.1	Introduction	519
34.2	Coding Symbols	521
34.3	Coding Terms	522
34.4	Coding Fórmulas	524
34.5	Substitution	525
34.6	Derivações in LK	526
34.7	Derivações in Natural Deduction	529
34.8	Axiomatic Derivações	534
35	Representability in $\mathbf{Q}$	538
35.1	Introduction	538
35.2	Functions Representable in $\mathbf{Q}$ are Computable	540
35.3	The Beta Function Lemma	541
35.4	Simulating Primitive Recursion	544
35.5	Basic Functions are Representable in $\mathbf{Q}$	545
35.6	Composition is Representable in $\mathbf{Q}$	548
35.7	Regular Minimization is Representable in $\mathbf{Q}$	549
35.8	Computable Functions are Representable in $\mathbf{Q}$	552
35.9	Representing Relations	553
35.10	Undecidability	554
35.11	Σ_1 completeness	555
36	Theories and Computability	558
36.1	Introduction	558
36.2	$\mathbf{Q}$ is C.e.-Complete	559
36.3	ω -Consistent Extensions of $\mathbf{Q}$ are Undecidable	560
36.4	Consistent Extensions of $\mathbf{Q}$ are Undecidable	560
36.5	Axiomatizável Theories	562
36.6	Axiomatizável Complete Theories are Decidable	562

36.7	Q has no Complete, Consistent, Axiomatizável Extensions . . .	563
36.8	Sentences Provable and Refutable in Q are Computably Inseparable	563
36.9	Theories Consistent with Q are Undecidable	564
36.10	Theories in which Q is Interpretable are Undecidable	565
37	Incompleteness and Provability	566
37.1	Introduction	566
37.2	The Fixed-Point Lemma	568
37.3	The First Incompleteness Theorem	570
37.4	Rosser's Theorem	571
37.5	Comparison with Gödel's Original Paper	573
37.6	The Derivabilidade Conditions for PA	574
37.7	The Second Incompleteness Theorem	575
37.8	Löb's Theorem	577
37.9	The Undefinability of Truth	580
VIII	Second-order Logic	582
38	Syntax and Semantics	582
38.1	Introduction	582
38.2	Terms and Fórmulas	583
38.3	Satisfaction	584
38.4	Semantic Notions	587
38.5	Expressive Power	587
38.6	Describing Infinite and Enumerável Domínios	589
39	Metatheory of Second-order Logic	590
39.1	Introduction	591
39.2	Second-order Arithmetic	591
39.3	Second-order Logic is not Axiomatizable	593
39.4	Second-order Logic is not Compact	594
39.5	The Löwenheim–Skolem Theorem Fails for Second-order Logic	595
40	Second-order Logic and Set Theory	595
40.1	Introduction	596
40.2	Comparing Sets	596
40.3	Cardinalities of Sets	597
40.4	The Power of the Continuum	598
IX	The Lambda Calculus	601
41	Introduction	601
41.1	Overview	601
41.2	The Syntax of the Lambda Calculus	602

SUMÁRIO

41.3	Reduction of Lambda Terms	604
41.4	The Church–Rosser Property	605
41.5	Currying	605
41.6	λ -definível Arithmetical Functions	606
41.7	λ -definível Functions are Computable	607
41.8	Computable Functions are λ -definível	607
41.9	The Basic Primitive Recursive Functions are λ -definível	608
41.10	The λ -definível Functions are Closed under Composition	608
41.11	λ -definível Functions are Closed under Primitive Recursion	609
41.12	Fixed-Point Combinators	611
41.13	The λ -definível Functions are Closed under Minimization	612
42	Syntax	613
42.1	Terms	613
42.2	Unique Readability	614
42.3	Abbreviated Syntax	615
42.4	Free Variables	616
42.5	Substitution	617
42.6	α -Conversion	620
42.7	The De Bruijn Index	624
42.8	Terms as α -Equivalence Classes	625
42.9	β -reduction	626
42.10	η -conversion	628
43	The Church–Rosser Property	629
43.1	Definition and Properties	630
43.2	Parallel β -reduction	631
43.3	β -reduction	633
43.4	Parallel $\beta\eta$ -reduction	634
43.5	$\beta\eta$ -reduction	636
44	Lambda Definability	637
44.1	Introduction	637
44.2	λ -definível Arithmetical Functions	638
44.3	Pairs and Predecessor	641
44.4	Truth Values and Relations	641
44.5	Primitive Recursive Functions are λ -definível	643
44.6	Fixpoints	645
44.7	Minimization	648
44.8	Partial Recursive Functions are λ -definível	649
44.9	λ -definível Functions are Recursive	649
X	Many-valued Logic	651
45	Syntax and Semantics	651

45.1	Introduction	651
45.2	Languages and Connectives	652
45.3	Fórmulas	653
45.4	Matrices	653
45.5	Valorações and Satisfaction	654
45.6	Semantic Notions	655
45.7	Many-valued logics as sublogics of C	656
46	Three-valued Logics	657
46.1	Introduction	657
46.2	Lukasiewicz logic	657
46.3	Kleene logics	661
46.4	Gödel logics	663
46.5	Designating not just T	665
47	Infinite-valued Logics	668
47.1	Introduction	668
47.2	Lukasiewicz logic	669
47.3	Gödel logics	670
48	Sequent Calculus	671
48.1	Introduction	672
48.2	Rules and Derivações	673
48.3	Structural Rules	674
48.4	Propositional Rules for Selected Logics	674
XI	Normal Modal Logics	678
49	Syntax and Semantics	678
49.1	Introduction	678
49.2	The Language of Basic Modal Logic	680
49.3	Simultaneous Substitution	681
49.4	Relational Models	682
49.5	Truth at a World	683
49.6	Truth in a Model	685
49.7	Validity	686
49.8	Tautological Instances	687
49.9	Schemas and Validity	689
49.10	Entailment	691
50	Frame Definability	693
50.1	Introduction	693
50.2	Properties of Accessibility Relations	694
50.3	Frames	696
50.4	Frame Definability	697

SUMÁRIO

50.5	First-order Definability	699
50.6	Equivalence Relations and S5	700
50.7	Second-order Definability	702
51	Axiomatic Derivações	705
51.1	Introduction	705
51.2	Normal Modal Logics	706
51.3	Derivações and Modal Systems	708
51.4	Proofs in K	709
51.5	Derived Rules	711
51.6	More Proofs in K	713
51.7	Dual Fórmulas	715
51.8	Proofs in Modal Systems	715
51.9	Soundness	717
51.10	Showing Systems are Distinct	717
51.11	Derivabilidade from a Set of Fórmulas	719
51.12	Properties of Derivabilidade	720
51.13	Consistency	720
52	Completeness and Canonical Models	721
52.1	Introduction	721
52.2	Complete Σ -Consistent Sets	722
52.3	Lindenbaum's Lemma	724
52.4	Modalities and Complete Consistent Sets	725
52.5	Canonical Models	727
52.6	The Truth Lemma	728
52.7	Determination and Completeness for K	729
52.8	Frame Completeness	730
53	Filtrations and Decidability	733
53.1	Introduction	733
53.2	Preliminaries	735
53.3	Filtrations	736
53.4	Examples of Filtrations	738
53.5	Filtrations are Finite	740
53.6	K and S5 have the Finite Model Property	741
53.7	S5 is Decidable	742
53.8	Filtrations and Properties of Accessibility	742
53.9	Filtrations of Euclidean Models	744
54	Modal Tableaux	746
54.1	Introduction	746
54.2	Rules for K	747
54.3	Tableaux for K	749
54.4	Soundness for K	750
54.5	Rules for Other Accessibility Relations	753

54.6	Soundness for Additional Rules	755
54.7	Simple Tableaux for S5	757
54.8	Completeness for K	758
54.9	Countermodels from Tableaux	761
55	Modal Sequent Calculus	763
55.1	Introduction	763
55.2	Rules for K	763
55.3	Sequent Derivações for K	764
55.4	Rules for Other Accessibility Relations	766
XII	Applied Modal Logic	768
56	Temporal Logics	768
56.1	Introduction	768
56.2	Semantics for Temporal Logic	769
56.3	Properties of Temporal Frames	771
56.4	Additional Operators for Temporal Logic	772
56.5	Possible Histories	772
57	Epistemic Logics	774
57.1	Introduction	774
57.2	The Language of Epistemic Logic	775
57.3	Relational Models	776
57.4	Truth at a World	777
57.5	Accessibility Relations and Epistemic Principles	779
57.6	Bisimulations	780
57.7	Public Announcement Logic	781
57.8	Semantics of Public Announcement Logic	783
XIII	Intuitionistic Logic	785
58	Introduction	785
58.1	Constructive Reasoning	785
58.2	Syntax of Intuitionistic Logic	786
58.3	The Brouwer–Heyting–Kolmogorov Interpretation	788
58.4	Natural Deduction	790
58.5	Axiomatic Derivações	793
59	Semantics	795
59.1	Introduction	795
59.2	Modelos relacionais	796
59.3	Semantic Notions	797
59.4	Topological Semantics	798

SUMÁRIO

60	Soundness and Completeness	800
60.1	Soundness of Axiomatic Derivações	800
60.2	Soundness of Natural Deduction	801
60.3	Lindenbaum’s Lemma	803
60.4	The Canonical Model	805
60.5	The Truth Lemma	806
60.6	The Completeness Theorem	806
60.7	Decidability	807
61	Intuitionistic Tableaux	808
61.1	Introduction	808
61.2	Rules for Intuitionistic Logic	809
61.3	Tableaux for Intuitionistic Logic	810
61.4	Soundness for Intuitionistic Tableaux	812
62	Propositions as Types	814
62.1	Introduction	815
62.2	Sequent Natural Deduction	816
62.3	Proof Terms	818
62.4	Converting Derivações to Proof Terms	819
62.5	Recovering Derivações from Proof Terms	822
62.6	Reduction	824
62.7	Normalization	826
XIV	Counterfactuals	829
63	Introduction	829
63.1	The Material Conditional	829
63.2	Paradoxes of the Material Conditional	831
63.3	The Strict Conditional	831
63.4	Counterfactuals	834
64	Minimal Change Semantics	835
64.1	Introduction	835
64.2	Sphere Models	836
64.3	Truth and Falsity of Counterfactuals	838
64.4	Antecedent Strengthening	840
64.5	Transitivity	841
64.6	Contraposition	842
XV	Set Theory	844
65	A Conceção Iterativa	844
65.1	Extensionalidade	844
65.2	Paradoxo de Russell (de novo)	845

65.3	Predicativo e Impredicativo	846
65.4	A Abordagem Cumulativa-Iterativa	848
65.5	Urelementos ou Não?	849
65.6	Apêndice: Lei Básica V de Frege	850
66	Passos rumo a $\mathbf{Z}$	851
66.1	A história com mais pormenor	852
66.2	Separação	852
66.3	União	854
66.4	Pares	854
66.5	Conjuntos potência	855
66.6	Infinito	856
66.7	$\mathbf{Z}^-$: um marco	858
66.8	A escolher os números naturais	859
66.9	Apêndice: Fechamento, compreensão e interseção	860
67	Ordinais	861
67.1	Introdução	861
67.2	A Ideia Geral de um Ordinal	861
67.3	Boas Ordenações	862
67.4	Isomorfismos de Ordem	863
67.5	Construção de Von Neumann	865
67.6	Propriedades Básicas dos Ordinais	866
67.7	Substituição	869
67.8	$\mathbf{ZF}^-$: um marco	870
67.9	Ordinais como Tipos de Ordem	870
67.10	Ordinais Sucessores e Limites	871
68	Estágios e postos	873
68.1	Definindo os estágios como os V_α	873
68.2	Teorema(s) de recursão transfinita	874
68.3	Propriedades básicas dos estágios	876
68.4	Fundação	877
68.5	$\mathbf{Z}$ e $\mathbf{ZF}$: um marco	879
68.6	Posto	879
69	Substituição	881
69.1	Introdução	881
69.2	A Força da Substituição	881
69.3	Considerações Extrínsecas	882
69.4	Limitação de Tamanho	884
69.5	Substituição e “Infinito Absoluto”	885
69.6	Substituição e Reflexão	887
69.7	Apêndice: Resultados sobre Substituição	888
69.8	Apêndice: Axiomatizabilidade finita	891

SUMÁRIO

70	Aritmética Ordinal	893
70.1	Introdução	893
70.2	Adição Ordinal	893
70.3	Usando Adição Ordinal	896
70.4	Multiplicação Ordinal	898
70.5	Exponenciação Ordinal	899
71	Cardinais	900
71.1	Princípio de Cantor	900
71.2	Cardinais como Ordinais	901
71.3	ZFC : um marco	903
71.4	Finito, Enumerável, Não Enumerável	903
71.5	Apêndice: Princípio de Hume	905
72	Aritmética Cardinal	907
72.1	Definindo as Operações Básicas	907
72.2	Simplificando Adição e Multiplicação	909
72.3	Algumas Simplificações	911
72.4	A Hipótese do Contínuo	912
72.5	Pontos Fixos de $\aleph$	914
73	Escolha	916
73.1	Introdução	916
73.2	O Truque de Tarski–Scott	917
73.3	Comparabilidade e Lema de Hartogs	918
73.4	O Problema da Boa Ordenação	919
73.5	Escolha Enumerável	920
73.6	Considerações Intrínsecas sobre a Escolha	923
73.7	Paradoxo de Banach–Tarski	924
73.8	Apêndice: Paradoxo de Vitali	925
XVI	Methods	930
74	Provas	930
74.1	Introdução	930
74.2	Começando uma Prova	931
74.3	Usando Definições	932
74.4	Padrões de Inferência	934
74.5	Um Exemplo	940
74.6	Outro Exemplo	943
74.7	Prova por Contradição	944
74.8	Lendo Provas	948
74.9	Não Consigo Fazer Isso!	949
74.10	Outros Recursos	950

75	Indução	951
75.1	Introdução	951
75.2	Indução em $\mathbb{N}$	952
75.3	Indução Forte	955
75.4	Definições Indutivas	955
75.5	Indução Estrutural	958
75.6	Relações e Funções	960
XVIII	History	963
76	Biografias	963
76.1	Georg Cantor	963
76.2	Alonzo Church	964
76.3	Gerhard Gentzen	965
76.4	Kurt Gödel	966
76.5	Emmy Noether	968
76.6	Rózsa Péter	969
76.7	Julia Robinson	970
76.8	Bertrand Russell	972
76.9	Alfred Tarski	973
76.10	Alan Turing	974
76.11	Ernst Zermelo	976
77	História e Mitologia da Teoria dos Conjuntos	977
77.1	Infinitesimais e Diferenciação	977
77.2	Definição Rigorosa de Limites	979
77.3	Patologias	981
77.4	Mais Mito do que História?	982
77.5	Cantor na Reta e no Plano	983
77.6	Apêndice: Curvas Preenchedoras de Espaço de Hilbert	984
XVIII	Reference	987
78	O Alfabeto Grego	987
79	The Fraktur Alphabet	988
	Créditos das Imagens	988
	Referências Bibliográficas	989

SUMÁRIO

This file loads all content included in the Open Logic Project. Editorial notes like this, if displayed, indicate that the file was compiled without any thought to how this material will be presented. If you can read this, it is probably *not advisable* to teach or study from this PDF.

The Open Logic Project provides many mechanisms by which a text can be generate which is more appropriate for teaching or self-study. For instance, by default, the text will make all logical operators primitives and carry out all cases for all operators in proofs. But it is much better to leave some of these cases as exercises. The Open Logic Project is also a work in progress. In an effort to stimulate collaboration and improvement, material is included even if it is only in draft form, is missing exercises, etc. A PDF produced for a course will exclude these sections.

To find PDFs more suitable for teaching and studying, have a look at the [sample courses available on the OLP website](#). To make your own, you might start from the [sample driver file](#) or look at the sources of the derived textbooks for more fancy and advanced examples.

Parte I

Teoria Ingênua dos Conjuntos

The material in this part is an introduction to basic naive set theory. With the inclusion of Tim Button's Teoria dos Conjuntos Aberta, this also covers the construction of number systems, and discussion of infinity, which are not required for the logical parts of the OLP.

Capítulo 1

Conjuntos

`content/sets-functions-relations/sets/basics.tex`

1.1 Extensionalidade

Um *conjunto* é uma coleção de objetos, considerada como um único objeto. Os objetos que compõem o conjunto chamam-se *elementos* ou *membros* do conjunto. Se x é um membro de um conjunto A , escrevemos $x \in A$; caso contrário, escrevemos $x \notin A$. O conjunto que não tem membros chama-se o *conjunto vazio* e denota-se “ $\emptyset$ ”.

sfr:set:bas:
sec

explanation

Não importa como *especificamos* o conjunto, ou como *ordenamos* os seus membros, ou mesmo quantas *vezes* contamos os seus membros. Só importa quais são os seus membros. Codificamos isto no princípio seguinte.

Definição 1.1 (Extensionalidade). Se A e B são conjuntos, então $A = B$ se e só se todo membro de A é também um membro de B , e vice-versa.

A extensionalidade autoriza certa notação. Em geral, quando temos objetos $a_1, \dots, a_n$, então $\{a_1, \dots, a_n\}$ é o conjunto cujos membros são $a_1, \dots, a_n$. Enfatizamos a palavra “o”, pois a extensionalidade nos diz que só pode haver

1.1. EXTENSIONALIDADE

um tal conjunto. De fato, a extensionalidade também autoriza o seguinte:

$$\{a, a, b\} = \{a, b\} = \{b, a\}.$$

Isto traduz a ideia de que, quando consideramos conjuntos, não nos importa a ordem dos seus **membros**, nem quantas vezes são especificados.

Exemplo 1.2. Sempre que temos vários objetos, podemos reuni-los num conjunto. O conjunto dos irmãos de Richard, por exemplo, é um conjunto que contém uma pessoa, e podemos escrevê-lo como $S = \{\text{Ruth}\}$. O conjunto dos inteiros positivos menores que 4 é $\{1, 2, 3\}$, mas também se pode escrever como $\{3, 2, 1\}$ ou mesmo como $\{1, 2, 1, 2, 3\}$. São todos o mesmo conjunto, pela extensionalidade. De fato, todo **membro** de $\{1, 2, 3\}$ é também **um membro** de $\{3, 2, 1\}$ (e de $\{1, 2, 1, 2, 3\}$), e vice-versa.

Com frequência especificaremos um conjunto por alguma propriedade que os seus **membros** compartilham. Usaremos a seguinte notação abreviada para isso: $\{x : \varphi(x)\}$, onde $\varphi(x)$ representa a propriedade que x tem de satisfazer para ser contado entre os **membros** do conjunto.

Exemplo 1.3. No nosso exemplo, também poderíamos ter especificado S como

$$S = \{x : x \text{ é irmão de Richard}\}.$$

Exemplo 1.4. Diz-se que um número é *perfeito* se e só se for igual à soma dos seus divisores próprios (isto é, números que o dividem sem resto mas não são iguais ao próprio número). Por exemplo, 6 é perfeito porque os seus divisores próprios são 1, 2, e 3, e $6 = 1 + 2 + 3$. De fato, 6 é o único inteiro positivo menor que 10 que é perfeito. Assim, usando a extensionalidade, podemos dizer:

$$\{6\} = \{x : x \text{ é perfeito e } 0 \leq x \leq 10\}$$

Lemos a notação à direita como “o conjunto dos x tais que x é perfeito e $0 \leq x \leq 10$ ”. A identidade aqui confirma que, quando consideramos conjuntos, não nos importa como são especificados. E, mais em geral, a extensionalidade garante que há sempre apenas um conjunto dos x tais que $\varphi(x)$. Logo, a extensionalidade justifica chamar a $\{x : \varphi(x)\}$ o conjunto dos x tais que $\varphi(x)$.

A extensionalidade nos dá um modo de mostrar que conjuntos são idênticos: para mostrar que $A = B$, mostre que sempre que $x \in A$ então também $x \in B$, e sempre que $y \in B$ então também $y \in A$.

Problema 1.1. Prove que há no máximo um conjunto vazio, isto é, mostre que se A e B são conjuntos sem **membros**, então $A = B$.

1.2 Subconjuntos e conjuntos potência

explanation Com frequência queremos comparar conjuntos. É um tipo óbvio de comparação que se pode fazer é o seguinte: *tudo o que está num conjunto está também no outro*. Esta situação é suficientemente importante para introduzirmos nova notação. sfr:set:sub:sec

Definição 1.5 (Subconjunto). Se todo **membro** de um conjunto A é também **um membro** de B , então dizemos que A é um *subconjunto* de B , e escrevemos $A \subseteq B$. Se A não é subconjunto de B , escrevemos $A \not\subseteq B$. Se $A \subseteq B$ mas $A \neq B$, escrevemos $A \subsetneq B$ e dizemos que A é um *subconjunto próprio* de B .

Exemplo 1.6. Todo conjunto é subconjunto de si próprio, e $\emptyset$ é subconjunto de todo conjunto. O conjunto dos números naturais pares é subconjunto do conjunto dos números naturais. Também $\{a, b\} \subseteq \{a, b, c\}$. Mas $\{a, b, e\}$ não é subconjunto de $\{a, b, c\}$.

Exemplo 1.7. O número 2 é um **membro** do conjunto dos inteiros, ao passo que o conjunto dos números pares é subconjunto do conjunto dos inteiros. Contudo, um conjunto pode tanto ser **um membro** quanto subconjunto de algum outro conjunto, por exemplo, $\{0\} \in \{0, \{0\}\}$ e também $\{0\} \subseteq \{0, \{0\}\}$.

A extensionalidade dá um critério de identidade para conjuntos: $A = B$ se e só se todo **membro** de A é também **um membro** de B e vice-versa. A definição de “subconjunto” define $A \subseteq B$ precisamente como a primeira metade deste critério: todo **membro** de A é também **um membro** de B . Claro que a definição também se aplica se trocarmos A e B : isto é, $B \subseteq A$ se e só se todo **membro** de B é também **um membro** de A . E isso, por sua vez, é exatamente a parte “vice-versa” da extensionalidade. Em outras palavras, a extensionalidade implica que conjuntos são iguais se e só se são subconjuntos um do outro.

Proposição 1.8. $A = B$ se e só se $A \subseteq B$ e $B \subseteq A$.

Agora é também boa ocasião para introduzir mais alguma notação útil. Ao definir quando A é subconjunto de B dissemos que “todo **membro** de A é ...”, e preenchamos os “...” com “**um membro** de B ”. Mas esta é uma *forma* tão comum de expressão que convém introduzir notação formal para ela.

Definição 1.9. $(\forall x \in A)\varphi$ abrevia $\forall x(x \in A \rightarrow \varphi)$. Do mesmo modo, $(\exists x \in A)\varphi$ abrevia $\exists x(x \in A \wedge \varphi)$. sfr:set:sub:forallxina

Com esta notação, podemos dizer que $A \subseteq B$ se e só se $(\forall x \in A)x \in B$.

Passamos agora a considerar um certo tipo de conjunto: o conjunto de todos os subconjuntos de um dado conjunto.

Definição 1.10 (Conjunto potência). O conjunto constituído por todos os subconjuntos de um conjunto A chama-se o *conjunto potência* de A , e denota-se $\wp(A)$.

$$\wp(A) = \{B : B \subseteq A\}$$

1.3. ALGUNS CONJUNTOS IMPORTANTES

Exemplo 1.11. Quais são todos os subconjuntos possíveis de $\{a, b, c\}$? São: $\emptyset$, $\{a\}$, $\{b\}$, $\{c\}$, $\{a, b\}$, $\{a, c\}$, $\{b, c\}$, $\{a, b, c\}$. O conjunto de todos estes subconjuntos é $\wp(\{a, b, c\})$:

$$\wp(\{a, b, c\}) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$$

Problema 1.2. Liste todos os subconjuntos de $\{a, b, c, d\}$.

Problema 1.3. Mostre que se A tem n **membros**, então $\wp(A)$ tem 2^n **membros**.

`content/sets-functions-relations/sets/important-sets.tex`

1.3 Alguns Conjuntos Importantes

sfr:set:imp:
sec

Exemplo 1.12. Trabalharemos sobretudo com conjuntos cujos **membros** são objetos matemáticos. Quatro desses conjuntos são tão importantes que têm nomes específicos:

$$\mathbb{N} = \{0, 1, 2, 3, \dots\}$$

o conjunto dos números naturais

$$\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$$

o conjunto dos números inteiros

$$\mathbb{Q} = \{m/n : m, n \in \mathbb{Z} \text{ e } n \neq 0\}$$

o conjunto dos números racionais

$$\mathbb{R} = (-\infty, \infty)$$

o conjunto dos números reais (o contínuo)

São todos conjuntos *infinitos*, isto é, cada um tem infinitos **membros**.

À medida que percorremos estes conjuntos, vamos acrescentando *mais* números ao nosso repertório. De fato, deve ficar claro que $\mathbb{N} \subseteq \mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R}$: afinal, todo número natural é um inteiro; todo inteiro é racional; e todo racional é real. Igualmente, deve ficar claro que $\mathbb{N} \subsetneq \mathbb{Z} \subsetneq \mathbb{Q}$, pois -1 é inteiro mas não natural, e $1/2$ é racional mas não inteiro. É menos óbvio que $\mathbb{Q} \subsetneq \mathbb{R}$, isto é, que há números reais que não são racionais.

Também usaremos por vezes o conjunto dos inteiros positivos $\mathbb{Z}^+ = \{1, 2, 3, \dots\}$ e o conjunto que contém apenas os dois primeiros naturais $\mathbb{B} = \{0, 1\}$.

Exemplo 1.13 (Cadeias). Outro exemplo interessante é o conjunto A^* das *cadeias finitas* sobre um alfabeto A : qualquer sequência finita de elementos de A é uma cadeia sobre A . Incluímos a *cadeia vazia* Λ entre as cadeias sobre A , para todo alfabeto A . Por exemplo,

$$\mathbb{B}^* = \{\Lambda, 0, 1, 00, 01, 10, 11,$$

$$000, 001, 010, 011, 100, 101, 110, 111, 0000, \dots\}.$$

Se $x = x_1 \dots x_n \in A^*$ é uma cadeia constituída por n “letras” de A , então dizemos que o *comprimento* da cadeia é n e escrevemos $\text{len}(x) = n$.

Exemplo 1.14 (Sequências infinitas). Para qualquer conjunto A podemos também considerar o conjunto A^ω das sequências infinitas de **membros** de A . Uma sequência infinita $a_1 a_2 a_3 a_4 \dots$ consiste numa lista infinita numa só direção de objetos, cada um dos quais é **um membro** de A .

`content/sets-functions-relations/sets/unions-and-intersections.tex`

1.4 Uniões e Interseções

explanation Na **Seção 1.1**, introduzimos definições de conjuntos por abstração, isto é, definições da forma $\{x : \varphi(x)\}$. Aqui invocamos alguma propriedade φ , e essa propriedade pode mencionar conjuntos que já definimos. Assim, por exemplo, se A e B são conjuntos, o conjunto $\{x : x \in A \vee x \in B\}$ consiste em todos aqueles objetos que são **membros** de A ou de B , isto é, é o conjunto que reúne os **membros** de A e B . Podemos visualizar isto como na **Figura 1.1**, onde a área realçada indica os **membros** dos dois conjuntos A e B juntos. sfr:set:uni:sec

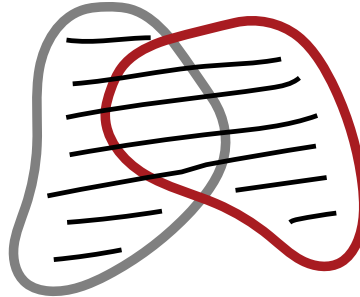


Figura 1.1: A união $A \cup B$ de dois conjuntos é o conjunto dos **membros** de A juntamente com os de B .

Esta operação sobre conjuntos—combiná-los—é muito útil e comum, e por isso damos-lhe um nome formal e um símbolo. sfr:set:uni:fig:union

Definição 1.15 (União). A *união* de dois conjuntos A e B , escrita $A \cup B$, é o conjunto de todas as coisas que são **membros** de A , de B , ou de ambos.

$$A \cup B = \{x : x \in A \vee x \in B\}$$

Exemplo 1.16. Como a multiplicidade de **membros** não importa, a união de dois conjuntos que têm **um membro** em comum contém esse **membro** apenas uma vez, por exemplo, $\{a, b, c\} \cup \{a, 0, 1\} = \{a, b, c, 0, 1\}$.

A união de um conjunto com um dos seus subconjuntos é simplesmente o conjunto maior: $\{a, b, c\} \cup \{a\} = \{a, b, c\}$.

1.4. UNIÕES E INTERSEÇÕES

A união de um conjunto com o conjunto vazio é idêntica ao conjunto: $\{a, b, c\} \cup \emptyset = \{a, b, c\}$.

Problema 1.4. Prove que se $A \subseteq B$, então $A \cup B = B$.

Também podemos considerar uma operação “dual” à união. É a operação [explanation](#) que forma o conjunto de todos os **membros** que são **membros** de A e são também **membros** de B . Esta operação chama-se *interseção*, e pode representar-se como na **Figura 1.2**.

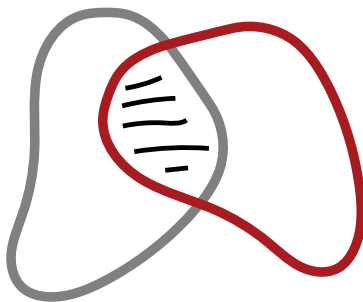


Figura 1.2: A interseção $A \cap B$ de dois conjuntos é o conjunto dos **membros** que têm em comum.

[sfr:set:uni:](#)
[fig:intersection](#)

Definição 1.17 (Interseção). A *interseção* de dois conjuntos A e B , escrita $A \cap B$, é o conjunto de todas as coisas que são **membros** de ambos A e B .

$$A \cap B = \{x : x \in A \wedge x \in B\}$$

Dois conjuntos dizem-se *disjuntos* se a sua interseção é vazia. Isso significa que não têm **membros** em comum.

Exemplo 1.18. Se dois conjuntos não têm **membros** em comum, a sua interseção é vazia: $\{a, b, c\} \cap \{0, 1\} = \emptyset$.

Se dois conjuntos têm **membros** em comum, a sua interseção é o conjunto de todos esses **membros**: $\{a, b, c\} \cap \{a, b, d\} = \{a, b\}$.

A interseção de um conjunto com um dos seus subconjuntos é simplesmente o conjunto menor: $\{a, b, c\} \cap \{a, b\} = \{a, b\}$.

A interseção de qualquer conjunto com o conjunto vazio é vazia: $\{a, b, c\} \cap \emptyset = \emptyset$.

Problema 1.5. Prove rigorosamente que se $A \subseteq B$, então $A \cap B = A$.

Também podemos formar a união ou a interseção de mais de dois conjuntos. [explanation](#) Uma forma elegante de tratar disso em geral é a seguinte: suponhamos que reunimos num único conjunto todos os conjuntos dos quais queremos formar a união (ou a interseção). Então podemos definir a união de todos os nossos conjuntos originais como o conjunto de todos os objetos que pertencem a pelo menos um **membro** desse conjunto, e a interseção como o conjunto de todos os objetos que pertencem a todo **membro** desse conjunto.

Definição 1.19. Se A é um conjunto de conjuntos, então $\bigcup A$ é o conjunto dos **membros** dos **membros** de A :

$$\begin{aligned}\bigcup A &= \{x : x \text{ pertence a um membro de } A\}, \text{ isto é,} \\ &= \{x : \text{existe um } B \in A \text{ tal que } x \in B\}\end{aligned}$$

Definição 1.20. Se A é um conjunto de conjuntos, então $\bigcap A$ é o conjunto dos objetos que todos os elementos de A têm em comum:

$$\begin{aligned}\bigcap A &= \{x : x \text{ pertence a todo membro de } A\}, \text{ isto é,} \\ &= \{x : \text{para todo } B \in A, x \in B\}\end{aligned}$$

Exemplo 1.21. Suponha $A = \{\{a, b\}, \{a, d, e\}, \{a, d\}\}$. Então $\bigcup A = \{a, b, d, e\}$ e $\bigcap A = \{a\}$.

Problema 1.6. Mostre que se A é um conjunto e $A \in B$, então $A \subseteq \bigcup B$.

Também poderíamos fazer o mesmo para uma sequência de conjuntos $A_1, A_2, \dots$

$$\begin{aligned}\bigcup_i A_i &= \{x : x \text{ pertence a um dos } A_i\} \\ \bigcap_i A_i &= \{x : x \text{ pertence a todos os } A_i\}.\end{aligned}$$

Quando temos um *índice* de conjuntos, isto é, algum conjunto I tal que consideramos A_i para cada $i \in I$, podemos também usar estas abreviaturas:

$$\begin{aligned}\bigcup_{i \in I} A_i &= \bigcup \{A_i : i \in I\} \\ \bigcap_{i \in I} A_i &= \bigcap \{A_i : i \in I\}\end{aligned}$$

Finalmente, podemos querer pensar no conjunto de todos os **membros** em A que não estão em B . Podemos representá-lo como na **Figura 1.3**.

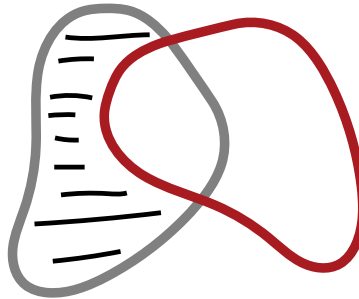


Figura 1.3: A diferença $A \setminus B$ de dois conjuntos é o conjunto dos **membros** de A que não são também **membros** de B .

1.5. PARES, TUPLAS, PRODUTOS CARTESIANOS

Definição 1.22 (Diferença). A *diferença* de conjuntos $A \setminus B$ é o conjunto de todos os **membros** de A que não são também **membros** de B , isto é,

$$A \setminus B = \{x : x \in A \text{ e } x \notin B\}.$$

Problema 1.7. Prove que se $A \subsetneq B$, então $B \setminus A \neq \emptyset$.

`content/sets-functions-relations/sets/pairs-and-products.tex`

1.5 Pares, Tuplas, Produtos Cartesianos

sfr:set:pai:
sec Da extensionalidade segue-se que os conjuntos não têm ordem entre os seus explanation elementos. Por isso, se queremos representar ordem, usamos *pares ordenados* $\langle x, y \rangle$. Num par não ordenado $\{x, y\}$, a ordem não importa: $\{x, y\} = \{y, x\}$. Num par ordenado, importa: se $x \neq y$, então $\langle x, y \rangle \neq \langle y, x \rangle$.

Como devemos pensar os pares ordenados na teoria dos conjuntos? O essencial é preservar a ideia de que pares ordenados são idênticos se e só se compartilham o mesmo primeiro elemento e o mesmo segundo elemento, isto é:

$$\langle a, b \rangle = \langle c, d \rangle \text{ se e só se } a = c \text{ e } b = d.$$

Podemos definir pares ordenados na teoria dos conjuntos usando a definição de Wiener–Kuratowski.

sfr:set:pai:
wienerkuratowski **Definição 1.23 (Par ordenado).** $\langle a, b \rangle = \{\{a\}, \{a, b\}\}$.

Problema 1.8. Usando **Definição 1.23**, prove que $\langle a, b \rangle = \langle c, d \rangle$ se e só se $a = c$ e $b = d$.

Tendo fixado uma definição de par ordenado, podemos usá-la para definir explanation outros conjuntos. Por exemplo, por vezes queremos sequências ordenadas de mais de dois objetos, p.ex., *triplos* $\langle x, y, z \rangle$, *quádruplos* $\langle x, y, z, u \rangle$, e assim por diante. Podemos pensar nos triplos como pares ordenados especiais, em que o primeiro elemento é ele próprio um par ordenado: $\langle x, y, z \rangle$ é $\langle \langle x, y \rangle, z \rangle$. O mesmo vale para quádruplos: $\langle x, y, z, u \rangle$ é $\langle \langle \langle x, y \rangle, z \rangle, u \rangle$, e assim por diante. Em geral, falamos de *n-tuplas ordenadas* $\langle x_1, \dots, x_n \rangle$.

Certos conjuntos de pares ordenados, ou outras *n-tuplas ordenadas*, serão úteis.

Definição 1.24 (Produto cartesiano). Dados conjuntos A e B , o seu *produto cartesiano* $A \times B$ é definido por

$$A \times B = \{\langle x, y \rangle : x \in A \text{ e } y \in B\}.$$

Exemplo 1.25. Se $A = \{0, 1\}$, e $B = \{1, a, b\}$, então o seu produto é

$$A \times B = \{\langle 0, 1 \rangle, \langle 0, a \rangle, \langle 0, b \rangle, \langle 1, 1 \rangle, \langle 1, a \rangle, \langle 1, b \rangle\}.$$

Exemplo 1.26. Se A é um conjunto, o produto de A consigo próprio, $A \times A$, também se escreve A^2 . É o conjunto de *todos* os pares $\langle x, y \rangle$ com $x, y \in A$. O conjunto de todos os triplos $\langle x, y, z \rangle$ é A^3 , e assim por diante. Podemos dar uma definição recursiva:

$$\begin{aligned} A^1 &= A \\ A^{k+1} &= A^k \times A \end{aligned}$$

Problema 1.9. Liste todos os **membros** de $\{1, 2, 3\}^3$.

Proposição 1.27. Se A tem n **membros** e B tem m **membros**, então $A \times B$ tem $n \cdot m$ elementos. sfr:set:pa: cardnmprod

Prova. Para todo **membro** x em A , há m **membros** da forma $\langle x, y \rangle \in A \times B$. Seja $B_x = \{\langle x, y \rangle : y \in B\}$. Como sempre que $x_1 \neq x_2$, $\langle x_1, y \rangle \neq \langle x_2, y \rangle$, tem-se $B_{x_1} \cap B_{x_2} = \emptyset$. Mas se $A = \{x_1, \dots, x_n\}$, então $A \times B = B_{x_1} \cup \dots \cup B_{x_n}$, e portanto tem $n \cdot m$ **membros**.

Para visualizar isto, dispomos os **membros** de $A \times B$ numa grade:

$$\begin{array}{llll} B_{x_1} = & \{\langle x_1, y_1 \rangle & \langle x_1, y_2 \rangle & \dots & \langle x_1, y_m \rangle\} \\ B_{x_2} = & \{\langle x_2, y_1 \rangle & \langle x_2, y_2 \rangle & \dots & \langle x_2, y_m \rangle\} \\ & \vdots & & & \vdots \\ B_{x_n} = & \{\langle x_n, y_1 \rangle & \langle x_n, y_2 \rangle & \dots & \langle x_n, y_m \rangle\} \end{array}$$

Como os x_i são todos distintos, e os y_j são todos distintos, quaisquer dois pares nesta grade são distintos, e há $n \cdot m$ deles. □

Problema 1.10. Mostre, por indução em k , que para todo $k \geq 1$, se A tem n **membros**, então A^k tem n^k **membros**.

Exemplo 1.28. Se A é um conjunto, uma *palavra* sobre A é qualquer sequência de **membros** de A . Uma sequência pode ser vista como uma n -tupla de **membros** de A . Por exemplo, se $A = \{a, b, c\}$, então a sequência “bac” pode ser vista como o triplo $\langle b, a, c \rangle$. Palavras, isto é, sequências de símbolos, são de importância crucial na ciência da computação. Por convenção, contamos os **membros** de A como sequências de comprimento 1, e $\emptyset$ como a sequência de comprimento 0. O conjunto de *todas* as palavras sobre A é então

$$A^* = \{\emptyset\} \cup A \cup A^2 \cup A^3 \cup \dots$$

1.6. PARADOXO DE RUSSELL

1.6 Paradoxo de Russell

sfr:set:rus:
sec A extensionalidade autoriza a notação $\{x : \varphi(x)\}$, para o conjunto dos x tais que $\varphi(x)$. Contudo, tudo o que a extensionalidade *realmente* autoriza é o seguinte pensamento. Se há um conjunto cujos membros são exatamente os que satisfazem φ , então há apenas um tal conjunto. Dito de outro modo: fixada alguma φ , o conjunto $\{x : \varphi(x)\}$ é único, se existir.

Mas este condicional é importante! Crucialmente, nem toda propriedade se presta à *compreensão*. Isto é, algumas propriedades *não* definem conjuntos. Se todas definissem, cairíamos em contradições flagrantes. O exemplo mais famoso disso é o Paradoxo de Russell.

Os conjuntos podem ser **membros** de outros conjuntos—por exemplo, o conjunto potência de um conjunto A é constituído por conjuntos. Por isso faz sentido perguntar ou investigar se um conjunto é **um membro** de outro conjunto. Pode um conjunto ser membro de si próprio? Nada na ideia de conjunto parece excluir isso. Por exemplo, se *todos* os conjuntos formam uma coleção de objetos, alguém poderia pensar que podem reunir-se num único conjunto—o conjunto de todos os conjuntos. E ele, sendo conjunto, seria **um membro** do conjunto de todos os conjuntos.

O Paradoxo de Russell surge quando consideramos a propriedade de não ter a si próprio como **um membro**, de ser *não auto-membro*. E se supusermos que há um conjunto de todos os conjuntos que não têm a si próprios como **um membro**? Será que

$$R = \{x : x \notin x\}$$

existe? Verifica-se que podemos provar que não.

sfr:set:rus:
thm:russells-paradox **Teorema 1.29 (Paradoxo de Russell).** Não há conjunto $R = \{x : x \notin x\}$.

Prova. Se $R = \{x : x \notin x\}$ existisse, então $R \in R$ se e só se $R \notin R$, o que é uma contradição. $\square$

Vamos percorrer esta prova mais devagar. Se R existe, faz sentido perguntar se $R \in R$ ou não. Suponhamos que de fato $R \in R$. Agora, R foi definido como o conjunto de todos os conjuntos que não são **membros** de si próprios. Logo, se $R \in R$, então R não tem a propriedade definidora de R . Mas só os conjuntos que têm essa propriedade estão em R , logo R não pode ser **um membro** de R , isto é, $R \notin R$. Mas R não pode ao mesmo tempo ser e não ser **um membro** de R , logo temos uma contradição.

Como a suposição de que $R \in R$ leva a uma contradição, temos $R \notin R$. Mas isto também leva a uma contradição! Pois se $R \notin R$, então R tem a propriedade definidora de R , e assim R seria **um membro** de R , tal como todos os outros conjuntos não auto-membros. E de novo, R não pode ao mesmo tempo não ser e ser **um membro** de R .

Como montar uma teoria dos conjuntos que evite cair no Paradoxo de Russell, isto é, que evite afirmar de modo *inconsistente* que $R = \{x : x \notin x\}$ existe?

explanation

digression

Teríamos de estabelecer axiomas que nos dêem condições muito precisas para dizer quando os conjuntos existem (e quando não existem).

A teoria dos conjuntos esboçada neste capítulo não faz isso. É *genuinamente ingênua*. Diz apenas que os conjuntos obedecem à extensionalidade e que, se temos alguns conjuntos, podemos formar a sua união, interseção, etc. É possível desenvolver a teoria dos conjuntos com mais rigor do que isto.

Capítulo 2

Relações

`content/sets-functions-relations/rerelations/rerelations-as-sets.tex`

2.1 Relações como Conjuntos

explanation Na **Seção 1.3**, mencionamos alguns conjuntos importantes: $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$. Sem sfr:rel:set:sec dúvida que se lembram de algumas relações interessantes entre os **membros** de alguns destes conjuntos. Por exemplo, cada um destes conjuntos tem uma *relação de ordem* completamente padrão sobre si. Há também a relação *ser idêntico a*, segundo a qual cada objeto só se relaciona consigo próprio e com nenhum outro. Há muitas outras relações interessantes que encontraremos, e ainda mais relações possíveis. Antes de as revermos, contudo, começaremos por observar que podemos ver relações como um tipo especial de conjunto.

Para isso, recordemos duas coisas da **Seção 1.5**. Primeiro, recordemos a noção de *par ordenado*: dados a e b , podemos formar $\langle a, b \rangle$. É importante notar que a ordem dos elementos *importa* aqui. Assim, se $a \neq b$ então $\langle a, b \rangle \neq \langle b, a \rangle$. (Contrastemos isto com pares não ordenados, isto é, conjuntos de 2 elementos, onde $\{a, b\} = \{b, a\}$.) Segundo, recordemos a noção de *produto cartesiano*: se A e B são conjuntos, então podemos formar $A \times B$, o conjunto de todos os pares $\langle x, y \rangle$ com $x \in A$ e $y \in B$. Em particular, $A^2 = A \times A$ é o conjunto de todos os pares ordenados de A .

Consideremos agora uma relação particular num conjunto: a relação $<$ no conjunto $\mathbb{N}$ dos números naturais. Seja o conjunto de todos os pares de números $\langle n, m \rangle$ onde $n < m$, isto é,

$$R = \{\langle n, m \rangle : n, m \in \mathbb{N} \text{ e } n < m\}.$$

2.1. RELAÇÕES COMO CONJUNTOS

Há uma ligação estreita entre n ser menor que m , e o par $\langle n, m \rangle$ ser membro de R , nomeadamente:

$$n < m \text{ se e só se } \langle n, m \rangle \in R.$$

De fato, sem qualquer perda de informação, podemos considerar o conjunto R como *sendo* a relação $<$ em $\mathbb{N}$.

Do mesmo modo podemos construir um subconjunto de $\mathbb{N}^2$ para qualquer relação entre números. Reciprocamente, dado qualquer conjunto de pares de números $S \subseteq \mathbb{N}^2$, há uma relação correspondente entre números, nomeadamente, a relação segundo a qual n se liga a m se e só se $\langle n, m \rangle \in S$. Isto justifica a definição seguinte:

Definição 2.1 (Relação binária). Uma *relação binária* num conjunto A é um subconjunto de A^2 . Se $R \subseteq A^2$ é uma relação binária em A e $x, y \in A$, por vezes escrevemos Rxy (ou xRy) para $\langle x, y \rangle \in R$.

Exemplo 2.2. O conjunto $\mathbb{N}^2$ dos pares de números naturais pode ser listado numa matriz bidimensional assim:

$$\begin{array}{ccccc} \langle \mathbf{0}, \mathbf{0} \rangle & \langle 0, 1 \rangle & \langle 0, 2 \rangle & \langle 0, 3 \rangle & \dots \\ \langle 1, 0 \rangle & \langle \mathbf{1}, \mathbf{1} \rangle & \langle 1, 2 \rangle & \langle 1, 3 \rangle & \dots \\ \langle 2, 0 \rangle & \langle 2, 1 \rangle & \langle \mathbf{2}, \mathbf{2} \rangle & \langle 2, 3 \rangle & \dots \\ \langle 3, 0 \rangle & \langle 3, 1 \rangle & \langle 3, 2 \rangle & \langle \mathbf{3}, \mathbf{3} \rangle & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{array}$$

Colocamos a diagonal, aqui, em negrito, pois o subconjunto de $\mathbb{N}^2$ constituído pelos pares que estão na diagonal, isto é,

$$\{\langle 0, 0 \rangle, \langle 1, 1 \rangle, \langle 2, 2 \rangle, \dots\},$$

é a *relação identidade* em $\mathbb{N}$. (Como a relação identidade é popular, definamos $\text{Id}_A = \{\langle x, x \rangle : x \in A\}$ para qualquer conjunto A .) O subconjunto de todos os pares acima da diagonal, isto é,

$$L = \{\langle 0, 1 \rangle, \langle 0, 2 \rangle, \dots, \langle 1, 2 \rangle, \langle 1, 3 \rangle, \dots, \langle 2, 3 \rangle, \langle 2, 4 \rangle, \dots\},$$

é a relação *menor que*, isto é, Lnm se e só se $n < m$. O subconjunto dos pares abaixo da diagonal, isto é,

$$G = \{\langle 1, 0 \rangle, \langle 2, 0 \rangle, \langle 2, 1 \rangle, \langle 3, 0 \rangle, \langle 3, 1 \rangle, \langle 3, 2 \rangle, \dots\},$$

é a relação *maior que*, isto é, Gnm se e só se $n > m$. A união de L com I , que podemos chamar $K = L \cup I$, é a relação *menor ou igual a*: $Kn m$ se e só se $n \leq m$. Do mesmo modo, $H = G \cup I$ é a *relação maior ou igual a*. Estas relações L , G , K , e H são tipos especiais de relações chamadas *ordens*. L e G gozam da propriedade de que nenhum número está em relação L ou G consigo próprio (isto é, para todo n , não se verifica nem Lnn nem Gnn). Relações com esta propriedade chamam-se *irreflexivas*, e, se também forem ordens, chamam-se *ordens estritas*.

explanation

Embora ordens e identidade sejam relações importantes e naturais, convém enfatizar que, segundo a nossa definição, *qualquer* subconjunto de A^2 é uma relação em A , por mais artificial ou forçado que pareça. Em particular, $\emptyset$ é uma relação em qualquer conjunto (a *relação vazia*, que nenhum par de elementos satisfaz), e A^2 em si é também uma relação em A (uma que todo par de elementos satisfaz), chamada a *relação universal*. Mas também algo como $E = \{\langle n, m \rangle : n > 5 \text{ ou } m \times n \geq 34\}$ conta como relação.

Problema 2.1. Liste os **membros** da relação $\subseteq$ no conjunto $\wp(\{a, b, c\})$.

`content/sets-functions-relations/relations/reflections.tex`

2.2 Reflexões Filosóficas

Na [Seção 2.1](#), definimos relações como certos conjuntos. Devemos fazer uma pausa e colocar uma questão filosófica rápida: o que é que tal definição *faz*? É extremamente duvidoso que devamos querer dizer que *descobrimos* fatos metafísicos de identidade; que, por exemplo, a relação de ordem em $\mathbb{N}$ *acabou por ser* o conjunto $R = \{\langle n, m \rangle : n, m \in \mathbb{N} \text{ e } n < m\}$ que definimos na [Seção 2.1](#). Eis três razões para isso.

[sfr:rel:ref:sec](#)

Primeiro: na [Definição 1.23](#), definimos $\langle a, b \rangle = \{\{a\}, \{a, b\}\}$. Consideremos em vez disso a definição $\|a, b\| = \{\{b\}, \{a, b\}\} = \langle b, a \rangle$. Quando $a \neq b$, temos que $\langle a, b \rangle \neq \|a, b\|$. Mas poderíamos com a mesma facilidade ter tomado $\|a, b\|$ como a nossa definição de par ordenado, em vez de $\langle a, b \rangle$. Ambas as definições teriam funcionado igualmente bem. Assim, temos dois candidatos igualmente bons a “ser” a relação de ordem nos números naturais, nomeadamente:

$$R = \{\langle n, m \rangle : n, m \in \mathbb{N} \text{ e } n < m\}$$

$$S = \{\|n, m\| : n, m \in \mathbb{N} \text{ e } n < m\}.$$

Como $R \neq S$, pela extensionalidade, é claro que não podem *ambos* ser idênticos à relação de ordem em $\mathbb{N}$. Mas seria meramente arbitrário, e por isso um pouco embaraçoso, afirmar que R em vez de S (ou vice-versa) *é* a relação de ordenação, como questão de fato. (Isto é um caso muito simples de um argumento contra o reducionismo em teoria dos conjuntos que Benacerraf tornou famoso em [1965](#). Voltaremos a ele várias vezes.)

Segundo: se pensamos que *toda* a relação deve ser identificada com um conjunto, então a própria relação de pertinência a conjuntos, $\in$, deveria ser um conjunto particular. De fato, teria de ser o conjunto $\{\langle x, y \rangle : x \in y\}$. Mas será que este conjunto existe? Dado o Paradoxo de Russell, é uma afirmação não trivial que tal conjunto exista. De fato, é possível desenvolver a teoria dos conjuntos de modo rigoroso como teoria axiomática, e essa teoria de fato nega a existência deste conjunto. Logo, mesmo que algumas relações possam ser tratadas como conjuntos, a relação de pertinência a conjuntos terá de ser um caso especial.

2.3. PROPRIEDADES ESPECIAIS DAS RELAÇÕES

Terceiro: quando “identificamos” relações com conjuntos, dissemos que nos permitiríamos escrever Rxy para $\langle x, y \rangle \in R$. Isto está bem, desde que a relação de pertinência, “ $\in$ ”, seja tratada *como* um predicado. Mas se pensamos que “ $\in$ ” denota um certo tipo de conjunto, então a expressão “ $\langle x, y \rangle \in R$ ” consiste apenas em três termos singulares que denotam conjuntos: “ $\langle x, y \rangle$ ”, “ $\in$ ”, e “ R ”. E tal lista de nomes não é mais capaz de exprimir uma proposição do que a sequência sem sentido: “o copo porta-canetas a mesa”. De novo, mesmo que algumas relações possam ser tratadas como conjuntos, a relação de pertinência a conjuntos tem de ser um caso especial. (Isto reúne num só argumento uma versão simples do paradoxo do conceito *cavalo* de Frege, e uma objeção célebre que Wittgenstein dirigiu a Russell.)

Então onde nos deixa isto? Bem, não há nada de *errado* em dizermos que as relações nos números são conjuntos. Só temos de compreender o espírito em que tal observação é feita. Não enunciamos um fato metafísico de identidade. Limitamo-nos a notar que, em certos contextos, podemos (e iremos) *tratar* (certas) relações como certos conjuntos.

`content/sets-functions-relations/relations/special-properties.tex`

2.3 Propriedades Especiais das Relações

sfr:rel:prp:
sec Alguns tipos de relações revelam-se tão comuns que receberam nomes especiais. intro Por exemplo, $\leq$ e $\subseteq$ relacionam os seus respectivos domínios (digamos, $\mathbb{N}$ no caso de $\leq$ e $\wp(A)$ no caso de $\subseteq$) de modos semelhantes. Para precisar exatamente como estas relações são semelhantes, e como diferem, as categorizamos segundo algumas propriedades especiais que as relações podem ter. Verifica-se que (combinações de) algumas destas propriedades especiais são especialmente importantes: ordens e relações de equivalência.

Definição 2.3 (Reflexividade). Uma relação $R \subseteq A^2$ é *reflexiva* se e só se, para todo $x \in A$, Rxx .

Definição 2.4 (Transitividade). Uma relação $R \subseteq A^2$ é *transitiva* se e só se, sempre que Rxy e Ryz , então também Rxz .

Definição 2.5 (Simetria). Uma relação $R \subseteq A^2$ é *simétrica* se e só se, sempre que Rxy , então também Ryx .

Definição 2.6 (Antissimetria). Uma relação $R \subseteq A^2$ é *antissimétrica* se e só se, sempre que tanto Rxy quanto Ryx , então $x = y$ (ou, noutras palavras: se $x \neq y$ então ou $\neg Rxy$ ou $\neg Ryx$).

Numa relação simétrica, Rxy e Ryx verificam-se sempre em conjunto, ou explanation nenhuma se verifica. Numa relação antissimétrica, a única forma de Rxy e Ryx se verificarem em conjunto é que $x = y$. Note-se que isto não *exige* que Rxy e Ryx se verifiquem quando $x = y$, apenas que isso não está excluído.

Assim, uma relação antissimétrica pode ser reflexiva, mas não é o caso que toda relação antissimétrica seja reflexiva. Note-se também que ser antissimétrica e ser meramente não simétrica são condições diferentes. De fato, uma relação pode ser ao mesmo tempo simétrica e antissimétrica (por exemplo, a relação identidade o é).

Definição 2.7 (Conexidade). Uma relação $R \subseteq A^2$ é *conexa* se para todos $x, y \in A$, se $x \neq y$, então ou Rxy ou Ryx .

Problema 2.2. Dê exemplos de relações que sejam (a) reflexivas e simétricas mas não transitivas, (b) reflexivas e antissimétricas, (c) antissimétricas, transitivas, mas não reflexivas, e (d) reflexivas, simétricas e transitivas. Não use relações sobre números ou conjuntos.

Definição 2.8 (Irreflexividade). Uma relação $R \subseteq A^2$ diz-se *irreflexiva* se, para todo $x \in A$, não Rxx .

Definição 2.9 (Assimetria). Uma relação $R \subseteq A^2$ diz-se *assimétrica* se para nenhum par $x, y \in A$ se tem tanto Rxy quanto Ryx .

Note-se que se $A \neq \emptyset$, então nenhuma relação irreflexiva em A é reflexiva e toda relação assimétrica em A é também antissimétrica. Contudo, há $R \subseteq A^2$ que não são reflexivas nem irreflexivas, e há relações antissimétricas que não são assimétricas.

`content/sets-functions-relations/relations/equivalence-relations.tex`

2.4 Relações de Equivalência

A relação identidade num conjunto é reflexiva, simétrica e transitiva. Relações R que têm todas as três destas propriedades são muito comuns. sfr:rel:eqv:
sec

Definição 2.10 (Relação de equivalência). Uma relação $R \subseteq A^2$ que é reflexiva, simétrica e transitiva chama-se uma *relação de equivalência*. Membros x e y de A dizem-se *R -equivalentes* se Rxy .

As relações de equivalência dão origem à noção de *classe de equivalência*. Uma relação de equivalência “reparte” o domínio em partições distintas. Dentro de cada partição, todos os objetos estão relacionados uns com os outros; e objetos de partições diferentes não se relacionam entre si. Por vezes, é útil falar destas partições *diretamente*. Para esse fim, introduzimos uma definição:

Definição 2.11. Seja $R \subseteq A^2$ uma relação de equivalência. Para cada $x \in A$, a *classe de equivalência* de x em A é o conjunto $[x]_R = \{y \in A : Rxy\}$. O *quociente* de A por R é $A/R = \{[x]_R : x \in A\}$, isto é, o conjunto destas classes de equivalência. sfr:rel:eqv:
def:equivalenceclass

2.5. ORDENS

O resultado seguinte justifica a definição de classe de equivalência, provando que as classes de equivalência são de fato as partições de A :

Proposição 2.12. *Se $R \subseteq A^2$ é uma relação de equivalência, então Rxy se e só se $[x]_R = [y]_R$.*

Prova. Para o sentido da esquerda para a direita, suponhamos Rxy , e seja $z \in [x]_R$. Por definição, então, Rxz . Como R é uma relação de equivalência, Ryz . (Explicando melhor: como Rxy e R é simétrica temos Ryx , e como Rxz e R é transitiva temos Ryz .) Logo $z \in [y]_R$. Generalizando, $[x]_R \subseteq [y]_R$. Mas de modo perfeitamente análogo, $[y]_R \subseteq [x]_R$. Logo $[x]_R = [y]_R$, pela extensionalidade.

Para o sentido da direita para a esquerda, suponhamos $[x]_R = [y]_R$. Como R é reflexiva, Ryy , logo $y \in [y]_R$. Assim também $y \in [x]_R$ pela suposição de que $[x]_R = [y]_R$. Logo Rxy . $\square$

Exemplo 2.13. Um bom exemplo de relações de equivalência vem da aritmética modular. Para quaisquer a, b , e $n \in \mathbb{Z}^+$, digamos que $a \equiv_n b$ se e só se dividir a por n dá o mesmo resto que dividir b por n . (De modo um pouco mais simbólico: $a \equiv_n b$ se e só se, para algum $k \in \mathbb{Z}$, $a - b = kn$.) Agora, $\equiv_n$ é uma relação de equivalência, para qualquer n . E há exatamente n classes de equivalência distintas geradas por $\equiv_n$; isto é, $\mathbb{N}/\equiv_n$ tem n membros. São: o conjunto dos números divisíveis por n sem resto, isto é, $[0]_{\equiv_n}$; o conjunto dos números divisíveis por n com resto 1, isto é, $[1]_{\equiv_n}$; ...; e o conjunto dos números divisíveis por n com resto $n - 1$, isto é, $[n - 1]_{\equiv_n}$.

Problema 2.3. Mostre que $\equiv_n$ é uma relação de equivalência, para qualquer $n \in \mathbb{Z}^+$, e que $\mathbb{N}/\equiv_n$ tem exatamente n membros.

content/sets-functions-relations/relations/orders.tex

2.5 Ordens

sfr:rel:ord: Muitas das nossas comparações envolvem descrever certos objetos como sendo sec “menores que”, “iguais a” ou “maiores que” outros objetos, sob algum aspecto. explanation Isso envolve relações de *ordem*. Mas há vários tipos de relação de ordem. Por exemplo, algumas exigem que quaisquer dois objetos sejam comparáveis, outras não. Algumas incluem a identidade (como $\leq$) e outras a excluem (como $<$). Ter aqui uma taxonomia ajuda-nos.

Definição 2.14 (Pré-ordem). Uma relação que é ao mesmo tempo reflexiva e transitiva chama-se uma *pré-ordem*.

Definição 2.15 (Ordem parcial). Uma pré-ordem que também é antissimétrica chama-se uma *ordem parcial*.

Definição 2.16 (Ordem linear). Uma ordem parcial que também é conexa chama-se uma *ordem total* ou *ordem linear*. sfr:rel:ord:
def:linearorder

Exemplo 2.17. Toda ordem linear é também uma ordem parcial, e toda ordem parcial é também uma pré-ordem, mas as implicações recíprocas não valem. A relação universal sobre A é uma pré-ordem, pois é reflexiva e transitiva. Mas, se A tem mais de um **membro**, a relação universal não é antissimétrica e, portanto, não é uma ordem parcial.

Exemplo 2.18. Considere a relação *não mais longa que* $\preceq$ sobre $\mathbb{B}^*$: $x \preceq y$ sse $\text{len}(x) \leq \text{len}(y)$. Trata-se de uma pré-ordem (reflexiva e transitiva), e até conexa, mas não é uma ordem parcial, pois não é antissimétrica. Por exemplo, $01 \preceq 10$ e $10 \preceq 01$, mas $01 \neq 10$.

Exemplo 2.19. Uma ordem parcial importante é a relação $\subseteq$ sobre um conjunto de conjuntos. Em geral não é uma ordem linear: se $a \neq b$ e consideramos $\wp(\{a, b\}) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$, vemos que $\{a\} \not\subseteq \{b\}$ e $\{a\} \neq \{b\}$ e $\{b\} \not\subseteq \{a\}$.

Exemplo 2.20. A relação de *divisibilidade sem resto* nos dá uma ordem parcial que não é linear. Para inteiros n e m , escrevemos $n \mid m$ para significar que n divide m (exatamente), isto é, sse existe algum inteiro k tal que $m = kn$. Sobre $\mathbb{N}$, isso é uma ordem parcial, mas não linear: por exemplo, $2 \nmid 3$ e também $3 \nmid 2$. Vista como relação sobre $\mathbb{Z}$, a divisibilidade é apenas uma pré-ordem, pois não é antissimétrica: $1 \mid -1$ e $-1 \mid 1$ mas $1 \neq -1$.

Exemplo 2.21. A relação de *extensão* sobre um conjunto de seqüências A^* é a seguinte: $s \sqsubseteq s'$ sse $s = \Lambda$ (a seqüência vazia), $s = s'$, ou $s = \langle s_1, \dots, s_n \rangle$ e $s' = \langle s_1, \dots, s_n, s_{n+1}, \dots, s_m \rangle$. Se $s \sqsubseteq s'$ dizemos também que s é um *segmento inicial* de s' . A relação de extensão sobre A^* é uma ordem parcial, mas não linear, p.ex., se $a \neq b$, então $ab \not\sqsubseteq ba$ e $ba \not\sqsubseteq ab$.

Definição 2.22 (Ordem estrita). Uma *ordem estrita* é uma relação irreflexiva, assimétrica e transitiva.

Definição 2.23 (Ordem linear estrita). Uma ordem estrita que também é conexa chama-se *ordem total estrita* ou *ordem linear estrita*. sfr:rel:ord:
def:strictlinearorder

Exemplo 2.24. $\leq$ é a ordem linear correspondente à ordem linear estrita $<$. $\subseteq$ é a ordem parcial correspondente à ordem estrita $\subsetneq$.

Qualquer ordem estrita R sobre A pode transformar-se numa ordem parcial adicionando a diagonal Id_A , isto é, adicionando todos os pares $\langle x, x \rangle$. (Isto chama-se o *fecho reflexivo* de R .) Reciprocamente, a partir de uma ordem parcial obtém-se uma ordem estrita removendo Id_A . Os dois resultados seguintes tornam isto preciso.

2.5. ORDENS

sfr:rel:ord:
prop:stricttopartial

Proposição 2.25. *Se R é uma ordem estrita sobre A , então $R^+ = R \cup \text{Id}_A$ é uma ordem parcial. Além disso, se R é uma ordem linear estrita, então R^+ é uma ordem linear.*

Prova. Suponhamos que R é uma ordem estrita, isto é, $R \subseteq A^2$ e R é irreflexiva, assimétrica e transitiva. Seja $R^+ = R \cup \text{Id}_A$. Temos de mostrar que R^+ é reflexiva, antissimétrica e transitiva.

R^+ é claramente reflexiva, pois $\langle x, x \rangle \in \text{Id}_A \subseteq R^+$ para todo $x \in A$.

Para mostrar que R^+ é antissimétrica, suponhamos, para redução ao absurdo, que R^+xy e R^+yx mas $x \neq y$. Como $\langle x, y \rangle \in R \cup \text{Id}_A$, mas $\langle x, y \rangle \notin \text{Id}_A$, temos de ter $\langle x, y \rangle \in R$, isto é, Rxy . Do mesmo modo, Ryx . Mas isto contradiz a hipótese de que R é assimétrica.

Para estabelecer a transitividade, suponhamos que R^+xy e R^+yz . Se tanto $\langle x, y \rangle \in R$ como $\langle y, z \rangle \in R$, então $\langle x, z \rangle \in R$, pois R é transitiva. Caso contrário, ou $\langle x, y \rangle \in \text{Id}_A$, isto é, $x = y$, ou $\langle y, z \rangle \in \text{Id}_A$, isto é, $y = z$. No primeiro caso, temos R^+yz por hipótese, $x = y$, logo R^+xz . Analogamente no segundo caso. Em qualquer caso, R^+xz ; assim, R^+ é também transitiva.

Quanto à cláusula “além disso”, suponhamos que R é também conexa. Então, para todo $x \neq y$, ou Rxy ou Ryx , isto é, ou $\langle x, y \rangle \in R$ ou $\langle y, x \rangle \in R$. Como $R \subseteq R^+$, isto continua a valer para R^+ , logo R^+ é também conexa. $\square$

sfr:rel:ord:
prop:partialtostrict

Proposição 2.26. *Se R é uma ordem parcial sobre A , então $R^- = R \setminus \text{Id}_A$ é uma ordem estrita. Além disso, se R é uma ordem linear, então R^- é uma ordem linear estrita.*

Prova. Deixamos como exercício. $\square$

Problema 2.4. Apresente uma prova da **Proposição 2.26**.

O resultado simples a seguir estabelece que ordens lineares estritas satisfazem uma propriedade no estilo da extensionalidade:

sfr:rel:ord:
p:extensionality-strictlinearorders

Proposição 2.27. *Se $<$ é uma ordem linear estrita sobre A , então:*

$$(\forall a, b \in A)((\forall x \in A)(x < a \leftrightarrow x < b) \rightarrow a = b).$$

Prova. Suponhamos $(\forall x \in A)(x < a \leftrightarrow x < b)$. Se $a < b$, então $a < a$, o que contradiz o fato de $<$ ser irreflexiva; logo $a \not< b$. De forma totalmente análoga, $b \not< a$. Logo $a = b$, pois $<$ é conexa. $\square$

2.6 Grafos

Um *grafo* é um diagrama no qual pontos—chamados “nós” ou “vértices” (plural de “vértice”)—são conectados por arestas. Grafos são uma ferramenta onipresente em matemática discreta e em ciência da computação. São incrivelmente úteis para representar e visualizar relações e estruturas, desde coisas concretas como redes de vários tipos até estruturas abstratas como os possíveis resultados de decisões. Existem muitos tipos diferentes de grafos na literatura que diferem, p.ex., quanto a se as arestas são direcionadas ou não, têm rótulos ou não, se pode haver arestas de um nó para o mesmo nó, múltiplas arestas entre os mesmos nós, etc. *Grafos direcionados* têm uma conexão especial com relações.

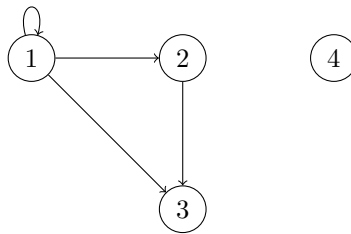
sfr:rel:grp:
sec

Definição 2.28 (Grafo direcionado). Um *grafo direcionado* $G = \langle V, E \rangle$ é um conjunto de *vértices* V e um conjunto de *arestas* $E \subseteq V^2$.

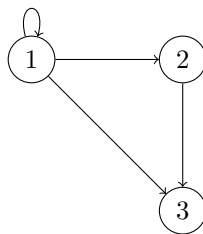
explanation

De acordo com nossa definição, um grafo é simplesmente um conjunto junto com uma relação nesse conjunto. Claro, ao falar de grafos, é natural esperar que sejam representados graficamente: podemos desenhar um grafo conectando dois vértices v_1 e v_2 por uma seta sse $\langle v_1, v_2 \rangle \in E$. A única diferença entre uma relação por si só e um grafo é que um grafo especifica o conjunto de vértices, isto é, um grafo pode ter vértices isolados. O ponto importante, porém, é que toda relação R em um conjunto X pode ser vista como um grafo direcionado $\langle X, R \rangle$, e reciprocamente, um grafo direcionado $\langle V, E \rangle$ pode ser visto como uma relação $E \subseteq V^2$ com o conjunto V explicitamente especificado.

Exemplo 2.29. O grafo $\langle V, E \rangle$ com $V = \{1, 2, 3, 4\}$ e $E = \{\langle 1, 1 \rangle, \langle 1, 2 \rangle, \langle 1, 3 \rangle, \langle 2, 3 \rangle\}$ se parece com isto:



Este é um grafo diferente de $\langle V', E \rangle$ com $V' = \{1, 2, 3\}$, que se parece com isto:



2.7. ÁRVORES

Problema 2.5. Considere a relação menor ou igual $\leq$ no conjunto $\{1, 2, 3, 4\}$ como um grafo e desenhe o diagrama correspondente.

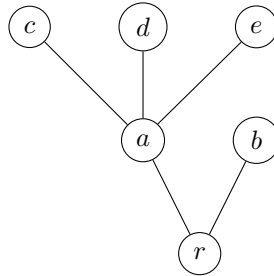
content/sets-functions-relations/relations/trees.tex

2.7 Árvores

sfr:rel:tre:
sec

Um tipo particular de ordem parcial que desempenha um papel importante em todas as partes da lógica é uma *árvore*. Árvores finitas aparecem em partes elementares da lógica: por exemplo, *fórmulas* podem ser entendidas em termos de sua decomposição em uma árvore sintática, enquanto *derivações* em muitos sistemas de *derivação* também tomam a forma de árvores finitas. Árvores infinitas aparecem já na prova dos teoremas de completude para a lógica proposicional e de primeira ordem, e são usadas em toda a lógica matemática.

O conceito de árvore da teoria dos conjuntos está intimamente relacionado à noção de árvore na teoria dos grafos. Aqui está uma figura de uma árvore (finita):



O nó mais baixo r é a raiz. Todo nó diferente de r tem exatamente um nó pai imediatamente abaixo dele. Podemos pensar na relação em que um nó x está para um nó y se y pode ser alcançado a partir de x seguindo arestas para cima como x sendo um *ancestral* de y .

A relação de ancestralidade em uma árvore é uma ordem parcial estrita. Isso motiva a definição teórico-conjuntista. Para enunciá-la, precisamos de dois conceitos. Um *menor elemento* em um conjunto A parcialmente ordenado por $\leq$ é um *membro* $x \in A$ tal que para todo $y \in A$ temos que $x \leq y$. Um conjunto é *bem ordenado* por $\leq$ se cada um de seus subconjuntos não vazios tem um menor elemento.

Definição 2.30 (Árvore). Uma *árvore* é um par $T = \langle A, \leq \rangle$ tal que A é um conjunto e $\leq$ é uma ordem parcial em A com um único menor elemento $r \in A$ (chamado de *raiz*) tal que para todo $x \in A$, o conjunto $\{y : y \leq x\}$ é bem ordenado por $\leq$.

Definição 2.31 (Sucessores). Suponha $T = \langle A, \leq \rangle$ é uma árvore. Se $x, y \in A$, $x < y$, e não existe $z \in A$ tal que $x < z < y$, então dizemos que y é um *sucessor* de x .

Os sucessores de $x \in A$ também são chamados de seus *filhos*. Se y é um sucessor de x , então chamamos x de *predecessor* ou *pai* de y .

Proposição 2.32. Se $\langle A, \leq \rangle$ é uma árvore, então todo $x \in A$ diferente da raiz tem no máximo um predecessor.

Prova. Suponha $y_1 < x$ e $y_2 < x$ e $y_1 \neq y_2$. Então $\{y_1, y_2\} \subseteq \{z : z < x\}$. Como $\{z : z < x\}$ é bem ordenado por $\leq$, seu subconjunto $\{y_1, y_2\}$ tem um menor elemento, que obviamente deve ser ou y_1 ou y_2 . Então ou $y_1 \leq y_2$ ou $y_2 \leq y_1$. Supusemos que $y_1 \neq y_2$, então na verdade ou $y_1 < y_2$ ou $y_2 < y_1$. Como supusemos que $y_1 < x$ e $y_2 < x$, temos ainda que ou $y_1 < y_2 < x$ ou $y_2 < y_1 < x$. Então y_1 e y_2 não podem ambos ser predecessores de x . $\square$

Definição 2.33. Uma árvore $T = \langle A, \leq \rangle$ é dita *infinita* se A é um conjunto infinito, e *finita* caso contrário. Se T é tal que todo $x \in A$ tem apenas um número finito de sucessores, então dizemos que T é *finitamente ramificada*.

Definição 2.34 (Ramos). Dada uma árvore $T = \langle A, \leq \rangle$, um *ramo* de T é uma cadeia maximal em T , isto é, um conjunto $B \subseteq A$ tal que para quaisquer $x, y \in B$ ou $x \leq y$ ou $y \leq x$, e para qualquer $z \in X \setminus B$ existe $u \in B$ tal que nem $z \leq u$ nem $u \leq z$. Usamos $[T]$ para denotar o conjunto de todos os ramos de T .

Exemplo 2.35. Um exemplo clássico de árvore finitamente ramificada é a *árvore binária infinita* de sequências finitas de 0s e 1s, às vezes denotada $\{0, 1\}^*$ ou $\mathbb{B}^*$, ordenada pela relação de extensão $\sqsubseteq$ (p.ex., $101 \sqsubseteq 101101$). Como qualquer cadeia binária pode sempre ser estendida adicionando um 0 ou um 1 no final, esta árvore contém infinitos elementos: todo elemento s tem exatamente dois sucessores, $s0$ e $s1$. Sua raiz é a sequência vazia Λ .

Exemplo 2.36. De maneira um pouco mais geral, o conjunto de sequências finitas de números naturais $\mathbb{N}^*$ com a relação de extensão $\sqsubseteq$ também é uma árvore. É obviamente não finitamente ramificada: todo $s \in \mathbb{N}^*$ tem infinitos sucessores sn , um para cada $n \in \mathbb{N}$. Todo $A \subseteq \mathbb{N}^*$ que é fechado sob $\sqsubseteq$ é uma *subárvore* de $\mathbb{N}^*$. (Isto é, A é tal que se $s \in A$ e $s' \sqsubseteq s$, então também $s' \in A$.) Todas as árvores finitas podem ser representadas como subárvores finitas de $\mathbb{N}^*$.

Proposição 2.37 (Lema de König). Se $T = \langle A, \leq \rangle$ é uma árvore infinita finitamente ramificada, então T tem um ramo infinito.

Um caso especial do Lema de König amplamente usado em teoria da computabilidade, conhecido como *Lema Fraco de König*, é o seguinte: qualquer subárvore infinita de $\{0, 1\}^*$ tem um ramo infinito.

2.8 Operações sobre Relações

É frequentemente útil modificar ou combinar relações. Na **Proposição 2.25**, consideramos a *união* de relações, que é apenas a união de duas relações vistas como conjuntos de pares. Do mesmo modo, na **Proposição 2.26**, consideramos a diferença relativa de relações. Eis algumas outras operações que podemos efetuar sobre relações.

sfr:rel:ops:
sec:
relationoperations

Definição 2.38. Sejam R e S relações, e A um conjunto qualquer.

A *inversa* de R é $R^{-1} = \{\langle y, x \rangle : \langle x, y \rangle \in R\}$.

O *produto relativo* de R e S é $(R \mid S) = \{\langle x, z \rangle : \exists y(Rxy \wedge Syz)\}$.

A *restrição* de R a A é $R \upharpoonright_A = R \cap A^2$.

A *aplicação* de R a A é $R[A] = \{y : (\exists x \in A)Rxy\}$

Exemplo 2.39. Seja $S \subseteq \mathbb{Z}^2$ a relação sucessora sobre $\mathbb{Z}$, isto é, $S = \{\langle x, y \rangle \in \mathbb{Z}^2 : x + 1 = y\}$, de modo que Sxy sse $x + 1 = y$.

S^{-1} é a relação antecessora sobre $\mathbb{Z}$, isto é, $\{\langle x, y \rangle \in \mathbb{Z}^2 : x - 1 = y\}$.

$S \mid S$ é $\{\langle x, y \rangle \in \mathbb{Z}^2 : x + 2 = y\}$

$S \upharpoonright_{\mathbb{N}}$ é a relação sucessora sobre $\mathbb{N}$.

$S[\{1, 2, 3\}]$ é $\{2, 3, 4\}$.

Definição 2.40 (Fecho transitivo). Seja $R \subseteq A^2$ uma relação binária.

O *fecho transitivo* de R é $R^+ = \bigcup_{0 < n \in \mathbb{N}} R^n$, onde definimos recursivamente $R^1 = R$ e $R^{n+1} = R^n \mid R$.

O *fecho reflexivo transitivo* de R é $R^* = R^+ \cup \text{Id}_A$.

Exemplo 2.41. Tome a relação sucessora $S \subseteq \mathbb{Z}^2$. S^2xy sse $x + 2 = y$, S^3xy sse $x + 3 = y$, etc. Logo S^+xy sse $x + n = y$ para algum $n \geq 1$. Em outras palavras, S^+xy sse $x < y$, e S^*xy sse $x \leq y$.

Problema 2.6. Mostre que o fecho transitivo de R é de fato transitivo.

Capítulo 3

Funções

content/sets-functions-relations/functions/function-basics.tex

3.1 Noções Básicas

explanation Uma *função* é um mapa que envia cada **membro** de um dado conjunto para um **membro** específico em algum (outro) conjunto dado. Por exemplo, a operação de somar 1 define uma função: cada número n é mapeado para um único número $n + 1$. sfr:fun:bas:sec

Mais em geral, funções podem tomar pares, triplos, etc., como entradas e devolver algum tipo de saída. Muitas funções nos são familiares da aritmética básica. Por exemplo, a adição e a multiplicação são funções. Tomam dois números e devolvem um terceiro.

Neste sentido matemático e abstrato, uma função é uma *caixa preta*: só importa que saída fica associada a que entrada, não o método de calcular a saída.

Definição 3.1 (Função). Uma *função* $f: A \rightarrow B$ é um mapeamento de cada **membro** de A para um **membro** de B .

Chamamos A de *domínio* de f e B de *contradomínio* de f . Os **membros** de A chamam-se entradas ou *argumentos* de f , e o **membro** de B que fica associado a um argumento x por f chama-se o *valor de f* para o argumento x , escrevendo-se $f(x)$.

A *imagem* $\text{Im}(f)$ de f é o subconjunto do contradomínio constituído pelos valores de f para algum argumento; $\text{Im}(f) = \{f(x) : x \in A\}$.

O diagrama na **Figura 3.1** pode ajudar a pensar em funções. A elipse à esquerda representa o *domínio* da função; a elipse à direita representa o *contradomínio*; e uma seta aponta de um *argumento* no domínio para o *valor* correspondente no contradomínio.

3.1. NOÇÕES BÁSICAS

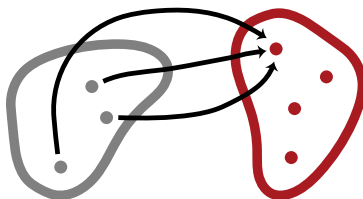


Figura 3.1: Uma função é um mapeamento de cada **membro** de um conjunto para **um membro** de outro. Uma seta vai de um argumento no domínio para o valor correspondente no contradomínio.

sfr:fun:bas:
fig:function

Exemplo 3.2. A multiplicação toma pares de números naturais como entradas e mapeia-os para números naturais como saídas, logo vai de $\mathbb{N} \times \mathbb{N}$ (o domínio) para $\mathbb{N}$ (o contradomínio). Como se vê, a imagem é também $\mathbb{N}$, pois todo $n \in \mathbb{N}$ é $n \times 1$.

Exemplo 3.3. A multiplicação é uma função porque associa cada entrada—cada par de números naturais—a uma única saída: $\times: \mathbb{N}^2 \rightarrow \mathbb{N}$. Em contraste, mapear um número natural n para um real x tal que $x^2 = n$ não é funcional, pois cada inteiro positivo n tem duas raízes quadradas: $\sqrt{n}$ e $-\sqrt{n}$. Podemos torná-la funcional devolvendo só a raiz quadrada positiva: $\sqrt{\cdot}: \mathbb{N} \rightarrow \mathbb{R}$.

Exemplo 3.4. A relação que associa cada aluno de uma turma à sua nota final é uma função—nenhum aluno pode ter duas notas finais diferentes na mesma turma. A relação que associa cada aluno de uma turma aos seus progenitores não é uma função: os alunos podem ter zero, ou dois, ou mais progenitores.

Podemos definir funções especificando de modo preciso qual é o valor da função para cada argumento possível. Formas de o fazer são dar uma fórmula, descrever um método para calcular o valor, ou listar os valores para cada argumento. Como quer que as funções sejam definidas, temos de garantir que, para cada argumento, especificamos um, e apenas um, valor.

explanation

Exemplo 3.5. Seja $f: \mathbb{N} \rightarrow \mathbb{N}$ definida por $f(x) = x + 1$. Esta é uma definição que especifica f como uma função que toma números naturais e devolve números naturais. Diz-nos que, dado um número natural x , f devolverá o seu sucessor $x + 1$. Neste caso, o contradomínio $\mathbb{N}$ não é a imagem de f , pois o número natural 0 não é sucessor de nenhum número natural. A imagem de f é o conjunto de todos os inteiros positivos, $\mathbb{Z}^+$.

sfr:fun:bas:
examplefunext

Exemplo 3.6. Seja $g: \mathbb{N} \rightarrow \mathbb{N}$ definida por $g(x) = x + 2 - 1$. Isto nos diz que g é uma função que toma números naturais e devolve números naturais. Dado um número natural n , g devolverá o predecessor do sucessor do sucessor de n , isto é, $n + 1$.

Acabamos de considerar duas funções, f e g , com *definições* diferentes. Contudo, são a *mesma função*. Afinal, para qualquer número natural n , temos

explanation

$f(n) = n + 1 = n + 2 - 1 = g(n)$. Em outras palavras: as nossas definições de f e g especificam o mesmo mapeamento por meio de equações diferentes. Implicitamente, pois, estamos nos apoiando num princípio de extensionalidade para funções,

$$\text{se } \forall x f(x) = g(x), \text{ então } f = g$$

desde que f e g compartilhem o mesmo domínio e contradomínio.

Exemplo 3.7. Também podemos definir funções por casos. Por exemplo, poderíamos definir $h: \mathbb{N} \rightarrow \mathbb{N}$ por

$$h(x) = \begin{cases} \frac{x}{2} & \text{se } x \text{ é par} \\ \frac{x+1}{2} & \text{se } x \text{ é ímpar.} \end{cases}$$

Como todo número natural é par ou ímpar, a saída desta função será sempre um número natural. Basta lembrar que, se se define uma função por casos, cada entrada possível tem de cair em exatamente um caso. Em alguns casos, isto exigirá uma prova de que os casos são exaustivos e exclusivos.

`content/sets-functions-relations/functions/function-kinds.tex`

3.2 Tipos de Funções

explanation Será útil introduzir uma espécie de taxonomia para alguns dos tipos de funções sfr:fun:kin:sec que encontramos com mais frequência.

Para começar, podemos querer considerar funções com a propriedade de que todo *membro* do contradomínio é um valor da função. Tais funções chamam-se **sobrejetivas**, e podem ser esquematizadas como na **Figura 3.2**.

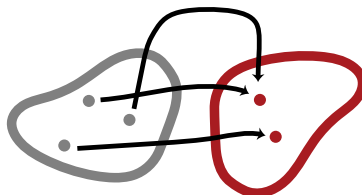


Figura 3.2: Uma função **sobrejetiva** tem todo **membro** do contradomínio como valor.

Definição 3.8 (Função Sobrejetiva). Uma função $f: A \rightarrow B$ é **sobrejetiva** sfr:fun:kin:fig:surjective se e só se B é também a imagem de f , isto é, para todo $y \in B$ existe pelo menos um $x \in A$ tal que $f(x) = y$, ou em símbolos:

$$(\forall y \in B)(\exists x \in A)f(x) = y.$$

Chamamos a tal função **uma sobrejeção** de A para B .

3.2. TIPOS DE FUNÇÕES

Se quiser mostrar que f é uma **sobrejeção**, tem de mostrar que todo objeto no contradomínio de f é o valor de $f(x)$ para alguma entrada x . explanation

Note-se que qualquer função *induz* uma **sobrejeção**. Afinal, dada uma função $f: A \rightarrow B$, seja $f': A \rightarrow \text{Im}(f)$ definida por $f'(x) = f(x)$. Como $\text{Im}(f)$ é *definida* como $\{f(x) \in B : x \in A\}$, esta função f' é garantidamente uma **sobrejeção**.

Agora, qualquer função mapeia cada entrada possível para uma única saída. Mas há também funções que nunca mapeiam entradas diferentes para a mesma saída. Tais funções chamam-se **injetivas**, e podem ser esquematizadas como na **Figura 3.3**. explanation



Figura 3.3: Uma função **injetiva** nunca mapeia dois argumentos diferentes para o mesmo valor.

sfr:fun:kin:
fig:injective

Definição 3.9 (Função Injetiva). Uma função $f: A \rightarrow B$ é **injetiva** se e só se, para cada $y \in B$, há no máximo um $x \in A$ tal que $f(x) = y$. Chamamos a tal função uma **injeção** de A para B .

Se quiser mostrar que f é uma **injeção**, tem de mostrar que, para quaisquer **membros** x e y do domínio de f , se $f(x) = f(y)$, então $x = y$. explanation

Exemplo 3.10. A função constante $f: \mathbb{N} \rightarrow \mathbb{N}$ dada por $f(x) = 1$ não é nem **injetiva**, nem **sobrejetiva**.

A função identidade $f: \mathbb{N} \rightarrow \mathbb{N}$ dada por $f(x) = x$ é tanto **injetiva** quanto **sobrejetiva**.

A função sucessora $f: \mathbb{N} \rightarrow \mathbb{N}$ dada por $f(x) = x + 1$ é **injetiva** mas não **sobrejetiva**.

A função $f: \mathbb{N} \rightarrow \mathbb{N}$ definida por:

$$f(x) = \begin{cases} \frac{x}{2} & \text{se } x \text{ é par} \\ \frac{x+1}{2} & \text{se } x \text{ é ímpar.} \end{cases}$$

é **sobrejetiva**, mas não **injetiva**.

Muitas vezes, queremos considerar funções que são tanto **injetivas** como **sobrejetivas**. Chamamos a tais funções **bijetivas**. Parecem-se com a função esquematizada na **Figura 3.4**. **Bijeções** são também por vezes chamadas *correspondências biunívocas*, pois associam de modo único elementos do contradomínio a elementos do domínio. explanation

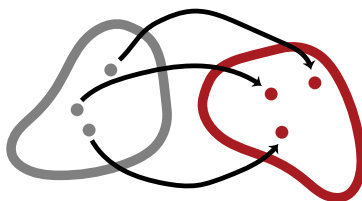


Figura 3.4: Uma função **bijetiva** associa de modo único os elementos do contradomínio aos do domínio.

Definição 3.11 (Bijecção). Uma função $f: A \rightarrow B$ é **bijetiva** se e só se é tanto **sobrejetiva** como **injetiva**. Chamamos a tal função **uma bijecção** de A para B (ou entre A e B).

sfr:fun:kin:
fig:bijjective

content/sets-functions-relations/functions/functions-relations.tex

3.3 Funções como Relações

explanation Uma função que mapeia **membros** de A para **membros** de B define obviamente uma relação entre A e B , nomeadamente a relação que se verifica entre x e y se e só se $f(x) = y$. De fato, podemos até—se nos interessa reduzir os blocos de construção da matemática, por exemplo—*identificar* a função f com esta relação, isto é, com um conjunto de pares. Isto levanta então a questão: que relações definem funções deste modo?

sfr:fun:rel:
sec

Definição 3.12 (Gráfico de uma função). Seja $f: A \rightarrow B$ uma função. O **gráfico** de f é a relação $R_f \subseteq A \times B$ definida por

$$R_f = \{\langle x, y \rangle : f(x) = y\}.$$

explanation O gráfico de uma função fica unicamente determinado, pela extensionalidade. Além disso, a extensionalidade (sobre conjuntos) justifica de imediato o princípio implícito de extensionalidade para funções, segundo o qual, se f e g compartilham domínio e contradomínio, então são idênticas se coincidem em todos os valores.

Do mesmo modo, se uma relação é “funcional”, então é o gráfico de uma função.

Proposição 3.13. *Seja $R \subseteq A \times B$ tal que:*

sfr:fun:rel:
prop:graph-function

1. *Se Rxy e Rxz , então $y = z$; e*
2. *para todo $x \in A$ existe algum $y \in B$ tal que $\langle x, y \rangle \in R$.*

Então R é o gráfico da função $f: A \rightarrow B$ definida por $f(x) = y$ se e só se Rxy .

3.4. INVERSAS DE FUNÇÕES

Prova. Suponhamos que existe y tal que Rxy . Se existisse outro $z \neq y$ tal que Rxz , a condição sobre R seria violada. Logo, se existe y tal que Rxy , esse y é único, e assim f está bem definida. Obviamente, $R_f = R$. $\square$

Toda função $f: A \rightarrow B$ tem um gráfico, isto é, uma relação em $A \times B$ definida por $f(x) = y$. Por outro lado, toda relação $R \subseteq A \times B$ com as propriedades dadas na **Proposição 3.13** é o gráfico de uma função $f: A \rightarrow B$. Por causa desta ligação estreita entre funções e os seus gráficos, podemos pensar numa função simplesmente como o seu gráfico. Em outras palavras, funções podem identificar-se com certas relações, isto é, com certos conjuntos de tuplos. Note, contudo, que o espírito desta “identificação” é como na **Seção 2.2**: não é uma afirmação sobre a metafísica das funções, mas a observação de que é conveniente *tratar* funções como certos conjuntos. Uma razão pela qual isto é tão conveniente é que podemos agora considerar fazer operações semelhantes sobre funções às que fizemos sobre relações (ver **Seção 2.8**). Em particular:

sfr:fun:rel:
def:funimage

Definição 3.14. Seja $f: A \rightarrow B$ uma função com $C \subseteq A$.

A *restrição* de f a C é a função $f|_C: C \rightarrow B$ definida por $(f|_C)(x) = f(x)$ para todo $x \in C$. Em outras palavras, $f|_C = \{\langle x, y \rangle \in R_f : x \in C\}$.

A *aplicação* de f a C é $f[C] = \{f(x) : x \in C\}$. Chamamos-lhe também a *imagem* de C por f .

Segue-se destas definições que $\text{Im}(f) = f[\text{dom}(f)]$, para qualquer função f . Estas noções são exatamente as que se esperaria, dadas as definições na **Seção 2.8** e a nossa identificação de funções com relações. Mas outras duas operações—*inversas* e *produtos relativos*—exigem um pouco mais de detalhe. Trataremos disso na **Seção 3.4** e na **Seção 3.5**.

content/sets-functions-relations/functions/inverses.tex

3.4 Inversas de Funções

sfr:fun:inv:
sec

Pensamos nas funções como mapas. Uma pergunta natural sobre funções é, pois, se o mapeamento pode ser “invertido”. Por exemplo, a função sucessora $f(x) = x + 1$ pode ser invertida, no sentido em que a função $g(y) = y - 1$ “desfaz” o que f faz.

Mas é preciso ter cuidado. Embora a definição de g defina uma função $\mathbb{Z} \rightarrow \mathbb{Z}$, não define uma *função* $\mathbb{N} \rightarrow \mathbb{N}$, pois $g(0) \notin \mathbb{N}$. Assim, mesmo em casos simples, não é óbvio que uma função possa ser invertida; pode depender do domínio e do contradomínio.

Isto torna-se mais preciso com a noção de inversa de uma função.

Definição 3.15. Uma função $g: B \rightarrow A$ é uma *inversa* de uma função $f: A \rightarrow B$ se $f(g(y)) = y$ e $g(f(x)) = x$ para todo $x \in A$ e $y \in B$.

Se f tem uma inversa g , escrevemos muitas vezes f^{-1} em vez de g .

explanation

Agora determinaremos quando as funções têm inversas. Um bom candidato a inversa de $f: A \rightarrow B$ é $g: B \rightarrow A$ “definida por”

$$g(y) = \text{“o” } x \text{ tal que } f(x) = y.$$

Mas as aspas em “definida por” (e em “o”) sugerem que isto não é uma definição. Pelo menos, nem sempre funcionará com total generalidade. Pois, para que esta definição especifique uma função, tem de haver um e apenas um x tal que $f(x) = y$ —a saída de g tem de ficar unicamente determinada. Além disso, tem de estar especificada para todo $y \in B$. Se existirem x_1 e $x_2 \in A$ com $x_1 \neq x_2$ mas $f(x_1) = f(x_2)$, então $g(y)$ não ficaria unicamente determinada para $y = f(x_1) = f(x_2)$. E se não existir x algum tal que $f(x) = y$, então $g(y)$ nem sequer fica especificada. Em outras palavras, para g estar definida, f tem de ser tanto **injetiva** quanto **sobrejetiva**.

Vamos com calma. Dividiremos a questão em duas: dada uma função $f: A \rightarrow B$, quando existe uma função $g: B \rightarrow A$ tal que $g(f(x)) = x$? Tal g “desfaz” o que f faz, e chama-se *inversa à esquerda* de f . Em segundo lugar, quando existe uma função $h: B \rightarrow A$ tal que $f(h(y)) = y$? Tal h chama-se *inversa à direita* de f — f “desfaz” o que h faz.

Proposição 3.16. *Se $f: A \rightarrow B$ é **injetiva**, então existe uma inversa à esquerda $g: B \rightarrow A$ de f tal que $g(f(x)) = x$ para todo $x \in A$.*

Prova. Suponhamos que $f: A \rightarrow B$ é **injetiva**. Consideremos $y \in B$. Se $y \in \text{Im}(f)$, existe $x \in A$ tal que $f(x) = y$. Como f é **injetiva**, há apenas um tal $x \in A$. Então podemos definir: $g(y) = x$, isto é, $g(y)$ é “o” $x \in A$ tal que $f(x) = y$. Se $y \notin \text{Im}(f)$, podemos mapeá-lo para qualquer $a \in A$. Assim, podemos escolher $a \in A$ e definir $g: B \rightarrow A$ por:

$$g(y) = \begin{cases} x & \text{se } f(x) = y \\ a & \text{se } y \notin \text{Im}(f). \end{cases}$$

Está definida para todo $y \in B$, pois para cada tal $y \in \text{Im}(f)$ há exatamente um $x \in A$ tal que $f(x) = y$. Por definição, se $y = f(x)$, então $g(y) = x$, isto é, $g(f(x)) = x$. $\square$

Problema 3.1. Mostre que, se $f: A \rightarrow B$ tem uma inversa à esquerda g , então f é **injetiva**.

Proposição 3.17. *Se $f: A \rightarrow B$ é **sobrejetiva**, então existe uma inversa à direita $h: B \rightarrow A$ de f tal que $f(h(y)) = y$ para todo $y \in B$.*

Prova. Suponhamos que $f: A \rightarrow B$ é **sobrejetiva**. Consideremos $y \in B$. Como f é **sobrejetiva**, existe $x_y \in A$ com $f(x_y) = y$. Então podemos definir: $h(y) = x_y$, isto é, para cada $y \in B$ escolhemos algum $x \in A$ tal que $f(x) = y$; como

3.4. INVERSAS DE FUNÇÕES

f é **sobrejetiva** há sempre pelo menos um de onde escolher.¹ Por definição, se $x = h(y)$, então $f(x) = y$, isto é, para todo $y \in B$, $f(h(y)) = y$. $\square$

Problema 3.2. Mostre que, se $f: A \rightarrow B$ tem uma inversa à direita h , então f é **sobrejetiva**.

Combinando as ideias da prova anterior, obtemos agora que toda **bijeção** explanation tem uma inversa, isto é, existe uma única função que é tanto inversa à esquerda quanto à direita de f .

sfr:fun:inv:
prop:bijection-inverse **Proposição 3.18.** Se $f: A \rightarrow B$ é **bijetiva**, existe uma função $f^{-1}: B \rightarrow A$ tal que, para todo $x \in A$, $f^{-1}(f(x)) = x$ e, para todo $y \in B$, $f(f^{-1}(y)) = y$.

Prova. Exercício. $\square$

Problema 3.3. Prove **Proposição 3.18**. Tem de definir f^{-1} , mostrar que é uma função, e mostrar que é uma inversa de f , isto é, $f^{-1}(f(x)) = x$ e $f(f^{-1}(y)) = y$ para todo $x \in A$ e $y \in B$.

Há uma forma ligeiramente mais geral de extrair inversas. Vimos na **Seção 3.2** explanation que toda função f induz uma **sobrejeção** $f': A \rightarrow \text{Im}(f)$ pondo $f'(x) = f(x)$ para todo $x \in A$. Claramente, se f é **injetiva**, então f' é **bijetiva**, logo tem uma inversa única pela **Proposição 3.18**. Por um pequeno abuso de notação, chamamos por vezes à inversa de f' simplesmente “a inversa de f .”

sfr:fun:inv:
prop:left-right **Proposição 3.19.** Mostre que, se $f: A \rightarrow B$ tem uma inversa à esquerda g e uma inversa à direita h , então $h = g$.

Prova. Exercício. $\square$

Problema 3.4. Prove **Proposição 3.19**.

sfr:fun:inv:
prop:inverse-unique **Proposição 3.20.** Toda função f tem no máximo uma inversa.

Prova. Suponhamos que g e h são ambas inversas de f . Então, em particular, g é uma inversa à esquerda de f e h é uma inversa à direita. Pela **Proposição 3.19**, $g = h$. $\square$

content/sets-functions-relations/functions/composition.tex

¹Como f é **sobrejetiva**, para todo $y \in B$ o conjunto $\{x : f(x) = y\}$ é não vazio. A nossa definição de h exige que escolhamos um único x em cada um destes conjuntos. Que isto seja sempre possível não é de todo óbvio—a possibilidade de fazer estas escolhas é simplesmente assumida como axioma. Em outras palavras, esta proposição pressupõe o chamado Axioma da Escolha, questão a que voltaremos no **Capítulo 73**. Contudo, em muitos casos concretos, p.ex., quando $A = \mathbb{N}$ ou é finito, ou quando f é **bijetiva**, o Axioma da Escolha não é necessário. (No caso em que f é **bijetiva**, para cada $y \in B$ o conjunto $\{x : f(x) = y\}$ tem exatamente um **membro**, logo não há escolha a fazer.)

3.5 Composição de Funções

explanation Vimos na [Seção 3.4](#) que a inversa f^{-1} de uma [bijeção](#) f é ela própria uma função. Outra operação sobre funções é a composição: podemos definir uma nova função compondo duas funções, f e $sfr:fun:cmp:sec$ g , isto é, aplicando primeiro f e depois g . Claro que isto só é possível se as imagens e os domínios forem compatíveis, isto é, a imagem de f tem de ser um subconjunto do domínio de g . Esta operação sobre funções é o análogo da operação de produto relativo sobre relações na [Seção 2.8](#).

Um diagrama pode ajudar a explicar a ideia de composição. Na [Figura 3.5](#), representamos duas funções $f: A \rightarrow B$ e $g: B \rightarrow C$ e a sua composição $(g \circ f)$. A função $(g \circ f): A \rightarrow C$ associa cada [membro](#) de A a um [membro](#) de C . Especificamos a que [membro](#) de C fica associado um [membro](#) de A do seguinte modo: dada uma entrada $x \in A$, aplicamos primeiro a função f a x , obtendo algum $f(x) = y \in B$, depois aplicamos a função g a y , obtendo algum $g(f(x)) = g(y) = z \in C$.

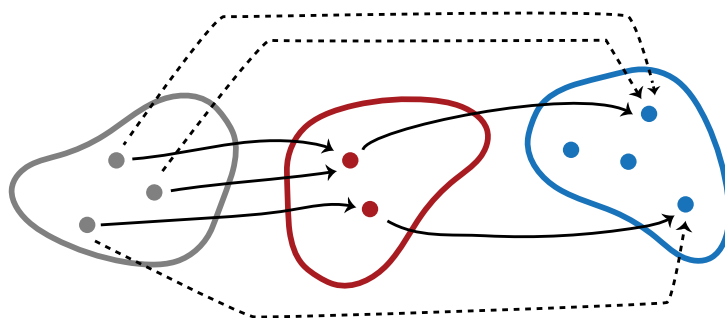


Figura 3.5: A composição $g \circ f$ de duas funções f e g .

Definição 3.21 (Composição). Sejam $f: A \rightarrow B$ e $g: B \rightarrow C$ funções. A [sfr:fun:cmp:fig:composition](#) *composição* de f com g é $g \circ f: A \rightarrow C$, onde $(g \circ f)(x) = g(f(x))$.

Exemplo 3.22. Consideremos as funções $f(x) = x + 1$, e $g(x) = 2x$. Como $(g \circ f)(x) = g(f(x))$, para cada entrada x deve primeiro tomar-se o sucessor, depois multiplicar o resultado por dois. Logo a composição é dada por $(g \circ f)(x) = 2(x + 1)$.

Problema 3.5. Mostre que, se $f: A \rightarrow B$ e $g: B \rightarrow C$ são ambas [injetiva](#), então $g \circ f: A \rightarrow C$ é [injetiva](#).

Problema 3.6. Mostre que, se $f: A \rightarrow B$ e $g: B \rightarrow C$ são ambas [sobrejetiva](#), então $g \circ f: A \rightarrow C$ é [sobrejetiva](#).

Problema 3.7. Suponhamos $f: A \rightarrow B$ e $g: B \rightarrow C$. Mostre que o gráfico de $g \circ f$ é $R_f \mid R_g$.

3.6 Funções Parciais

Às vezes é útil relaxar a definição de função de modo que não seja exigido que a saída da função esteja definida para todas as entradas possíveis. Tais mapeamentos são chamados de *funções parciais*.

Definição 3.23. Uma *função parcial* $f: A \rightarrow B$ é um mapeamento que associa a cada **membro** de A no máximo um **membro** de B . Se f associa um **membro** de B a $x \in A$, dizemos que $f(x)$ está *definido*, e caso contrário *indefinido*. Se $f(x)$ está definido, escrevemos $f(x) \downarrow$, caso contrário $f(x) \uparrow$. O *domínio* de uma função parcial f é o subconjunto de A onde ela está definida, isto é, $\text{dom}(f) = \{x \in A : f(x) \downarrow\}$.

Exemplo 3.24. Toda função $f: A \rightarrow B$ é também uma função parcial. Funções parciais que estão definidas em todo A —isto é, o que até agora chamávamos simplesmente de função—também são chamadas de funções *totais*.

Exemplo 3.25. A função parcial $f: \mathbb{R} \rightarrow \mathbb{R}$ dada por $f(x) = 1/x$ é indefinida para $x = 0$ e definida em todo o restante.

Problema 3.8. Dada $f: A \rightarrow B$, defina a função parcial $g: B \rightarrow A$ por: para qualquer $y \in B$, se existe um único $x \in A$ tal que $f(x) = y$, então $g(y) = x$; caso contrário $g(y) \uparrow$. Mostre que se f é injetiva, então $g(f(x)) = x$ para todo $x \in \text{dom}(f)$ e $f(g(y)) = y$ para todo $y \in \text{Im}(f)$.

Definição 3.26 (Gráfico de uma função parcial). Seja $f: A \rightarrow B$ uma função parcial. O *gráfico* de f é a relação $R_f \subseteq A \times B$ definida por

$$R_f = \{\langle x, y \rangle : f(x) = y\}.$$

Proposição 3.27. Suponha que $R \subseteq A \times B$ tem a propriedade de que sempre que Rxy e Rxy' então $y = y'$. Então R é o gráfico da função parcial $f: A \rightarrow B$ definida por: se existe um y tal que Rxy , então $f(x) = y$, caso contrário $f(x) \uparrow$. Se R também é serial, isto é, para cada $x \in A$ existe um $y \in B$ tal que Rxy , então f é total.

Prova. Suponha que existe um y tal que Rxy . Se existisse outro $y' \neq y$ tal que Rxy' , a condição sobre R seria violada. Portanto, se existe um y tal que Rxy , esse y é único, e assim f é bem definida. Obviamente, $R_f = R$ e f é total se R é serial. $\square$

Capítulo 4

O Tamanho dos Conjuntos

Este capítulo discute enumerações e conjuntos contáveis e incontáveis. Várias seções vêm em duas versões: uma mais elementar, que toma enumerações como listas, ou sobrejeções de $\mathbb{Z}^+$; e uma mais abstrata que define enumerações como bijeções com $\mathbb{N}$.

`content/sets-functions-relations/size-of-sets/introduction.tex`

4.1 Introdução

Quando Georg Cantor desenvolveu a teoria dos conjuntos na década de 1870, um dos seus objetivos era tornar aceitável a ideia de uma coleção infinita—um infinito *atual*, como diriam os medievais. Parte central disto foi o seu tratamento do *tamanho* de conjuntos diferentes. Se a , b e c são todos distintos, então o conjunto $\{a, b, c\}$ é intuitivamente *maior* que $\{a, b\}$. Mas e os conjuntos infinitos? Serão todos do mesmo tamanho? Acontece que não.

A primeira ideia importante aqui é a de *enumeração*. Podemos listar qualquer conjunto finito enumerando todos os seus **membros**. Para alguns conjuntos infinitos, também podemos listar todos os seus **membros** se permitirmos que a própria lista seja infinita. Tais conjuntos chamam-se **enumeráveis**. O resultado surpreendente de Cantor, que compreenderemos completamente no fim deste capítulo, foi que alguns conjuntos infinitos *não* são **enumeráveis**.

`content/sets-functions-relations/size-of-sets/enumerability.tex`

4.2 Enumerações e Conjuntos Enumeráveis

`sfr:siz:enm:
sec`

4.2. ENUMERAÇÕES E CONJUNTOS ENUMERÁVEIS

Esta seção discute enumerações de conjuntos, definindo-as como sobrejeções de $\mathbb{Z}^+$. Faz as coisas devagar, para leitores com pouca formação matemática. Uma versão alternativa, mais concisa, é dada na [Seção 4.11](#), que define enumerações de outro modo: como bijeções com $\mathbb{N}$ (ou um segmento inicial).

Já demos exemplos de conjuntos listando seus **membros**. Discutamos agora, de maneira mais geral, como e quando podemos listar os **membros** de um conjunto, mesmo que esse conjunto seja infinito. explanation

Definição 4.1 (Enumeração, informalmente). Informalmente, uma *enumeração* de um conjunto A é uma lista (possivelmente infinita) de **membros** de A tal que todo **membro** de A aparece na lista em alguma posição finita. Se A tem uma enumeração, então diz-se que A é *enumerável*.

Alguns pontos sobre enumerações: explanation

1. Contamos como enumerações apenas listas que têm um começo e em que todo **membro** além do primeiro tem um único **membro** imediatamente anterior. Em outras palavras, há apenas um número finito de elementos entre o primeiro **membro** da lista e qualquer outro **membro**. Em particular, isso significa que todo **membro** de uma enumeração tem uma posição finita: o primeiro **membro** tem posição 1, o segundo posição 2, etc.
2. Podemos ter enumerações diferentes do mesmo conjunto A que diferem pela ordem em que os **membros** aparecem: 4, 1, 25, 16, 9 enumera os (conjunto dos) cinco primeiros quadrados tão bem quanto 1, 4, 9, 16, 25.
3. Enumerações redundantes ainda são enumerações: 1, 1, 2, 2, 3, 3, ... enumera o mesmo conjunto que 1, 2, 3, ...
4. Ordem e redundância *importam* quando especificamos uma enumeração: podemos enumerar os inteiros positivos começando com 1, 2, 3, 1, ..., mas o padrão é mais fácil de ver quando enumerados da maneira padrão como 1, 2, 3, 4, ...
5. Enumerações devem ter um começo: ..., 3, 2, 1 não é uma enumeração dos inteiros positivos porque não tem primeiro **membro**. Para ver como isso segue da definição informal, pergunte a si mesmo: “em que posição na lista o número 76 aparece?”
6. O seguinte não é uma enumeração dos inteiros positivos: 1, 3, 5, ..., 2, 4, 6, ... O problema é que os números pares ocorrem nas posições $\infty + 1$, $\infty + 2$, $\infty + 3$, em vez de em posições finitas.
7. O conjunto vazio é **enumerável**: é enumerado pela lista vazia!

Proposição 4.2. *Se A tem uma enumeração, tem uma enumeração sem repetições.*

Prova. Suponha que A tem uma enumeração $x_1, x_2, \dots$ em que cada x_i é um **membro** de A . Podemos remover repetições de uma enumeração removendo **membros** repetidos. Por exemplo, podemos transformar a enumeração em uma nova na qual listamos x_i se ele é um **um membro** de A que não está entre $x_1, \dots, x_{i-1}$ ou removemos x_i da lista se já aparece entre $x_1, \dots, x_{i-1}$. $\square$

O último argumento mostra que, para obter um bom controle sobre enumerações e conjuntos **enumeráveis** e provar coisas sobre eles, precisamos de uma definição mais precisa. O seguinte a fornece.

Definição 4.3 (Enumeração, formalmente). Uma *enumeração* de um conjunto $A \neq \emptyset$ é qualquer função **sobrejetiva** $f: \mathbb{Z}^+ \rightarrow A$.

explanation

Convençamo-nos de que a definição formal e a definição informal usando uma lista possivelmente infinita são equivalentes. Primeiro, qualquer função **sobrejetiva** de $\mathbb{Z}^+$ para um conjunto A enumera A . Tal função determina uma enumeração conforme definido informalmente acima: a lista $f(1), f(2), f(3), \dots$. Como f é **sobrejetiva**, todo **membro** de A é garantidamente o valor de $f(n)$ para algum $n \in \mathbb{Z}^+$. Logo, todo **membro** de A aparece em alguma posição finita na lista. Como a função pode não ser **injetiva**, a lista pode ser redundante, mas isso é aceitável (como observado acima).

Por outro lado, dada uma lista que enumera todos os **membros** de A , podemos definir uma função **sobrejetiva** $f: \mathbb{Z}^+ \rightarrow A$ fazendo $f(n)$ ser o n -ésimo **membro** da lista, ou o último **membro** da lista se não há n -ésimo **membro**. O único caso em que isso não produz uma função **sobrejetiva** é quando A é vazio, e portanto a lista é vazia. Assim, toda lista não vazia determina uma função **sobrejetiva** $f: \mathbb{Z}^+ \rightarrow A$.

Definição 4.4. Um conjunto A é **enumerável** sse é vazio ou tem uma enumeração.

sfr:siz:enm:
defn:enumerable

Exemplo 4.5. Uma função que enumera os inteiros positivos ($\mathbb{Z}^+$) é simplesmente a função identidade dada por $f(n) = n$. Uma função que enumera os números naturais $\mathbb{N}$ é a função $g(n) = n - 1$.

Exemplo 4.6. As funções $f: \mathbb{Z}^+ \rightarrow \mathbb{Z}^+$ e $g: \mathbb{Z}^+ \rightarrow \mathbb{Z}^+$ dadas por

$$\begin{aligned} f(n) &= 2n \text{ e} \\ g(n) &= 2n - 1 \end{aligned}$$

enumeram os inteiros positivos pares e os inteiros positivos ímpares, respectivamente. Contudo, nenhuma das funções é uma enumeração de $\mathbb{Z}^+$, já que nenhuma é **sobrejetiva**.

4.2. ENUMERAÇÕES E CONJUNTOS ENUMERÁVEIS

Problema 4.1. Defina uma enumeração dos quadrados positivos 1, 4, 9, 16, ...

Exemplo 4.7. A função $f(n) = (-1)^n \lceil \frac{n-1}{2} \rceil$ (onde $\lceil x \rceil$ denota a função *teto*, que arredonda x para cima até o inteiro mais próximo) enumera o conjunto dos inteiros $\mathbb{Z}$. Note como f gera os valores de $\mathbb{Z}$ “saltando” de um lado para outro entre inteiros positivos e negativos:

$$\begin{array}{cccccccc} f(1) & f(2) & f(3) & f(4) & f(5) & f(6) & f(7) & \dots \\ -\lceil \frac{0}{2} \rceil & \lceil \frac{1}{2} \rceil & -\lceil \frac{2}{2} \rceil & \lceil \frac{3}{2} \rceil & -\lceil \frac{4}{2} \rceil & \lceil \frac{5}{2} \rceil & -\lceil \frac{6}{2} \rceil & \dots \\ 0 & 1 & -1 & 2 & -2 & 3 & \dots \end{array}$$

Você também pode pensar em f como definida por casos da seguinte forma:

$$f(n) = \begin{cases} 0 & \text{se } n = 1 \\ n/2 & \text{se } n \text{ é par} \\ -(n-1)/2 & \text{se } n \text{ é ímpar e } > 1 \end{cases}$$

Problema 4.2. Mostre que se A e B são **enumeráveis**, então $A \cup B$ também é. Para isso, suponha que há funções **sobrejetivas** $f: \mathbb{Z}^+ \rightarrow A$ e $g: \mathbb{Z}^+ \rightarrow B$, e defina uma função **sobrejetiva** $h: \mathbb{Z}^+ \rightarrow A \cup B$ e mostre que ela é **sobrejetiva**. Considere também os casos em que A ou $B = \emptyset$.

Problema 4.3. Mostre que se $B \subseteq A$ e A é **enumerável**, então B também é. Para isso, suponha que há uma função **sobrejetiva** $f: \mathbb{Z}^+ \rightarrow A$. Defina uma função **sobrejetiva** $g: \mathbb{Z}^+ \rightarrow B$ e mostre que ela é **sobrejetiva**. O que acontece se $B = \emptyset$?

Problema 4.4. Mostre por indução em n que se $A_1, A_2, \dots, A_n$ são todos **enumeráveis**, então $A_1 \cup \dots \cup A_n$ também é. Você pode assumir o fato de que se dois conjuntos A e B são **enumeráveis**, então $A \cup B$ também é.

Embora seja talvez mais natural ao listar os **membros** de um conjunto começar a contar a partir do 1º **membro**, os matemáticos gostam de usar os números naturais $\mathbb{N}$ para contar coisas. Eles falam do 0º, 1º, 2º, e assim por diante, **membro** de uma lista. Correspondentemente, podemos definir uma enumeração como uma função **sobrejetiva** de $\mathbb{N}$ para A . É claro que as duas definições são equivalentes.

sfr:siz:enm:
prop:enum-shift **Proposição 4.8.** *Há uma sobrejeção $f: \mathbb{Z}^+ \rightarrow A$ sse há uma sobrejeção $g: \mathbb{N} \rightarrow A$.*

Prova. Dada uma **sobrejeção** $f: \mathbb{Z}^+ \rightarrow A$, podemos definir $g(n) = f(n+1)$ para todo $n \in \mathbb{N}$. É fácil ver que $g: \mathbb{N} \rightarrow A$ é **sobrejetiva**. Reciprocamente, dada uma **sobrejeção** $g: \mathbb{N} \rightarrow A$, defina $f(n) = g(n-1)$. $\square$

CAPÍTULO 4. O TAMANHO DOS CONJUNTOS

Isso nos dá o seguinte resultado:

Corolário 4.9. *Um conjunto A é **enumerável** sse é vazio ou há uma função **sobrejetiva** $f: \mathbb{N} \rightarrow A$.*

sfr:siz:enm:
cor:enum-nat

Discutimos acima que uma lista de **membros** de um conjunto A pode ser transformada em uma lista sem repetições. Isso também vale para enumerações, mas é um pouco mais difícil de formular e provar rigorosamente. Qualquer função $f: \mathbb{Z}^+ \rightarrow A$ deve estar definida para todo $n \in \mathbb{Z}^+$. Se há apenas um número finito de **membros** em A , então claramente não podemos ter uma função definida nos infinitos **membros** de $\mathbb{Z}^+$ que toma como valores todos os **membros** de A mas nunca toma o mesmo valor duas vezes. Nesse caso, isto é, no caso em que a lista sem repetições é finita, devemos escolher um domínio diferente para f , um com apenas um número finito de **membros**. Não ter repetições significa que f deve ser **injetiva**. Como também é **sobrejetiva**, estamos procurando **uma bijeção** entre algum conjunto finito $\{1, \dots, n\}$ ou $\mathbb{Z}^+$ e A .

Proposição 4.10. *Se $f: \mathbb{Z}^+ \rightarrow A$ é **sobrejetiva** (isto é, uma enumeração de A), há **uma bijeção** $g: Z \rightarrow A$ onde Z é $\mathbb{Z}^+$ ou $\{1, \dots, n\}$ para algum $n \in \mathbb{Z}^+$.*

sfr:siz:enm:
prop:enum-bij

Prova. Definimos a função g recursivamente: seja $g(1) = f(1)$. Se $g(i)$ já foi definido, seja $g(i+1)$ o primeiro valor de $f(1), f(2), \dots$ que ainda não está entre $g(1), \dots, g(i)$, se houver um. Se A tem apenas n **membros**, então $g(1), \dots, g(n)$ estão todos definidos, e assim definimos uma função $g: \{1, \dots, n\} \rightarrow A$. Se A tem infinitos **membros**, então para qualquer i deve haver **um membro** de A na enumeração $f(1), f(2), \dots$, que ainda não está entre $g(1), \dots, g(i)$. Nesse caso definimos uma função $g: \mathbb{Z}^+ \rightarrow A$.

A função g é **sobrejetiva**, já que qualquer elemento de A está entre $f(1), f(2), \dots$ (pois f é **sobrejetiva**) e assim eventualmente será um valor de $g(i)$ para algum i . Também é **injetiva**, já que se houvesse $j < i$ tal que $g(j) = g(i)$, então $g(i)$ já estaria entre $g(1), \dots, g(i-1)$, contrariamente à forma como definimos g . □

Corolário 4.11. *Um conjunto A é **enumerável** sse é vazio ou há **uma bijeção** $f: N \rightarrow A$ onde $N = \mathbb{N}$ ou $N = \{0, \dots, n\}$ para algum $n \in \mathbb{N}$.*

sfr:siz:enm:
cor:enum-nat-bij

Prova. A é **enumerável** sse A é vazio ou há **uma sobrejetiva** $f: \mathbb{Z}^+ \rightarrow A$. Pela **Proposição 4.10**, o último vale sse há uma função **bijetiva** $f: Z \rightarrow A$ onde $Z = \mathbb{Z}^+$ ou $Z = \{1, \dots, n\}$ para algum $n \in \mathbb{Z}^+$. Pelo mesmo argumento da prova da **Proposição 4.8**, isso por sua vez é o caso sse há **uma bijeção** $g: N \rightarrow A$ onde $N = \mathbb{N}$ ou $N = \{0, \dots, n-1\}$. □

Problema 4.5. De acordo com a **Definição 4.4**, um conjunto A é **enumerável** sse $A = \emptyset$ ou há **uma sobrejetiva** $f: \mathbb{Z}^+ \rightarrow A$. Também é possível definir “conjunto **enumerável**” precisamente por: um conjunto é **enumerável** sse há uma função **injetiva** $g: A \rightarrow \mathbb{Z}^+$. Mostre que as definições são equivalentes, isto é,

4.3. MÉTODO DO ZIGUE-ZAGUE DE CANTOR

mostre que há uma função **injetiva** $g: A \rightarrow \mathbb{Z}^+$ sse $A = \emptyset$ ou há uma **sobrejetiva** $f: \mathbb{Z}^+ \rightarrow A$.

content/sets-functions-relations/size-of-sets/zig-zag.tex

4.3 Método do Zigue-Zague de Cantor

sfr:siz:zigzag:
sec

Já consideramos algumas enumerações “fáceis”. Vejamos algo um pouco mais difícil. Considere-se o conjunto dos pares de números naturais, que definimos na **Seção 1.5** assim:

explanation

$$\mathbb{N} \times \mathbb{N} = \{\langle n, m \rangle : n, m \in \mathbb{N}\}$$

Podemos organizar estes pares ordenados numa *matriz*, assim:

	0	1	2	3	...
0	$\langle 0, 0 \rangle$	$\langle 0, 1 \rangle$	$\langle 0, 2 \rangle$	$\langle 0, 3 \rangle$	...
1	$\langle 1, 0 \rangle$	$\langle 1, 1 \rangle$	$\langle 1, 2 \rangle$	$\langle 1, 3 \rangle$	...
2	$\langle 2, 0 \rangle$	$\langle 2, 1 \rangle$	$\langle 2, 2 \rangle$	$\langle 2, 3 \rangle$	...
3	$\langle 3, 0 \rangle$	$\langle 3, 1 \rangle$	$\langle 3, 2 \rangle$	$\langle 3, 3 \rangle$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Claramente, cada par ordenado em $\mathbb{N} \times \mathbb{N}$ aparece exatamente uma vez na matriz. Em particular, $\langle n, m \rangle$ aparece na n -ésima linha e na m -ésima coluna. Mas como organizar os elementos de tal matriz numa lista “unidimensional”? O padrão na matriz abaixo demonstra uma forma de o fazer (embora haja, claro, muitas outras):

	0	1	2	3	4	...
0	0	1	3	6	10	...
1	2	4	7	11	...	...
2	5	8	12	...	...	...
3	9	13	...	...	...	...
4	14	...	...	...	...	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\ddots$

Este padrão chama-se *método do zigue-zague de Cantor*. Enumera $\mathbb{N} \times \mathbb{N}$ do seguinte modo:

$$\langle 0, 0 \rangle, \langle 0, 1 \rangle, \langle 1, 0 \rangle, \langle 0, 2 \rangle, \langle 1, 1 \rangle, \langle 2, 0 \rangle, \langle 0, 3 \rangle, \langle 1, 2 \rangle, \langle 2, 1 \rangle, \langle 3, 0 \rangle, \dots$$

Isto estabelece o seguinte:

sfr:siz:zigzag:
natsquaredenumerable

Proposição 4.12. $\mathbb{N} \times \mathbb{N}$ é **enumerável**.

Prova. Seja $f: \mathbb{N} \rightarrow \mathbb{N} \times \mathbb{N}$ a função que a cada $k \in \mathbb{N}$ associa o tuplo $\langle n, m \rangle \in \mathbb{N} \times \mathbb{N}$ tal que k é o valor na n -ésima linha e m -ésima coluna da matriz do zigue-zague de Cantor. $\square$

explanation Esta técnica generaliza-se bem. Por exemplo, podemos usá-la para enumerar o conjunto dos triplos ordenados de números naturais, isto é:

$$\mathbb{N} \times \mathbb{N} \times \mathbb{N} = \{\langle n, m, k \rangle : n, m, k \in \mathbb{N}\}$$

Pensamos em $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$ como o produto cartesiano de $\mathbb{N} \times \mathbb{N}$ com $\mathbb{N}$, isto é,

$$\mathbb{N}^3 = (\mathbb{N} \times \mathbb{N}) \times \mathbb{N} = \{\langle \langle n, m \rangle, k \rangle : n, m, k \in \mathbb{N}\}$$

e assim podemos enumerar $\mathbb{N}^3$ com uma matriz rotulando um eixo com a enumeração de $\mathbb{N}$ e o outro com a enumeração de $\mathbb{N}^2$:

	0	1	2	3	...
$\langle 0, 0 \rangle$	$\langle 0, 0, 0 \rangle$	$\langle 0, 0, 1 \rangle$	$\langle 0, 0, 2 \rangle$	$\langle 0, 0, 3 \rangle$	...
$\langle 0, 1 \rangle$	$\langle 0, 1, 0 \rangle$	$\langle 0, 1, 1 \rangle$	$\langle 0, 1, 2 \rangle$	$\langle 0, 1, 3 \rangle$	...
$\langle 1, 0 \rangle$	$\langle 1, 0, 0 \rangle$	$\langle 1, 0, 1 \rangle$	$\langle 1, 0, 2 \rangle$	$\langle 1, 0, 3 \rangle$	...
$\langle 0, 2 \rangle$	$\langle 0, 2, 0 \rangle$	$\langle 0, 2, 1 \rangle$	$\langle 0, 2, 2 \rangle$	$\langle 0, 2, 3 \rangle$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Assim, com um método como o zigue-zague de Cantor, obtemos analogamente uma enumeração de $\mathbb{N}^3$. E podemos continuar, obtendo enumerações de $\mathbb{N}^n$ para qualquer número natural n . Logo:

Proposição 4.13. $\mathbb{N}^n$ é *enumerável*, para todo $n \in \mathbb{N}$.

Problema 4.6. Mostre que $(\mathbb{Z}^+)^n$ é *enumerável*, para todo $n \in \mathbb{N}$.

Problema 4.7. Mostre que $(\mathbb{Z}^+)^*$ é *enumerável*. Pode assumir **Problema 4.6**.

`content/sets-functions-relations/size-of-sets/pairing.tex`

4.4 Funções de Pareamento e Códigos

explanation O método do zigue-zague de Cantor torna visualmente evidente a enumerabilidade de $\mathbb{N}^n$. Mas vamos concentrar-nos na nossa matriz que representa $\mathbb{N}^2$. Seguindo a linha em zigue-zague na matriz e contando as posições, podemos verificar que $\langle 1, 2 \rangle$ fica associado ao número 7. Seria útil, contudo, poder calcular isto de modo mais direto. Ou seja, seria útil dispor da *inversa* da enumeração em zigue-zague, $g: \mathbb{N}^2 \rightarrow \mathbb{N}$, tal que

$$g(\langle 0, 0 \rangle) = 0, g(\langle 0, 1 \rangle) = 1, g(\langle 1, 0 \rangle) = 2, \dots, g(\langle 1, 2 \rangle) = 7, \dots$$

Isto permitir-nos-ia calcular exatamente em que posição da enumeração cai $\langle n, m \rangle$.

De fato, podemos definir g diretamente com base em duas observações. Primeiro: se a n -ésima linha e a m -ésima coluna contêm o valor v , então a $(n+1)$ -ésima linha e a $(m-1)$ -ésima coluna contêm o valor $v+1$. Segundo: a primeira

4.4. FUNÇÕES DE PAREAMENTO E CÓDIGOS

linha da nossa enumeração consiste nos números triangulares, começando por 0, 1, 3, 6, etc. O k -ésimo número triangular é a soma dos números naturais $< k$, que se pode calcular como $k(k+1)/2$. Juntando estas duas observações, consideremos a função:

$$g(n, m) = \frac{(n+m+1)(n+m)}{2} + n$$

Escreve-se frequentemente $g(n, m)$ em vez de $g(\langle n, m \rangle)$, já que é mais fácil para os olhos. Isto diz-nos que se determine primeiro o $(n+m)$ -ésimo número triangular e depois que lhe somemos n . A função preenche a matriz exatamente da maneira que desejamos. Em particular, o par $\langle 1, 2 \rangle$ é enviado para $\frac{4 \times 3}{2} + 1 = 7$.

Esta função g é a *inversa* de uma enumeração de um conjunto de pares. Tais funções chamam-se *funções de emparelhamento*.

Definição 4.14 (Função de emparelhamento). Uma função $f: A \times B \rightarrow \mathbb{N}$ é uma *função de emparelhamento* aritmética se f for injetiva. Dizemos também que f *codifica* $A \times B$, e que $f(x, y)$ é o *código* de $\langle x, y \rangle$.

Podemos usar funções de emparelhamento para codificar, por exemplo, pares de números naturais; ou, em outras palavras, podemos representar cada *par* de elementos por um *único* número. Usando a inversa da função de emparelhamento, podemos *descodificar* o número, isto é, determinar que par representa.

explanation

Problema 4.8. Dê uma enumeração do conjunto de todos os números racionais não negativos.

Problema 4.9. Mostre que $\mathbb{Q}$ é **enumerável**. Recordemos que qualquer número racional se pode escrever como uma fração z/m com $z \in \mathbb{Z}$, $m \in \mathbb{N}^+$.

Problema 4.10. Defina uma enumeração de $\mathbb{B}^*$.

Problema 4.11. Recordando o curso introdutório de lógica, cada tabela de verdade possível exprime uma função de verdade. Em outras palavras, as funções de verdade são todas as funções de $\mathbb{B}^k \rightarrow \mathbb{B}$ para algum k . Prove que o conjunto de todas as funções de verdade é **enumerável**.

Problema 4.12. Mostre que o conjunto de todos os subconjuntos finitos de um conjunto infinito **enumerável** arbitrário é **enumerável**.

Problema 4.13. Um subconjunto de $\mathbb{N}$ diz-se *confinito* sse for o complemento de um conjunto finito; isto é, $A \subseteq \mathbb{N}$ é confinito sse $\mathbb{N} \setminus A$ for finito. Seja I o conjunto cujos **membros** são exatamente os subconjuntos finitos e confinitos de $\mathbb{N}$. Mostre que I é **enumerável**.

Problema 4.14. Mostre que a união **enumerável** de conjuntos **enumeráveis** é **enumerável**. Isto é, sempre que $A_1, A_2, \dots$ são conjuntos e cada A_i é **enumerável**, então a união $\bigcup_{i=1}^{\infty} A_i$ de todos eles também é **enumerável**. [Nota: isto é difícil!]

Problema 4.15. Seja $f: A \times B \rightarrow \mathbb{N}$ uma função de emparelhamento arbitrária. Mostre que a inversa de f é uma enumeração de $A \times B$.

Problema 4.16. Especifique uma função que codifique $\mathbb{N}^3$.

`content/sets-functions-relations/size-of-sets/pairing-alt.tex`

4.5 Uma Função de Emparelhamento Alternativa

explanation Há outras enumerações de $\mathbb{N}^2$ que facilitam descobrir quais são suas inversas. sfr:siz:pai-alt: sec Eis uma. Em vez de visualizar a enumeração em uma matriz, comece com a lista de inteiros positivos associados a espaços (inicialmente) vazios. Imagine preencher esses espaços sucessivamente com pares $\langle n, m \rangle$ da seguinte forma. Começando com os pares que têm 0 na primeira posição (isto é, pares $\langle 0, m \rangle$), coloque o primeiro (isto é, $\langle 0, 0 \rangle$) no primeiro espaço vazio, depois pule um espaço vazio, coloque o segundo (isto é, $\langle 0, 2 \rangle$) no próximo espaço vazio, pule outro, e assim por diante. O começo (incompleto) de nossa enumeração agora se parece com isto

1	2	3	4	5	6	7	8	9	10	...
$\langle 0, 0 \rangle$		$\langle 0, 1 \rangle$		$\langle 0, 2 \rangle$		$\langle 0, 3 \rangle$		$\langle 0, 4 \rangle$		...

Repita isso com pares $\langle 1, m \rangle$ nos lugares que ainda permanecem vazios, novamente pulando cada outro espaço vazio:

1	2	3	4	5	6	7	8	9	10	...
$\langle 0, 0 \rangle$	$\langle 1, 0 \rangle$	$\langle 0, 1 \rangle$		$\langle 0, 2 \rangle$	$\langle 1, 1 \rangle$	$\langle 0, 3 \rangle$		$\langle 0, 4 \rangle$	$\langle 1, 2 \rangle$	...

Insira pares $\langle 2, m \rangle$, $\langle 2, m \rangle$, etc., da mesma forma. Nossa enumeração completa assim começa assim:

1	2	3	4	5	6	7	8	9	10	...
$\langle 0, 0 \rangle$	$\langle 1, 0 \rangle$	$\langle 0, 1 \rangle$	$\langle 2, 0 \rangle$	$\langle 0, 2 \rangle$	$\langle 1, 1 \rangle$	$\langle 0, 3 \rangle$	$\langle 3, 0 \rangle$	$\langle 0, 4 \rangle$	$\langle 1, 2 \rangle$	...

Se numerarmos as células na matriz acima de acordo com essa enumeração, não encontraremos uma linha zigue-zague arrumada, mas este arranjo:

	0	1	2	3	4	5	...
0	1	3	5	7	9	11	...
1	2	6	10	14	18	...	...
2	4	12	20	28	...	...	...
3	8	24	40	...	...	...	...
4	16	48	...	...	...	...	...
5	32	...	...	...	...	...	...
...	...	...	...	...	...	...	...

4.6. CONJUNTOS NÃO ENUMERÁVEIS

Podemos ver que os pares na linha 0 estão nas posições ímpares de nossa enumeração, isto é, o par $\langle 0, m \rangle$ está na posição $2m + 1$; os pares na segunda linha, $\langle 1, m \rangle$, estão em posições cujo número é o dobro de um número ímpar, especificamente, $2 \cdot (2m + 1)$; os pares na terceira linha, $\langle 2, m \rangle$, estão em posições cujo número é quatro vezes um número ímpar, $4 \cdot (2m + 1)$; e assim por diante. Os fatores de $(2m + 1)$ para cada linha, 1, 2, 4, 8, ..., são exatamente as potências de 2: $1 = 2^0$, $2 = 2^1$, $4 = 2^2$, $8 = 2^3$, ... De fato, o expoente relevante é sempre o primeiro membro do par em questão. Assim, para o par $\langle n, m \rangle$ o fator é 2^n . Isso nos dá a fórmula geral: $2^n \cdot (2m + 1)$. Contudo, isso é um mapeamento de pares para inteiros *positivos*, isto é, $\langle 0, 0 \rangle$ tem posição 1. Se quisermos começar na posição 0 devemos subtrair 1 do resultado. Isso nos dá:

Exemplo 4.15. A função $h: \mathbb{N}^2 \rightarrow \mathbb{N}$ dada por

$$h(n, m) = 2^n(2m + 1) - 1$$

é uma função de emparelhamento para o conjunto de pares de números naturais $\mathbb{N}^2$.

De acordo com nossa segunda enumeração de $\mathbb{N}^2$, o par $\langle 0, 0 \rangle$ tem código [explanation](#) $h(0, 0) = 2^0(2 \cdot 0 + 1) - 1 = 0$; $\langle 1, 2 \rangle$ tem código $2^1 \cdot (2 \cdot 2 + 1) - 1 = 2 \cdot 5 - 1 = 9$; $\langle 2, 6 \rangle$ tem código $2^2 \cdot (2 \cdot 6 + 1) - 1 = 51$.

Às vezes basta codificar pares de números naturais $\mathbb{N}^2$ sem exigir que a codificação seja sobrejetora. Tais codificações têm inversas que são apenas funções parciais.

Exemplo 4.16. A função $j: \mathbb{N}^2 \rightarrow \mathbb{N}^+$ dada por

$$j(n, m) = 2^n 3^m$$

é uma função [injetiva](#) $\mathbb{N}^2 \rightarrow \mathbb{N}$.

[content/sets-functions-relations/size-of-sets/non-enumerability.tex](#)

4.6 Conjuntos Não enumeráveis

[sfr:siz:nen:](#)
[sec](#)

Esta seção prova a não enumerabilidade de $\mathbb{B}^\omega$ e $\wp(\mathbb{Z}^+)$ usando a definição na [Seção 4.2](#). Foi concebida para ser um pouco mais elementar e um pouco mais detalhada que a versão na [Seção 4.11](#)

Alguns conjuntos, como o conjunto $\mathbb{Z}^+$ dos inteiros positivos, são infinitos. Até agora vimos exemplos de conjuntos infinitos que eram todos [enumerável](#).

CAPÍTULO 4. O TAMANHO DOS CONJUNTOS

Contudo, há também conjuntos infinitos que não têm essa propriedade. Tais conjuntos são chamados de *não enumerável*.

Em primeiro lugar, talvez já seja surpreendente que existam conjuntos *não enumerável*. Para qualquer conjunto *enumerável* A há uma função *sobrejetiva* $f: \mathbb{Z}^+ \rightarrow A$. Se um conjunto é *não enumerável*, não há tal função. Isto é, nenhuma função mapeando os infinitos *membros* de $\mathbb{Z}^+$ para A pode esgotar todo A . Assim, há “mais” *membros* em A do que os infinitos inteiros positivos.

Como se provaria que um conjunto é *não enumerável*? É preciso mostrar que nenhuma tal função sobrejetora pode existir. Equivalentemente, é preciso mostrar que os elementos de A não podem ser enumerados em uma lista infinita unidirecional. A melhor maneira de fazer isso é mostrar que toda lista de *membros* de A deve deixar de fora pelo menos um elemento; ou que nenhuma função $f: \mathbb{Z}^+ \rightarrow A$ pode ser *sobrejetiva*. Podemos fazer isso usando o *método diagonal* de Cantor. Dada uma lista de *membros* de A , digamos, $x_1, x_2, \dots$, construímos outro elemento de A que, por sua construção, não pode estar naquela lista.

Nosso primeiro exemplo é o conjunto $\mathbb{B}^\omega$ de todas as seqüências infinitas, sem lacunas, de 0s e 1s.

Teorema 4.17. $\mathbb{B}^\omega$ é *não enumerável*.

*sfr:siz:nen:
thm:nonenum-bin-omega*

Prova. Suponha, por absurdo, que $\mathbb{B}^\omega$ é *enumerável*, isto é, suponha que há uma lista $s_1, s_2, s_3, s_4, \dots$ de todos os *membros* de $\mathbb{B}^\omega$. Cada um desses s_i é em si uma seqüência infinita de 0s e 1s. Chamemos o j -ésimo elemento da i -ésima seqüência nessa lista de $s_i(j)$. Então a i -ésima seqüência s_i é

$$s_i(1), s_i(2), s_i(3), \dots$$

Podemos organizar essa lista, e os elementos de cada seqüência s_i nela, em uma matriz:

	1	2	3	4	...
1	$s_1(1)$	$s_1(2)$	$s_1(3)$	$s_1(4)$	...
2	$s_2(1)$	$s_2(2)$	$s_2(3)$	$s_2(4)$	...
3	$s_3(1)$	$s_3(2)$	$s_3(3)$	$s_3(4)$	...
4	$s_4(1)$	$s_4(2)$	$s_4(3)$	$s_4(4)$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Os rótulos ao lado indicam o número da seqüência na lista $s_1, s_2, \dots$; os números no topo rotulam os *membros* das seqüências individuais. Por exemplo, $s_1(1)$ é um nome para qualquer número, um 0 ou um 1, que é o primeiro *membro* na seqüência s_1 , e assim por diante.

Agora construímos uma seqüência infinita, $\bar{s}$, de 0s e 1s que não pode estar nessa lista. A definição de $\bar{s}$ dependerá da lista $s_1, s_2, \dots$. Qualquer lista infinita de seqüências infinitas de 0s e 1s dá origem a uma seqüência infinita $\bar{s}$ que está garantidamente fora da lista.

Para definir $\bar{s}$, especificamos todos os seus *membros*, isto é, especificamos $\bar{s}(n)$ para todo $n \in \mathbb{Z}^+$. Fazemos isso lendo a diagonal da matriz acima (daí

4.6. CONJUNTOS NÃO ENUMERÁVEIS

o nome “método diagonal”) e depois trocando cada 1 por 0 e cada 0 por 1. De maneira mais abstrata, definimos $\bar{s}(n)$ como 0 ou 1 conforme o n -ésimo **membro** da diagonal, $s_n(n)$, é 1 ou 0.

$$\bar{s}(n) = \begin{cases} 1 & \text{se } s_n(n) = 0 \\ 0 & \text{se } s_n(n) = 1. \end{cases}$$

Se você prefere fórmulas a definições por casos, também pode definir $\bar{s}(n) = 1 - s_n(n)$.

Claramente $\bar{s}$ é uma sequência infinita de 0s e 1s, já que é apenas a sequência espelhada da sequência de 0s e 1s que aparecem na diagonal de nossa matriz. Assim $\bar{s}$ é **um membro** de $\mathbb{B}^\omega$. Mas não pode estar na lista $s_1, s_2, \dots$. Por quê?

Não pode ser a primeira sequência na lista, s_1 , porque difere de s_1 no primeiro **membro**. Seja qual for $s_1(1)$, definimos $\bar{s}(1)$ como o oposto. Não pode ser a segunda sequência na lista, porque $\bar{s}$ difere de s_2 no segundo elemento: se $s_2(2)$ é 0, $\bar{s}(2)$ é 1, e vice-versa. E assim por diante.

Mais precisamente: se $\bar{s}$ estivesse na lista, haveria algum k tal que $\bar{s} = s_k$. Duas sequências são idênticas sse concordam em toda posição, isto é, para qualquer n , $\bar{s}(n) = s_k(n)$. Em particular, tomando $n = k$ como caso especial, $\bar{s}(k) = s_k(k)$ teria de valer. $s_k(k)$ é 0 ou 1. Se é 0, então $\bar{s}(k)$ deve ser 1—é assim que definimos $\bar{s}$. Mas se $s_k(k) = 1$, de novo por causa da forma como definimos $\bar{s}$, $\bar{s}(k) = 0$. Em ambos os casos $\bar{s}(k) \neq s_k(k)$.

Começamos supondo que há uma lista de **membros** de $\mathbb{B}^\omega$, $s_1, s_2, \dots$. A partir dessa lista construímos uma sequência $\bar{s}$ que provamos não poder estar na lista. Mas ela definitivamente é uma sequência de 0s e 1s se todos os s_i são sequências de 0s e 1s, isto é, $\bar{s} \in \mathbb{B}^\omega$. Isso mostra em particular que não pode haver lista de *todos* os **membros** de $\mathbb{B}^\omega$, já que para qualquer tal lista também poderíamos construir uma sequência $\bar{s}$ garantidamente fora da lista, de modo que a suposição de que há uma lista de todas as sequências em $\mathbb{B}^\omega$ leva a uma contradição. $\square$

Esse método de prova é chamado de “diagonalização” porque usa a diagonal da matriz para definir $\bar{s}$. A diagonalização não precisa envolver a presença de uma matriz: podemos mostrar que conjuntos não são **enumerável** usando uma ideia semelhante mesmo quando não há matriz nem diagonal propriamente dita.

Teorema 4.18. $\wp(\mathbb{Z}^+)$ não é **enumerável**.

Prova. Procedemos da mesma forma, mostrando que para toda lista de subconjuntos de $\mathbb{Z}^+$ há um subconjunto de $\mathbb{Z}^+$ que não pode estar na lista. Suponha que a seguinte é uma lista dada de subconjuntos de $\mathbb{Z}^+$:

$$Z_1, Z_2, Z_3, \dots$$

Agora definimos um conjunto $\bar{Z}$ tal que para qualquer $n \in \mathbb{Z}^+$, $n \in \bar{Z}$ sse $n \notin Z_n$:

$$\bar{Z} = \{n \in \mathbb{Z}^+ : n \notin Z_n\}$$

$\square$

$\bar{Z}$ é claramente um conjunto de inteiros positivos, já que por suposição cada Z_n o é, e assim $\bar{Z} \in \wp(\mathbb{Z}^+)$. Mas $\bar{Z}$ não pode estar na lista. Para mostrar isso, estabeleceremos que para cada $k \in \mathbb{Z}^+$, $\bar{Z} \neq Z_k$.

Seja $k \in \mathbb{Z}^+$ arbitrário. Definimos $\bar{Z}$ de modo que para qualquer $n \in \mathbb{Z}^+$, $n \in \bar{Z}$ sse $n \notin Z_n$. Em particular, tomando $n = k$, $k \in \bar{Z}$ sse $k \notin Z_k$. Mas isso mostra que $\bar{Z} \neq Z_k$, já que k é um membro de um mas não do outro, e assim $\bar{Z}$ e Z_k têm membros diferentes. Como k foi arbitrário, $\bar{Z}$ não está na lista $Z_1, Z_2, \dots$.

explanation

A prova precedente não mencionou uma diagonal, mas você pode pensar nela como envolvendo uma diagonal se a imaginar assim: imagine os conjuntos $Z_1, Z_2, \dots$, escritos em uma matriz, onde cada membro $j \in Z_i$ é listado na j -ésima coluna. Digamos que os primeiros quatro conjuntos nessa lista são $\{1, 2, 3, \dots\}$, $\{2, 4, 6, \dots\}$, $\{1, 2, 5\}$ e $\{3, 4, 5, \dots\}$. Então a matriz começaria com

$$\begin{array}{cccccccc} Z_1 = \{ & 1, & 2, & 3, & 4, & 5, & 6, & \dots \} \\ Z_2 = \{ & & 2, & & 4, & & 6, & \dots \} \\ Z_3 = \{ & 1, & 2, & & & 5, & & \} \\ Z_4 = \{ & & & 3, & 4, & 5, & 6, & \dots \} \\ & \vdots & & & & & & \ddots \end{array}$$

Então $\bar{Z}$ é o conjunto obtido percorrendo a diagonal, deixando de fora quaisquer números que aparecem ao longo da diagonal e incluindo aqueles j onde a matriz tem uma lacuna na j -ésima linha/coluna. No caso acima, deixaríamos de fora 1 e 2, incluiríamos 3, deixaríamos de fora 4, etc.

Problema 4.17. Mostre que $\wp(\mathbb{N})$ é não enumerável por um argumento diagonal.

Problema 4.18. Mostre que o conjunto de funções $f: \mathbb{Z}^+ \rightarrow \mathbb{Z}^+$ é não enumerável por um argumento diagonal explícito. Isto é, mostre que se $f_1, f_2, \dots$, é uma lista de funções e cada $f_i: \mathbb{Z}^+ \rightarrow \mathbb{Z}^+$, então há alguma $\bar{f}: \mathbb{Z}^+ \rightarrow \mathbb{Z}^+$ fora dessa lista.

content/sets-functions-relations/size-of-sets/reduction.tex

4.7 Redução

sfr:siz:red:
sec

Esta seção prova a não enumerabilidade por redução, correspondendo aos resultados na Seção 4.6. Uma versão alternativa, ligeiramente mais condensada, correspondente aos resultados na Seção 4.12, é fornecida na Seção 4.13.

4.7. REDUÇÃO

Mostramos $\wp(\mathbb{Z}^+)$ como **não enumerável** por um argumento de diagonalização. Já tínhamos uma prova de que $\mathbb{B}^\omega$, o conjunto de todas as sequências infinitas de 0s e 1s, é **não enumerável**. Eis outra maneira de provar que $\wp(\mathbb{Z}^+)$ é **não enumerável**: mostre que se $\wp(\mathbb{Z}^+)$ é **enumerável**, então $\mathbb{B}^\omega$ também é **enumerável**. Como sabemos que $\mathbb{B}^\omega$ não é **enumerável**, $\wp(\mathbb{Z}^+)$ também não pode ser. Isso é chamado de *reduzir* um problema a outro—neste caso, reduzimos o problema de enumerar $\mathbb{B}^\omega$ ao problema de enumerar $\wp(\mathbb{Z}^+)$. Uma solução para o último—uma enumeração de $\wp(\mathbb{Z}^+)$ —daria uma solução para o primeiro—uma enumeração de $\mathbb{B}^\omega$.

Como reduzimos o problema de enumerar um conjunto B ao de enumerar um conjunto A ? Fornecemos uma maneira de transformar uma enumeração de A em uma enumeração de B . A maneira mais fácil de fazer isso é definir uma função **sobrejetiva** $f: A \rightarrow B$. Se $x_1, x_2, \dots$ enumera A , então $f(x_1), f(x_2), \dots$ enumeraria B . Em nosso caso, procuramos uma função sobrejetora $f: \wp(\mathbb{Z}^+) \rightarrow \mathbb{B}^\omega$.

Problema 4.19. Mostre que se há uma função **injetiva** $g: B \rightarrow A$, e B é **não enumerável**, então A também é. Faça isso mostrando como você pode usar g para transformar uma enumeração de A em uma de B .

Prova do Teorema 4.18 por redução. Suponha que $\wp(\mathbb{Z}^+)$ fosse **enumerável**, e portanto que há uma enumeração dele, $Z_1, Z_2, Z_3, \dots$

Defina a função $f: \wp(\mathbb{Z}^+) \rightarrow \mathbb{B}^\omega$ fazendo $f(Z)$ ser a sequência s_k tal que $s_k(n) = 1$ sse $n \in Z$, e $s_k(n) = 0$ caso contrário. Isso define claramente uma função, já que sempre que $Z \subseteq \mathbb{Z}^+$, qualquer $n \in \mathbb{Z}^+$ ou é **um membro** de Z ou não é. Por exemplo, o conjunto $2\mathbb{Z}^+ = \{2, 4, 6, \dots\}$ dos pares positivos é mapeado para a sequência 010101..., o conjunto vazio é mapeado para 0000... e o conjunto $\mathbb{Z}^+$ para 1111....

Também é **sobrejetiva**: toda sequência de 0s e 1s corresponde a algum conjunto de inteiros positivos, nomeadamente aquele que tem como membros os inteiros correspondentes às posições onde a sequência tem 1s. Mais precisamente, suponha $s \in \mathbb{B}^\omega$. Defina $Z \subseteq \mathbb{Z}^+$ por:

$$Z = \{n \in \mathbb{Z}^+ : s(n) = 1\}$$

Então $f(Z) = s$, como pode ser verificado consultando a definição de f .

Agora considere a lista

$$f(Z_1), f(Z_2), f(Z_3), \dots$$

Como f é **sobrejetiva**, todo membro de $\mathbb{B}^\omega$ deve aparecer como valor de f para algum argumento, e assim deve aparecer na lista. Essa lista portanto deve enumerar todo $\mathbb{B}^\omega$.

Assim, se $\wp(\mathbb{Z}^+)$ fosse **enumerável**, $\mathbb{B}^\omega$ seria **enumerável**. Mas $\mathbb{B}^\omega$ é **não enumerável** (Teorema 4.17). Logo $\wp(\mathbb{Z}^+)$ é **não enumerável**. $\square$

explanation

É fácil confundir a direção em que a redução vai. Por exemplo, uma função **sobrejetiva** $g: \mathbb{B}^\omega \rightarrow B$ **não** estabelece que B é **não enumerável**. (Considere $g: \mathbb{B}^\omega \rightarrow \mathbb{B}$ definida por $g(s) = s(1)$, a função que mapeia uma sequência de 0s e 1s para seu primeiro **membro**. Ela é **sobrejetiva**, porque algumas sequências começam com 0 e outras começam com 1. Mas $\mathbb{B}$ é finito.) Note também que a função f deve ser **sobrejetiva**, caso contrário o argumento não se sustenta: $f(x_1), f(x_2), \dots$ não estariam garantidos de incluir todos os **membros** de B . Por exemplo,

$$h(n) = \underbrace{000 \dots 0}_n$$

define uma função $h: \mathbb{Z}^+ \rightarrow \mathbb{B}^\omega$, mas $\mathbb{Z}^+$ é **enumerável**.

Problema 4.20. Mostre que o conjunto de todos os *conjuntos* de pares de inteiros positivos é **não enumerável** por um argumento de redução.

Problema 4.21. Mostre que o conjunto X de todas as funções $f: \mathbb{N} \rightarrow \mathbb{N}$ é **não enumerável** por um argumento de redução (Dica: dê uma função sobrejetora de X para $\mathbb{B}^\omega$.)

Problema 4.22. Mostre que $\mathbb{N}^\omega$, o conjunto de sequências infinitas de números naturais, é **não enumerável** por um argumento de redução.

Problema 4.23. Seja P o conjunto de funções do conjunto dos inteiros positivos para o conjunto $\{0\}$, e seja Q o conjunto de funções *parciais* do conjunto dos inteiros positivos para o conjunto $\{0\}$. Mostre que P é **enumerável** e Q não é. (Dica: reduza o problema de enumerar $\mathbb{B}^\omega$ a enumerar Q).

Problema 4.24. Seja S o conjunto de todas as funções **sobrejetivas** do conjunto dos inteiros positivos para o conjunto $\{0,1\}$, isto é, S consiste de todas as **sobrejetivas** $f: \mathbb{Z}^+ \rightarrow \mathbb{B}$. Mostre que S é **não enumerável**.

Problema 4.25. Mostre que o conjunto $\mathbb{R}$ de todos os números reais é **não enumerável**.

`content/sets-functions-relations/size-of-sets/equenumerous-sets.tex`

4.8 Equinumerosidade

Temos uma noção intuitiva do “tamanho” dos conjuntos, que funciona bem para conjuntos finitos. E quanto aos infinitos? Se quisermos uma forma formal de comparar os tamanhos de dois conjuntos de *qualquer* cardinalidade, é boa ideia começar por definir quando dois conjuntos têm o mesmo tamanho. Eis Frege:

sfr:siz:equ:
sec

4.8. EQUINUMEROSIDADE

Se um garçom quer ter certeza de que pôs exatamente tantas facas quantos pratos sobre a mesa, não precisa contar nem uns nem outros, se simplesmente puser uma faca à direita de cada prato, de modo que cada faca na mesa fique à direita de algum prato. Os pratos e as facas ficam assim correlacionados de modo único, e de fato por essa mesma relação espacial. (Frege, 1884, § 70)

A intuição central desta passagem traduz-se numa definição formal:

Definição 4.19. *A é equinumeroso a B, escrevendo-se $A \approx B$, sse existir uma bijeção $f: A \rightarrow B$.*

Proposição 4.20. *A relação de equinumerosidade é uma relação de equivalência.*

Prova. É preciso mostrar que a equinumerosidade é reflexiva, simétrica e transitiva. Sejam A, B e C conjuntos.

Reflexividade. A aplicação identidade $\text{Id}_A: A \rightarrow A$, com $\text{Id}_A(x) = x$ para todo $x \in A$, é uma bijeção. Logo $A \approx A$.

Simetria. Suponhamos $A \approx B$, isto é, que existe uma bijeção $f: A \rightarrow B$. Como f é bijetiva, a inversa f^{-1} existe e também é bijetiva. Logo $f^{-1}: B \rightarrow A$ é uma bijeção, pelo que $B \approx A$.

Transitividade. Suponhamos $A \approx B$ e $B \approx C$, isto é, que existem bijeções $f: A \rightarrow B$ e $g: B \rightarrow C$. Então a composição $g \circ f: A \rightarrow C$ é bijetiva, logo $A \approx C$. $\square$

Proposição 4.21. *Se $A \approx B$, então A é enumerável se e só se B o é.*

A prova seguinte usa Definição 4.4 se Seção 4.2 estiver incluída e Definição 4.27 caso contrário.

Prova. Suponhamos $A \approx B$, logo existe alguma bijeção $f: A \rightarrow B$, e suponhamos que A é enumerável. Então ou $A = \emptyset$ ou existe uma função sobrejetiva $g: \mathbb{Z}^+ \rightarrow A$. Se $A = \emptyset$, então $B = \emptyset$ também (caso contrário existiria um membro $y \in B$ sem $x \in A$ com $g(x) = y$). Se, por outro lado, $g: \mathbb{Z}^+ \rightarrow A$ for sobrejetiva, então $f \circ g: \mathbb{Z}^+ \rightarrow B$ é sobrejetiva. De fato, seja $y \in B$. Como f é sobrejetiva, existe $x \in A$ tal que $f(x) = y$. Como g é sobrejetiva, existe $n \in \mathbb{Z}^+$ tal que $g(n) = x$. Logo,

$$(f \circ g)(n) = f(g(n)) = f(x) = y$$

e portanto $f \circ g$ é sobrejetiva. Temos que $f \circ g$ é uma enumeração de B , logo B é enumerável.

Se B é enumerável, obtemos que A é enumerável repetindo o argumento com a bijeção $f^{-1}: B \rightarrow A$ em lugar de f . $\square$

Problema 4.26. Mostre que, se $A \approx C$ e $B \approx D$, e $A \cap B = C \cap D = \emptyset$, então $A \cup B \approx C \cup D$.

Problema 4.27. Mostre que, se A é infinito e enumerável, então $A \approx \mathbb{N}$.

content/sets-functions-relations/size-of-sets/comparing-size.tex

4.9 Conjuntos de Tamanhos Diferentes e o Teorema de Cantor

explanation

Apresentamos uma formulação precisa da ideia de que dois conjuntos têm o mesmo tamanho. Também podemos precisar a ideia de que um conjunto é menor que outro. A nossa definição de “é menor do que (ou equinumeroso)” exigirá, em vez de uma bijeção entre os conjuntos, uma injeção do primeiro para o segundo. Se tal função existir, o tamanho do primeiro conjunto é menor ou igual ao do segundo. Intuitivamente, uma injeção de um conjunto em outro garante que a imagem da função tem pelo menos tantos membros quanto o domínio, pois não há dois membros do domínio que vão parar ao mesmo membro da imagem.

sfr:siz:car:
sec

Definição 4.22. A não é maior do que B , escrevendo-se $A \preceq B$, sse existir uma injeção $f: A \rightarrow B$.

É claro que isto é uma relação reflexiva e transitiva, mas não simétrica (fica como exercício). Podemos ainda introduzir uma noção que diz que um conjunto é (estritamente) menor que outro.

Definição 4.23. A é menor do que B , escrevendo-se $A \prec B$, sse existir uma injeção $f: A \rightarrow B$ mas não existir bijeção $g: A \rightarrow B$, isto é, $A \preceq B$ e $A \not\approx B$.

É claro que esta relação é irreflexiva e transitiva. (Também fica como exercício.) Com esta notação, um conjunto A é enumerável sse $A \preceq \mathbb{N}$, e A é não enumerável sse $\mathbb{N} \prec A$. Isto permite reformular o Teorema 4.32 como a observação de que $\mathbb{N} \prec \wp(\mathbb{N})$. De fato, Cantor (1892) provou que este último ponto é completamente geral:

Teorema 4.24 (Cantor). $A \prec \wp(A)$, para qualquer conjunto A .

sfr:siz:car:
thm:cantor

Prova. A aplicação $f(x) = \{x\}$ é uma injeção $f: A \rightarrow \wp(A)$, pois se $x \neq y$, então também $\{x\} \neq \{y\}$ por extensionalidade, logo $f(x) \neq f(y)$. Temos pois $A \preceq \wp(A)$.

Apresentamos a prova longa se Seção 4.6 estiver presente; caso contrário, uma prova mais curta alinhada com a Seção 4.12.

4.9. CONJUNTOS DE TAMANHOS DIFERENTES E O TEOREMA DE CANTOR

Mostraremos agora que não pode existir função **sobrejetiva** $g: A \rightarrow \wp(A)$, quanto mais **bijetiva**, e portanto que $A \not\approx \wp(A)$. Suponhamos que $g: A \rightarrow \wp(A)$. Como g é total, todo $x \in A$ é mapeado para um subconjunto $g(x) \subseteq A$. Podemos mostrar que g não pode ser sobrejetora. Para isso, definimos um subconjunto $\bar{A} \subseteq A$ que, por definição, não pode estar no contradomínio de g . Seja

$$\bar{A} = \{x \in A : x \notin g(x)\}.$$

Como $g(x)$ está definido para todo $x \in A$, $\bar{A}$ é claramente um subconjunto bem definido de A . Mas não pode estar no contradomínio de g . Seja $x \in A$ arbitrário; mostramos que $\bar{A} \neq g(x)$. Se $x \in g(x)$, então não satisfaz $x \notin g(x)$, logo, pela definição de $\bar{A}$, temos $x \notin \bar{A}$. Se $x \in \bar{A}$, tem de satisfazer a propriedade definidora de $\bar{A}$, isto é, $x \in A$ e $x \notin g(x)$. Como x era arbitrário, isto mostra que, para cada $x \in A$, $x \in g(x)$ sse $x \notin \bar{A}$, logo $g(x) \neq \bar{A}$. Em outras palavras, $\bar{A}$ não pode estar no contradomínio de g , o que contradiz a suposição de que g é sobrejetora. $\square$

É instrutivo comparar a prova do Teorema 4.24 com a do Teorema 4.18. Ali mostramos que, para qualquer lista $Z_1, Z_2, \dots$, de subconjuntos de $\mathbb{Z}^+$, se pode construir um conjunto $\bar{Z}$ de números garantidamente fora da lista. Estava garantido que não estava na lista porque, para todo $n \in \mathbb{Z}^+$, $n \in Z_n$ sse $n \notin \bar{Z}$. Assim, há sempre algum número que é **um membro** de um entre Z_n ou $\bar{Z}$ mas não do outro. Seguimos a mesma ideia aqui, exceto que os índices n são agora **membros** de A em vez de $\mathbb{Z}^+$. O conjunto $\bar{A}$ está definido de modo a diferir de $g(x)$ para cada $x \in A$, porque $x \in g(x)$ sse $x \notin \bar{A}$. De novo, há sempre **um membro** de A que é **um membro** de um entre $g(x)$ e $\bar{A}$ mas não do outro. E tal como $\bar{Z}$ portanto não pode estar na lista $Z_1, Z_2, \dots$, $\bar{A}$ não pode estar no contradomínio de g . explanation

É instrutivo comparar a prova do Teorema 4.24 com a do Teorema 4.32. Ali mostramos que, para qualquer lista $N_0, N_1, N_2, \dots$, de subconjuntos de $\mathbb{N}$, podemos construir um conjunto D de números garantidamente fora da lista. Estava garantido que não estava na lista porque $n \in N_n$ sse $n \notin D$, para todo $n \in \mathbb{N}$. Seguimos a mesma ideia aqui, exceto que os índices n são agora **membros** de A em vez de $\mathbb{N}$. O conjunto D está definido de modo a diferir de $g(x)$ para cada $x \in A$, porque $x \in g(x)$ sse $x \notin D$.

A prova vale também a pena comparar com a do Paradoxo de Russell, Teorema 1.29. De fato, o Teorema de Cantor foi a inspiração para o paradoxo de Russell.

Problema 4.28. Mostre que não pode existir **uma injeção** $g: \wp(A) \rightarrow A$, para qualquer conjunto A . Sugestão: suponha $g: \wp(A) \rightarrow A$ **injetiva**. Considere $D = \{g(B) : B \subseteq A \text{ e } g(B) \notin B\}$. Seja $x = g(D)$. Use o fato de g ser **injetiva** para obter uma contradição.

4.10 A Noção de Tamanho e Schröder-Bernstein

explanation Eis um pensamento intuitivo: se A não é maior do que B e B não é maior do que A , então A e B são equinumerosos. Sejam honestos: se isto estivesse *errado*, dificilmente poderíamos justificar que a noção definida de equinumerosidade tem a ver com comparações de “tamanhos” entre conjuntos. Felizmente, o pensamento intuitivo está certo. Isto justifica-se pelo Teorema de Schröder-Bernstein.

Teorema 4.25 (Schröder-Bernstein). *Se $A \preceq B$ e $B \preceq A$, então $A \approx B$.*

sfr:siz:sb:
sec
thm:schroder-bernstein

explanation Em outras palavras, se existir uma **injeção** de A para B e uma **injeção** de B para A , então existe uma **bijeção** de A para B .

Este resultado é, contudo, realmente bastante *difícil* de provar. De fato, embora Cantor o tivesse enunciado, outros provaram-no.¹ Por agora, pode (e deve) aceitá-lo como verdadeiro.

Felizmente, Schröder-Bernstein é *correto* e legítima pensarmos nas relações que definimos, isto é, $A \approx B$ e $A \preceq B$, como tendo a ver com “tamanho”. Além disso, Schröder-Bernstein é muito *útil*. Pode ser difícil imaginar uma **bijeção** entre dois conjuntos equinumerosos. O Teorema de Schröder-Bernstein permite decompor a comparação em casos em que só precisamos de uma **injeção** do primeiro para o segundo e vice-versa.

As seções **Seção 4.11**, **Seção 4.12**, **Seção 4.13** são versões alternativas da **Seção 4.2**, **Seção 4.6**, **Seção 4.7** de Tim Button para uso em seu texto Open Set Theory. São um pouco mais avançadas e usam uma definição diferente de enumerabilidade mais adequada em um contexto de teoria dos conjuntos (isto é, **bijeção** com $\mathbb{N}$ ou um segmento inicial, em vez de ser listável ou ser a imagem de uma função sobrejetora de $\mathbb{Z}^+$).

`content/sets-functions-relations/size-of-sets/enumerability-alt.tex`

4.11 Enumerações e Conjuntos Enumeráveis

sfr:siz:enm-alt:
sec

Esta seção define enumerações como **bijeções** com (segmentos iniciais) de $\mathbb{N}$, como se faz em teoria dos conjuntos. Por isso difere ligeiramente das definições em **Seção 4.2** e repete os exemplos dessa seção. É também um pouco mais concisa.

¹Para mais sobre a história, ver por exemplo **Potter** (2004, pp. 165–6).

4.11. ENUMERAÇÕES E CONJUNTOS ENUMERÁVEIS

Podemos especificar um conjunto finito simplesmente enumerando os seus **membros**. Fazemo-lo quando definimos um conjunto assim:

$$A = \{a_1, a_2, \dots, a_n\}.$$

Supondo que os **membros** $a_1, \dots, a_n$ são todos distintos, obtemos **uma bijeção** entre A e os primeiros n números naturais $0, \dots, n-1$. Reciprocamente, como todo o conjunto finito tem apenas finitos **membros**, todo o conjunto finito pode ser posto nessa correspondência. Em outras palavras, se A é finito, existe **uma bijeção** entre A e $\{0, \dots, n-1\}$, onde n é o número de **membros** de A .

Se admitirmos certos tipos de conjuntos infinitos, admitiremos também que alguns conjuntos infinitos possam ser enumerados. Podemos precisar isto dizendo que um conjunto infinito é enumerado por **uma bijeção** entre ele e todo o $\mathbb{N}$.

Definição 4.26 (Enumeração, no sentido conjuntista). Uma *enumeração* de um conjunto A é **uma bijeção** cujo contradomínio é A e cujo domínio é ou um conjunto inicial de números naturais $\{0, 1, \dots, n\}$ ou todo o conjunto dos números naturais $\mathbb{N}$.

Há uma base intuitiva para este uso da palavra *enumeração*. Dizer que enumeramos um conjunto A é dizer que existe **uma bijeção** f que nos permite **contar** os elementos do conjunto A . O elemento de índice 0 é $f(0)$, o de índice 1 é $f(1)$, ... o de índice n é $f(n)$.² O raciocínio fica ainda mais claro com o seguinte:

sfr:siz:enm-alt:
defn:enumerable

Definição 4.27. Um conjunto A é **enumerável** se e só se $A = \emptyset$ ou existe uma enumeração de A . Dizemos que A é **não enumerável** se e só se A **não** é **enumerável**.

Logo um conjunto é **enumerável** se e só se é vazio ou se podemos usar uma enumeração para contar os seus **membros**.

Exemplo 4.28. Uma função que enumera os números naturais é simplesmente a função identidade $\text{Id}_{\mathbb{N}}: \mathbb{N} \rightarrow \mathbb{N}$ dada por $\text{Id}_{\mathbb{N}}(n) = n$. Uma função que enumera os números naturais *positivos*, $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$, é a função $g(n) = n + 1$, isto é, a função sucessora.

Problema 4.29. Mostre que um conjunto A é **enumerável** se e só se $A = \emptyset$ ou existe **uma sobrejeção** $f: \mathbb{N} \rightarrow A$. Mostre que A é **enumerável** se e só se existe **uma injeção** $g: A \rightarrow \mathbb{N}$.

Exemplo 4.29. As funções $f: \mathbb{N} \rightarrow \mathbb{N}$ e $g: \mathbb{N} \rightarrow \mathbb{N}$ dadas por

$$\begin{aligned} f(n) &= 2n \text{ e} \\ g(n) &= 2n + 1 \end{aligned}$$

²Sim, contamos a partir de 0. Claro que também poderíamos começar em 1. Não faria grande diferença. Bastaria substituir $\mathbb{N}$ por $\mathbb{Z}^+$.

CAPÍTULO 4. O TAMANHO DOS CONJUNTOS

enumeram, respectivamente, os números naturais pares e os ímpares. Mas nenhuma é **sobrejetiva**, logo nenhuma é uma enumeração de $\mathbb{N}$.

Problema 4.30. Defina uma enumeração dos quadrados perfeitos 1, 4, 9, 16, ...

Exemplo 4.30. Seja $\lceil x \rceil$ a função *teto*, que arredonda x por excesso para o inteiro mais próximo. Então a função $f: \mathbb{N} \rightarrow \mathbb{Z}$ dada por:

$$f(n) = (-1)^n \left\lceil \frac{n}{2} \right\rceil$$

enumera o conjunto dos inteiros $\mathbb{Z}$ do seguinte modo:

$$\begin{array}{cccccccc} f(0) & f(1) & f(2) & f(3) & f(4) & f(5) & f(6) & \dots \\ \left\lceil \frac{0}{2} \right\rceil & -\left\lceil \frac{1}{2} \right\rceil & \left\lceil \frac{2}{2} \right\rceil & -\left\lceil \frac{3}{2} \right\rceil & \left\lceil \frac{4}{2} \right\rceil & -\left\lceil \frac{5}{2} \right\rceil & \left\lceil \frac{6}{2} \right\rceil & \dots \\ 0 & -1 & 1 & -2 & 2 & -3 & 3 & \dots \end{array}$$

Repare como f gera os valores de $\mathbb{Z}$ “saltitando” entre inteiros positivos e negativos. Também se pode ver f definida por casos assim:

$$f(n) = \begin{cases} \frac{n}{2} & \text{se } n \text{ é par} \\ -\frac{n+1}{2} & \text{se } n \text{ é ímpar} \end{cases}$$

Problema 4.31. Mostre que se A e B são **enumeráveis**, então $A \cup B$ também é.

Problema 4.32. Mostre por indução em n que se $A_1, A_2, \dots, A_n$ são todos **enumeráveis**, então $A_1 \cup \dots \cup A_n$ também é.

content/sets-functions-relations/size-of-sets/non-enumerability-alt.tex

4.12 Conjuntos Não Enumeráveis

sfr:siz:nen-alt:
sec

Esta seção prova a não enumerabilidade de $\mathbb{B}^\omega$ e de $\wp(\mathbb{N})$ usando as definições na **Seção 4.11**, isto é, exigindo uma bijeção com $\mathbb{N}$ em vez de uma sobrejeção a partir de $\mathbb{Z}^+$.

explanation O conjunto $\mathbb{N}$ dos números naturais é infinito. Também é trivialmente **enumerável**. O fato notável, porém, é que existem conjuntos **não enumeráveis**, isto é, conjuntos que não são **enumeráveis** (ver **Definição 4.27**).

4.12. CONJUNTOS NÃO ENUMERÁVEIS

Isto pode surpreender. Afinal, dizer que A é não enumerável é dizer que não existe bijeção $f: \mathbb{N} \rightarrow A$; isto é, nenhuma função que mapeie os infinitos membros de $\mathbb{N}$ para A esgota todo A . Logo, se A é não enumerável, há “mais” membros em A do que números naturais.

Para provar que um conjunto é não enumerável, é preciso mostrar que não pode existir uma bijeção adequada. A melhor forma é mostrar que qualquer tentativa de enumerar membros de A tem de deixar pelo menos um membro de fora; isto mostra que nenhuma função $f: \mathbb{N} \rightarrow A$ é sobrejetiva. Uma estratégia geral para o estabelecer é usar o *método diagonal* de Cantor. Dada uma lista de membros de A , digamos $x_1, x_2, \dots$, construímos outro membro de A que, pela sua construção, não pode estar nessa lista.

Tudo isto compreende-se melhor com um exemplo. O nosso primeiro exemplo é o conjunto $\mathbb{B}^\omega$ de todas as cadeias infinitas de 0s e 1s. (O símbolo ‘ $\mathbb{B}$ ’ significa binário, e podemos pensar nele como o conjunto de dois elementos $\{0, 1\}$.)

Teorema 4.31. $\mathbb{B}^\omega$ é não enumerável.

Prova. Considere-se qualquer enumeração de um subconjunto de $\mathbb{B}^\omega$. Temos então uma lista $s_0, s_1, s_2, \dots$ em que cada s_n é uma cadeia infinita de 0s e 1s. Seja $s_n(m)$ o m -ésimo dígito da cadeia s_n . Podemos pois pensar na lista como uma matriz, colocando $s_n(m)$ na n -ésima linha e m -ésima coluna:

	0	1	2	3	...
0	$s_0(0)$	$s_0(1)$	$s_0(2)$	$s_0(3)$	...
1	$s_1(0)$	$s_1(1)$	$s_1(2)$	$s_1(3)$	...
2	$s_2(0)$	$s_2(1)$	$s_2(2)$	$s_2(3)$	...
3	$s_3(0)$	$s_3(1)$	$s_3(2)$	$s_3(3)$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Construiremos agora uma cadeia infinita, d , de 0s e 1s que não está nesta lista, especificando cada uma das suas entradas, isto é, $d(n)$ para todo $n \in \mathbb{N}$. Intuitivamente, lemos a diagonal da matriz acima (daí o nome “método diagonal”) e trocamos cada 0 por 1 e cada 1 por 0. De modo mais abstrato, definimos $d(n)$ como 0 ou 1 consoante o n -ésimo membro da diagonal, $s_n(n)$, seja 1 ou 0, isto é:

$$d(n) = \begin{cases} 1 & \text{se } s_n(n) = 0 \\ 0 & \text{se } s_n(n) = 1 \end{cases}$$

Claramente $d \in \mathbb{B}^\omega$, pois é uma cadeia infinita de 0s e 1s. Mas construímos d de modo que $d(n) \neq s_n(n)$ para qualquer $n \in \mathbb{N}$, isto é, d difere de s_n na sua n -ésima entrada. Logo $d \neq s_n$ para qualquer $n \in \mathbb{N}$, pelo que d não pode estar na lista $s_0, s_1, s_2, \dots$

Mostramos assim que, dada uma enumeração arbitrária de algum subconjunto de $\mathbb{B}^\omega$, ela omite algum membro de $\mathbb{B}^\omega$. Logo não existe enumeração do conjunto $\mathbb{B}^\omega$, isto é, $\mathbb{B}^\omega$ é não enumerável. $\square$

explanation

Este método de prova chama-se “diagonalização” porque usa a diagonal da matriz para definir d . Contudo, a diagonalização não exige necessariamente uma matriz. De fato, podemos mostrar que um conjunto é **não enumerável** por uma ideia semelhante mesmo quando não há matriz nem diagonal literal. O resultado seguinte ilustra como.

Teorema 4.32. $\wp(\mathbb{N})$ não é **enumerável**.

*sfr:siz:nen-alt:
thm:nonenum-pownat*

Prova. Procedemos da mesma forma, mostrando que qualquer lista de subconjuntos de $\mathbb{N}$ omite algum subconjunto de $\mathbb{N}$. Suponhamos, pois, que temos uma lista $N_0, N_1, N_2, \dots$ de subconjuntos de $\mathbb{N}$. Definimos um conjunto D assim: $n \in D$ sse $n \notin N_n$:

$$D = \{n \in \mathbb{N} : n \notin N_n\}$$

Claramente $D \subseteq \mathbb{N}$. Mas D não pode estar na lista: por construção, $n \in D$ sse $n \notin N_n$, logo $D \neq N_n$ para qualquer $n \in \mathbb{N}$. $\square$

explanation

A prova precedente não mencionou uma diagonal. Ainda assim, pode pensar-se nela como envolvendo uma diagonal, imaginando os conjuntos $N_0, N_1, \dots$ dispostos numa matriz em que escrevemos N_n na n -ésima linha colocando m na m -ésima coluna sse $m \in N_n$. Por exemplo, se os primeiros quatro conjuntos da lista são $\{0, 1, 2, \dots\}$, $\{1, 3, 5, \dots\}$, $\{0, 1, 4\}$ e $\{2, 3, 4, \dots\}$, a matriz começaria com

$$\begin{array}{ccccccc} N_0 = \{ & \mathbf{0}, & 1, & 2, & & & \dots \} \\ N_1 = \{ & & \mathbf{1}, & & 3, & & 5, \dots \} \\ N_2 = \{ & 0, & 1, & & & 4 & \} \\ N_3 = \{ & & & 2, & \mathbf{3}, & 4, & \dots \} \\ & \vdots & & & & & \ddots \end{array}$$

Então D é o conjunto obtido percorrendo a diagonal, pondo $n \in D$ sse n não está na diagonal. No caso acima, excluiríamos 0 e 1, incluiríamos 2, excluiríamos 3, etc.

Problema 4.33. Mostre que o conjunto de todas as funções $f: \mathbb{N} \rightarrow \mathbb{N}$ é **não enumerável** por um argumento diagonal explícito. Isto é, mostre que, se $f_1, f_2, \dots$ é uma lista de funções e cada $f_i: \mathbb{N} \rightarrow \mathbb{N}$, então existe alguma $g: \mathbb{N} \rightarrow \mathbb{N}$ que não está nesta lista.

content/sets-functions-relations/size-of-sets/reduction-alt.tex

4.13 Redução

*sfr:siz:red-alt:
sec*

4.13. REDUÇÃO

Esta seção prova a não enumerabilidade por redução, alinhada com os resultados na [Seção 4.12](#). Uma versão alternativa, ligeiramente mais elaborada, alinhada com os resultados na [Seção 4.6](#), encontra-se na [Seção 4.7](#).

Provamos que $\mathbb{B}^\omega$ é **não enumerável** por um argumento de diagonalização. Usamos um argumento semelhante para mostrar que $\wp(\mathbb{N})$ é **não enumerável**. Há outra forma de provar que $\wp(\mathbb{N})$ é **não enumerável**: mostrar que *se $\wp(\mathbb{N})$ é enumerável então $\mathbb{B}^\omega$ também é enumerável*. Como sabemos que $\mathbb{B}^\omega$ é **não enumerável**, segue que $\wp(\mathbb{N})$ também o é.

Isto chama-se *reduzir* um problema a outro. Neste caso, reduzimos o problema de enumerar $\mathbb{B}^\omega$ ao problema de enumerar $\wp(\mathbb{N})$. Uma solução para este último—uma enumeração de $\wp(\mathbb{N})$ —daria uma solução para o primeiro—uma enumeração de $\mathbb{B}^\omega$.

Para reduzir o problema de enumerar um conjunto B ao de enumerar um conjunto A , indicamos uma forma de transformar uma enumeração de A numa enumeração de B . A forma mais simples é definir **uma sobrejeção** $f: A \rightarrow B$. Se $x_1, x_2, \dots$ enumeram A , então $f(x_1), f(x_2), \dots$ enumerariam B . No nosso caso, procuramos **uma sobrejeção** $f: \wp(\mathbb{N}) \rightarrow \mathbb{B}^\omega$.

Problema 4.34. Mostre que, se existir uma função **injetiva** $g: B \rightarrow A$, e B for **não enumerável**, então A também o é. Faça-o mostrando como pode usar g para transformar uma enumeração de A numa de B .

Prova do Teorema 4.32 por redução. Para a redução, suponhamos que $\wp(\mathbb{N})$ é **enumerável** e portanto que existe uma enumeração $N_1, N_2, N_3, \dots$

Definimos a função $f: \wp(\mathbb{N}) \rightarrow \mathbb{B}^\omega$ pondo $f(N)$ igual à cadeia s_k tal que $s_k(n) = 1$ sse $n \in N$, e $s_k(n) = 0$ caso contrário.

Isto define claramente uma função, pois sempre que $N \subseteq \mathbb{N}$, qualquer $n \in \mathbb{N}$ ou é **um membro** de N ou não é. Por exemplo, o conjunto $2\mathbb{N} = \{2n : n \in \mathbb{N}\} = \{0, 2, 4, 6, \dots\}$ dos naturais pares é mapeado para a cadeia $1010101\dots$; $\emptyset$ é mapeado para $0000\dots$; $\mathbb{N}$ é mapeado para $1111\dots$.

Também é **sobrejetiva**: toda a cadeia de 0s e 1s corresponde a algum conjunto de números naturais, nomeadamente aquele que tem como membros os naturais nas posições em que a cadeia tem um 1. Com mais precisão, se $s \in \mathbb{B}^\omega$, defina-se $N \subseteq \mathbb{N}$ por:

$$N = \{n \in \mathbb{N} : s(n) = 1\}$$

Então $f(N) = s$, como se verifica consultando a definição de f .

Consideremos agora a lista

$$f(N_1), f(N_2), f(N_3), \dots$$

Como f é **sobrejetiva**, todo o membro de $\mathbb{B}^\omega$ tem de aparecer como valor de f para algum argumento, e portanto na lista. Esta lista teria de enumerar todo $\mathbb{B}^\omega$.

Logo, se $\wp(\mathbb{N})$ fosse **enumerável**, $\mathbb{B}^\omega$ seria **enumerável**. Mas $\mathbb{B}^\omega$ é **não enumerável** ([Teorema 4.31](#)). Logo $\wp(\mathbb{N})$ é **não enumerável**. $\square$

Problema 4.35. Mostre que o conjunto X de todas as funções $f: \mathbb{N} \rightarrow \mathbb{N}$ é **não enumerável** por um argumento de redução (Sugestão: dê uma função sobrejetora de X para $\mathbb{B}^\omega$.)

Problema 4.36. Mostre que o conjunto de todos os *conjuntos* de pares de números naturais, isto é, $\wp(\mathbb{N} \times \mathbb{N})$, é **não enumerável** por um argumento de redução.

Problema 4.37. Mostre que $\mathbb{N}^\omega$, o conjunto das sequências infinitas de números naturais, é **não enumerável** por um argumento de redução.

Problema 4.38. Seja S o conjunto de todas as **sobrejeções** de $\mathbb{N}$ para o conjunto $\{0, 1\}$, isto é, S consiste em todas as **sobrejeções** $f: \mathbb{N} \rightarrow \mathbb{B}$. Mostre que S é **não enumerável**.

Problema 4.39. Mostre que o conjunto $\mathbb{R}$ de todos os números reais é **não enumerável**.

Capítulo 5

Aritmetização

O material deste capítulo apresenta a construção dos sistemas numéricos na teoria ingênua dos conjuntos. É retirado do texto *Open Set Theory* de Tim Button.

`content/sets-functions-relations/arithmeticization/integers.tex`

5.1 De $\mathbb{N}$ para $\mathbb{Z}$

Eis duas constatações básicas (no sentido de “ideias que retemos”):

`sfr:arith:int:
sec`

1. Todo inteiro pode escrever-se na forma $n - m$, com $n, m \in \mathbb{N}$.

5.1. DE $\mathbb{N}$ PARA $\mathbb{Z}$

2. A informação codificada numa expressão $n - m$ pode igualmente codificar-se por um par ordenado $\langle n, m \rangle$.

Já sabemos que os pares ordenados de números naturais são os **membros** de $\mathbb{N}^2$; estamos a assumir, também, que compreendemos $\mathbb{N}$. Eis uma sugestão ingênua, fundada nestas duas constatações: *tratar os inteiros como pares ordenados de números naturais*.

De fato, esta sugestão é demasiado ingênua. Obviamente, queremos que seja verdade que $0 - 2 = 4 - 6$. Mas, evidentemente, $\langle 0, 2 \rangle \neq \langle 4, 6 \rangle$. Logo, não podemos simplesmente dizer que $\mathbb{N}^2$ é o conjunto dos inteiros.

Generalizando a partir do problema precedente, o que queremos é o seguinte:

$$a - b = c - d \text{ sse } a + d = c + b$$

(Deve ser óbvio que assim é que os inteiros *devem* comportar-se: basta somar b e d a ambos os lados.) A forma fácil de garantir este comportamento é definir uma relação de equivalência entre pares ordenados, $\sim$, do seguinte modo:

$$\langle a, b \rangle \sim \langle c, d \rangle \text{ sse } a + d = c + b$$

Resta-nos mostrar que isto é uma relação de equivalência.

Proposição 5.1. $\sim$ é uma relação de equivalência.

Prova. Há que mostrar que $\sim$ é reflexiva, simétrica e transitiva.

Reflexividade: Evidentemente $\langle a, b \rangle \sim \langle a, b \rangle$, pois $a + b = b + a$.

Simetria: Suponhamos $\langle a, b \rangle \sim \langle c, d \rangle$, logo $a + d = c + b$. Então $c + b = a + d$, pelo que $\langle c, d \rangle \sim \langle a, b \rangle$.

Transitividade: Suponhamos $\langle a, b \rangle \sim \langle c, d \rangle \sim \langle m, n \rangle$. Logo, $a + d = c + b$ e $c + n = m + d$. Assim, $a + d + c + n = c + b + m + d$, e portanto $a + n = m + b$. Logo, $\langle a, b \rangle \sim \langle m, n \rangle$. $\square$

Podemos agora usar esta relação de equivalência para formar classes de equivalência:

Definição 5.2. Os inteiros são as classes de equivalência, sob $\sim$, dos pares ordenados de números naturais; isto é, $\mathbb{Z} = \mathbb{N}^2 / \sim$.

Agora, podem surgir várias reações *filosóficas* distintas face a esta definição estipulativa. Antes de as considerarmos, porém, vale a pena continuar com alguns pormenores técnicos.

Tendo dito o que são os inteiros, precisamos de definir funções e relações básicas sobre eles. Escrevamos $[m, n]_\sim$ para a classe de equivalência sob $\sim$ que tem $\langle m, n \rangle$ como **um membro**.¹ Isto é:

$$[m, n]_\sim = \{ \langle a, b \rangle \in \mathbb{N}^2 : \langle a, b \rangle \sim \langle m, n \rangle \}$$

¹Nota: com a notação introduzida na **Definição 2.11**, escreveríamos $[\langle m, n \rangle]_\sim$ para o mesmo objeto; isso, porém, torna a leitura um pouco mais difícil.

Apresentamos, pois, algumas definições:

$$\begin{aligned}[a, b]_{\sim} + [c, d]_{\sim} &= [a + c, b + d]_{\sim} \\ [a, b]_{\sim} \times [c, d]_{\sim} &= [ac + bd, ad + bc]_{\sim} \\ [a, b]_{\sim} \leq [c, d]_{\sim} &\text{ sse } a + d \leq b + c\end{aligned}$$

(Como é habitual, uso ‘ ab ’ no sentido de ‘ $(a \times b)$ ’, só para tornar os axiomas mais legíveis.) Há, ainda, que garantir que estas definições se comportam como *devem*. Explicitar o que isto significa, e verificá-lo, é trabalhoso; relegamos os pormenores a [Seção 5.6](#). Em resumo, contudo: tudo funciona!

Resta um último ponto. Construimos os inteiros a partir dos números naturais. Isto significa que os naturais *não são, eles próprios, inteiros*. Voltaremos ao significado filosófico disto na [Seção 5.5](#). Do ponto de vista puramente técnico, porém, precisamos de alguma forma de tratar números naturais *como* inteiros. A ideia é simples: para cada $n \in \mathbb{N}$, estipulamos $n_{\mathbb{Z}} = [n, 0]_{\sim}$. Há que confirmar que esta definição é bem comportada, isto é, que para quaisquer $m, n \in \mathbb{N}$

$$\begin{aligned}(m + n)_{\mathbb{Z}} &= m_{\mathbb{Z}} + n_{\mathbb{Z}} \\ (m \times n)_{\mathbb{Z}} &= m_{\mathbb{Z}} \times n_{\mathbb{Z}} \\ m \leq n &\leftrightarrow m_{\mathbb{Z}} \leq n_{\mathbb{Z}}\end{aligned}$$

Mas isto é bastante direto. Por exemplo, para mostrar que a segunda condição se verifica, podemos apoiar-nos no comportamento dos naturais e raciocinar assim:

$$\begin{aligned}(m \times n)_{\mathbb{Z}} &= [m \times n, 0]_{\sim} \\ &= [m \times n + 0 \times 0, m \times 0 + 0 \times n]_{\sim} \\ &= [m, 0]_{\sim} \times [n, 0]_{\sim} \\ &= m_{\mathbb{Z}} \times n_{\mathbb{Z}}\end{aligned}$$

Deixamos como exercício confirmar que as outras duas condições se verificam.

Problema 5.1. Mostre que $(m + n)_{\mathbb{Z}} = m_{\mathbb{Z}} + n_{\mathbb{Z}}$ e $m \leq n \leftrightarrow m_{\mathbb{Z}} \leq n_{\mathbb{Z}}$, para quaisquer $m, n \in \mathbb{N}$.

[content/sets-functions-relations/arithmeticization/rationals.tex](#)

5.2 De $\mathbb{Z}$ para $\mathbb{Q}$

Acabamos de ver como construir os inteiros a partir dos naturais, usando alguma teoria ingênua dos conjuntos. Veremos agora como construir os racionais a partir dos inteiros, de modo muito semelhante. As constatações iniciais são:

[sfr:arith:rat:sec](#)

5.2. DE $\mathbb{Z}$ PARA $\mathbb{Q}$

1. Todo racional pode escrever-se na forma i/j , onde i e j são inteiros, mas j é diferente de zero.
2. A informação codificada numa expressão i/j pode igualmente codificar-se num par ordenado $\langle i, j \rangle$.

A abordagem óbvia seria pensar nos racionais *como* pares ordenados de $\mathbb{Z} \times (\mathbb{Z} \setminus \{0_{\mathbb{Z}}\})$. Como antes, porém, isso seria um pouco demasiado ingênuo, pois queremos $3/2 = 6/4$, mas $\langle 3, 2 \rangle \neq \langle 6, 4 \rangle$. Mais geralmente, quereremos o seguinte:

$$a/b = c/d \text{ sse } a \times d = b \times c$$

Para obter isto, definimos uma *relação de equivalência* em $\mathbb{Z} \times (\mathbb{Z} \setminus \{0_{\mathbb{Z}}\})$ assim:

$$\langle a, b \rangle \sim \langle c, d \rangle \text{ sse } a \times d = b \times c$$

Há que verificar que isto é uma relação de equivalência. É muito parecido com o caso de $\sim$, e deixamo-lo como exercício.

Problema 5.2. Mostre que $\sim$ é uma relação de equivalência.

Mas isto permite-nos dizer o seguinte:

Definição 5.3. Os racionais são as classes de equivalência, sob $\sim$, dos pares de inteiros (cujo segundo elemento é não nulo). Isto é, $\mathbb{Q} = (\mathbb{Z} \times (\mathbb{Z} \setminus \{0_{\mathbb{Z}}\})) / \sim$.

Tal como para os inteiros, queremos também definir algumas operações básicas. Sendo $[i, j]_{\sim}$ a classe de equivalência sob $\sim$ com $\langle i, j \rangle$ como **um membro**, definimos:

$$\begin{aligned} [a, b]_{\sim} + [c, d]_{\sim} &= [ad + bc, bd]_{\sim} \\ [a, b]_{\sim} \times [c, d]_{\sim} &= [ac, bd]_{\sim}. \end{aligned}$$

Para definir $r \leq s$ nestes racionais, usamos o fato de que $r \leq s$ sse $s - r$ não é negativo, isto é, $r - s$ pode escrever-se como i/j com i não negativo e j positivo:

$$[a, b]_{\sim} \leq [c, d]_{\sim} \text{ sse } [c, d]_{\sim} - [a, b]_{\sim} = [i_{\mathbb{Z}}, j_{\mathbb{Z}}]_{\sim}$$

para algum $i \in \mathbb{N}$ e $0 \neq j \in \mathbb{N}$.

Depois há que verificar que estas definições se comportam como *devem*; relegamo-lo a **Seção 5.6**. Mas de fato comportam-se! Por fim, queremos alguma forma de tratar inteiros *como* racionais; logo, para cada $i \in \mathbb{Z}$, estipulamos $i_{\mathbb{Q}} = [i, 1]_{\sim}$. De novo, verificamos na **Seção 5.6** que tudo isto se comporta corretamente.

Problema 5.3. Mostre que $(i + j)_{\mathbb{Q}} = i_{\mathbb{Q}} + j_{\mathbb{Q}}$ e $(i \times j)_{\mathbb{Q}} = i_{\mathbb{Q}} \times j_{\mathbb{Q}}$ e $i \leq j \leftrightarrow i_{\mathbb{Q}} \leq j_{\mathbb{Q}}$, para quaisquer $i, j \in \mathbb{Z}$.

5.3 A Reta Real

O passo seguinte é mostrar como construir os reais a partir dos racionais. Antes disso, há que perceber o que há de *distintivo* nos reais.

sfr:arith:real:
sec

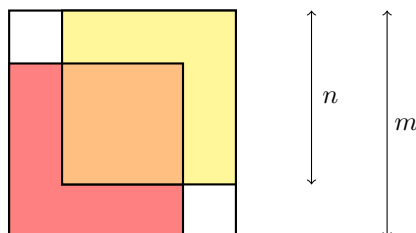
Os reais comportam-se de modo muito parecido com os racionais. (Técnicamente, ambos são exemplos de *corpos ordenados*; para a definição, ver [Definição 5.9](#).) Agora, se resolveu os exercícios de [Capítulo 4](#), saberá que há estritamente mais reais do que racionais, isto é, que $\mathbb{Q} \prec \mathbb{R}$. Cantor foi o primeiro a provar isto. Há, contudo, cerca de dois milênios e meio que se conhecem números irracionais, isto é, reais que não são racionais. De fato:

Teorema 5.4. $\sqrt{2}$ não é racional, isto é, $\sqrt{2} \notin \mathbb{Q}$

sfr:arith:real:
root2irrational

Prova. Suponhamos, por *reductio*, que $\sqrt{2}$ é racional. Então $\sqrt{2} = m/n$ para alguns números naturais m e n . De fato, podemos escolher m e n de modo a que a fração não possa ser mais reduzida. Reorganizando, $m^2 = 2n^2$. A partir daqui, completamos a prova de duas formas:

Primeiro, geometricamente (seguindo Tennenbaum).² Consideremos estes quadrados:



□

Como $m^2 = 2n^2$, a região em que os dois quadrados de lado n se sobrepõem tem a mesma área que a região que nenhum dos dois quadrados cobre; isto é, a área do quadrado cor de laranja é igual à soma das áreas dos dois quadrados não sombreados. Logo, se o quadrado cor de laranja tem lado p e cada quadrado não sombreado tem lado q , então $p^2 = 2q^2$. Mas então $\sqrt{2} = p/q$, com $p < m$, $q < n$ e $p, q \in \mathbb{N}$. Isto contradiz o fato de m e n terem sido escolhidos tão pequenos quanto possível.

Segundo, formalmente. Como $m^2 = 2n^2$, segue-se que m é par. (É fácil mostrar que, se x é ímpar, então x^2 é ímpar.) Logo $m = 2r$, para algum $r \in \mathbb{N}$. Rearranjando, $2r^2 = n^2$, logo n também é par. Assim m e n são ambos pares, e portanto a fração m/n pode ser ainda reduzida. Contradição!

De passagem, esta prova diagramática permite lembrar o material de [Seção 77.4](#). Tennenbaum (1927–2006) era um matemático inequivocamente moderno; mas a prova é inegavelmente bela, completamente rigorosa e apela à intuição geométrica!

²Esta prova é relatada por [Conway \(2006\)](#).

5.4. DE $\mathbb{Q}$ PARA $\mathbb{R}$

Em todo o caso: os reais são “mais extensos” do que os racionais. Nalgum sentido, há “lacunas” nos racionais, e os reais preenchem-nas. Weierstrass apercebeu-se de que isto descreve uma única propriedade dos números reais que os distingue dos racionais, nomeadamente a *propriedade de completude*: *Todo conjunto não vazio de números reais com majorante tem um supremo (menor majorante)*.

É fácil ver que os racionais não têm a propriedade de completude. Por exemplo, considere-se o conjunto dos racionais menores que $\sqrt{2}$, isto é:

$$\{p \in \mathbb{Q} : p^2 < 2 \text{ ou } p < 0\}$$

Este conjunto tem um majorante nos racionais; os seus membros são todos menores que 3, por exemplo. Mas qual é o seu supremo? Queremos dizer “ $\sqrt{2}$ ”; contudo, acabamos de ver que $\sqrt{2}$ não é racional. E não existe racional mínimo entre os que são estritamente maiores que $\sqrt{2}$. Logo, o conjunto tem majorante, mas não tem supremo. Assim, os racionais não satisfazem a propriedade de completude.

Em contraste, o contínuo “moralmente deveria” ter a propriedade de completude. Não queremos apenas que $\sqrt{2}$ seja um número real; queremos preencher todas as “lacunas” da reta racional. Queremos, ainda, que o próprio contínuo não tenha “lacunas”. É exatamente isso que obteremos via completude.

content/sets-functions-relations/arithmetization/cuts.tex

5.4 De $\mathbb{Q}$ para $\mathbb{R}$

sfr:arith:cuts:
sec

Em essência, a propriedade de completude mostra que qualquer ponto α da reta real divide essa reta em duas metades de modo perfeito: numa, α é o supremo; noutra, α é o ínfimo. Para *construir* os números reais a partir dos racionais, Dedekind sugeriu que pensássemos nos reais simplesmente como os *cortes* que particionam os racionais. Isto é, identificamos $\sqrt{2}$ com o *corte* que separa os racionais $< \sqrt{2}$ dos racionais $> \sqrt{2}$.

Vamos pôr isto em ordem. Se cortarmos os racionais em duas metades, podemos identificar de modo único a partição que fizemos considerando apenas a metade *inferior*. Sendo precisos, apresentamos a seguinte definição:

Definição 5.5 (Corte). Um *corte* α é qualquer segmento inicial próprio não vazio dos racionais sem máximo. Isto é, α é um corte sse:

1. *não vazio, próprio*: $\emptyset \neq \alpha \subsetneq \mathbb{Q}$
2. *inicial*: para todos $p, q \in \mathbb{Q}$: se $p < q \in \alpha$ então $p \in \alpha$
3. *sem máximo*: para todo $p \in \alpha$ existe $q \in \alpha$ tal que $p < q$

Então $\mathbb{R}$ é o conjunto dos cortes.

Podemos pois dizer que $\sqrt{2} = \{p \in \mathbb{Q} : p^2 < 2 \text{ ou } p < 0\}$. Claro que há que verificar que isto é de fato um corte, mas relegamo-lo a [Seção 5.6](#).

Como antes, tendo definido certas entidades, precisamos a seguir de definir funções e relações básicas sobre elas. Começamos por uma fácil:

$$\alpha \leq \beta \text{ sse } \alpha \subseteq \beta$$

Esta definição de ordem permite *enunciar* o resultado central, a saber que o conjunto dos cortes tem a propriedade de completude. Por extenso, o enunciado tem esta forma. Se S é um conjunto não vazio de cortes com majorante, então S tem supremo. Com mais pormenor: existe um corte, λ , que é majorante de S , isto é, $(\forall \alpha \in S) \alpha \subseteq \lambda$, e λ é o menor tal corte, isto é, $(\forall \beta \in \mathbb{R})((\forall \alpha \in S) \alpha \subseteq \beta \rightarrow \lambda \subseteq \beta)$. Segue-se a prova do resultado:

Teorema 5.6. *O conjunto dos cortes tem a propriedade de completude.*

*sfr:arith:cuts:
realcompleteness*

Prova. Seja S um conjunto não vazio de cortes com majorante. Seja $\lambda = \bigcup S$. Começamos por alegar que λ é um corte:

1. Como S tem majorante, pelo menos um corte pertence a S , logo $\emptyset \neq \lambda$. Como S é um conjunto de cortes, $\lambda \subseteq \mathbb{Q}$. Como S tem majorante, algum $p \in \mathbb{Q}$ está ausente de todo o corte $\alpha \in S$. Logo $p \notin \lambda$, e portanto $\lambda \subsetneq \mathbb{Q}$.
2. Suponhamos $p < q \in \lambda$. Então existe algum $\alpha \in S$ tal que $q \in \alpha$. Como α é um corte, $p \in \alpha$. Logo $p \in \lambda$.
3. Suponhamos $p \in \lambda$. Então existe algum $\alpha \in S$ tal que $p \in \alpha$. Como α é um corte, existe algum $q \in \alpha$ tal que $p < q$. Logo $q \in \lambda$.

Isto prova a alegação. Além disso, claramente $(\forall \alpha \in S) \alpha \subseteq \bigcup S = \lambda$, isto é, λ é majorante de S . Suponhamos agora que $\beta \in \mathbb{R}$ também é majorante, isto é, $(\forall \alpha \in S) \alpha \subseteq \beta$. Para qualquer $p \in \mathbb{Q}$, se $p \in \lambda$, então existe $\alpha \in S$ tal que $p \in \alpha$, logo $p \in \beta$. Generalizando, $\lambda \subseteq \beta$. Logo λ é o supremo de S . $\square$

Temos pois um conjunto de entidades que satisfaz a propriedade de completude. Uma forma de o dizer é: não há “lacunas” nos nossos cortes. (Assim: tomar novos “cortes” de reais, em vez de racionais, não produziria objetos novos interessantes.)

Segue-se a definição de algumas operações sobre os reais. Começamos por embutir os racionais nos reais estipulando $p_{\mathbb{R}} = \{q \in \mathbb{Q} : q < p\}$ para cada $p \in \mathbb{Q}$. Definimos então:

$$\begin{aligned} \alpha + \beta &= \{p + q : p \in \alpha \wedge q \in \beta\} \\ \alpha \times \beta &= \{p \times q : 0 \leq p \in \alpha \wedge 0 \leq q \in \beta\} \cup 0^{\mathbb{R}} \quad \text{se } \alpha, \beta \geq 0_{\mathbb{R}} \end{aligned}$$

Para tratar os outros casos da multiplicação, definamos primeiro:

$$-\alpha = \{p - q : p < 0 \wedge q \notin \alpha\}$$

5.5. ALGUMAS REFLEXÕES FILOSÓFICAS

e estipulemos:

$$\alpha \times \beta = \begin{cases} -\alpha \times -\beta & \text{se } \alpha < 0_{\mathbb{R}} \text{ e } \beta < 0_{\mathbb{R}} \\ -(-\alpha \times \beta) & \text{se } \alpha < 0_{\mathbb{R}} \text{ e } \beta > 0_{\mathbb{R}} \\ -(\alpha \times -\beta) & \text{se } \alpha > 0_{\mathbb{R}} \text{ e } \beta < 0_{\mathbb{R}} \end{cases}$$

Depois há que verificar que cada uma destas definições produz sempre um corte. E, por fim, há que fazer uma demonstração fácil (mas longa) de que os cortes, assim definidos, se comportam exatamente como devem. Mas relegamos tudo isto a [Seção 5.6](#).

`content/sets-functions-relations/arithmetization/reflections.tex`

5.5 Algumas Reflexões Filosóficas

sfr:arith:ref:
sec

Quanto aos pormenores técnicos, por ora basta. Mas o que foi que alcançaram?

Bem, de forma pouco contestável, deram-nos uma matemática pura *belíssima*. Além disso, houve conquistas conceptuais profundas. Foi um insight profundo perceber que a propriedade de completude exprime a diferença crucial entre os reais e os racionais. Além disso, a construção explícita dos reais, como cortes de Dedekind, põe o objeto de estudo da análise em bases sólidas. Sabemos que a noção de *corpo ordenado completo* é coerente, pois os cortes formam precisamente um tal corpo.

Todavia, importa também levantar algumas reservas quanto a essas conquistas.

Primeiro, não é claro que pensar os reais em termos de cortes seja *mais* rigoroso do que pensá-los nas suas expansões decimais familiares (eventualmente infinitas). Esta última “construção” dos reais tem alguma semelhança com a construção via sequências de Cauchy; mas, de fato, era, essencialmente, conhecida dos matemáticos desde princípios do século XVII (ver [Seção 5.7](#)). O verdadeiro aumento de rigor veio da tomada de consciência de que os reais têm a propriedade de completude; a possibilidade de construir números reais como conjuntos particulares talvez não seja, por si só, tão interessante.

É ainda menos claro que a aritmetização (muito mais fácil) dos inteiros ou dos racionais aumente o rigor nessas áreas. Aqui, vale a pena uma observação simples. Tendo *construído* os inteiros como classes de equivalência de pares ordenados de naturais, e depois construído os racionais como classes de equivalência de pares ordenados de inteiros, e depois construído os reais como conjuntos de racionais, *nos esquecemos de imediato* das construções. Em particular: ninguém quereria *invocar* estas construções no decurso de uma prova matemática (exceto, claro, numa prova de que as construções se comportam como deviam). É muito mais fácil falar de um real diretamente do que falar de algum conjunto de conjuntos de conjuntos de conjuntos de conjuntos de conjuntos de conjuntos de naturais.

O mais duvidoso, porém, é que estas definições nos digam o que os inteiros, racionais ou reais *são, do ponto de vista metafísico*. Isto é, é duvidoso que

os reais (digamos) *sejam* certos conjuntos (de conjuntos de conjuntos...). A principal barreira a tal visão é que a construção poderia ter sido feita de muitas maneiras diferentes. No caso dos reais, há construções genuinamente diferentes e interessantes (ver [Seção 5.7](#)). Mas eis uma forma trivial de obter construções diferentes: como em [Seção 2.2](#), poderíamos ter definido pares ordenados de modo ligeiramente diferente; se tivéssemos usado essa noção alternativa de par ordenado, as nossas construções teriam funcionado exatamente tão bem como funcionaram, mas teríamos acabado com objetos diferentes. Assim, há muitas construções rivais, em teoria dos conjuntos, dos inteiros, racionais e reais. Seria, ademais, arbitrário (e embaraçoso) afirmar que os inteiros (digamos) são *estes* conjuntos, e não *aqueles*. (Como na [Seção 2.2](#), isto é um caso de um argumento tornado célebre por [Benacerraf 1965](#).)

Um outro ponto merece ser levantado: há algo de bastante *estranho* nas nossas construções. Começamos pelos números naturais. Depois, construímos os inteiros e construímos “o 0 dos inteiros”, isto é, $[0, 0]_{\sim}$. Mas $0 \neq [0, 0]_{\sim}$. De fato, dadas as nossas construções, *nenhum* natural é um inteiro. Isto parece fortemente contraintuitivo. Na [Seção 1.3](#), afirmamos, sem grande argumento, que $\mathbb{N} \subseteq \mathbb{Q}$. Agora, se as construções nos dissessem exatamente *o que* são os números, esta afirmação seria trivialmente falsa.

Recuando um pouco, onde chegamos? Trabalhando numa teoria ingênua dos conjuntos e tomando os naturais como dados, podemos *tratar* inteiros, racionais e reais como certos conjuntos. Nesse sentido, podemos *embutir* as teorias destas entidades numa teoria dos conjuntos. Mas a importância filosófica deste embutimento não é assim tão direta.

Claro que isto não é a última palavra! O ponto é apenas este: mostrar que a aritmetização dos reais *é de* profunda significação filosófica exigiria argumento filosófico adicional.

`content/sets-functions-relations/arithmetization/checking-details.tex`

5.6 Anéis e Corpos Ordenados

Ao longo deste capítulo, afirmamos que certas definições se comportam “como devem”. Neste apêndice técnico, explicitaremos o que queremos dizer com isso e (esboçaremos como) mostrar que as definições se comportam “corretamente”.

`sfr:arith:check:
sec`

Na [Seção 5.1](#), definimos adição e multiplicação em $\mathbb{Z}$. Queremos mostrar que, assim definidas, conferem a $\mathbb{Z}$ a estrutura que “quereríamos” que tivesse. Em particular, a estrutura em causa é a de um anel comutativo.

Definição 5.7. Um *anel comutativo* é um conjunto S , equipado com elementos

5.6. ANÉIS E CORPOS ORDENADOS

distintos 0 e 1 e operações $+$ e $\times$, que satisfazem estas oito condições:

<i>Associatividade</i>	$a + (b + c) = (a + b) + c$ $(a \times b) \times c = a \times (b \times c)$
<i>Comutatividade</i>	$a + b = b + a$ $a \times b = b \times a$
<i>Neutros</i>	$a + 0 = a$ $a \times 1 = a$
<i>Simétrico aditivo</i>	$(\exists b \in S) 0 = a + b$
<i>Distributividade</i>	$a \times (b + c) = (a \times b) + (a \times c)$

Implicitamente, todas estas fórmulas estão quantificadas universalmente sobre S . Note-se que os elementos 0 e 1 aqui não têm de ser os números naturais com o mesmo nome.

Logo, para verificar que os inteiros formam um anel comutativo, basta verificar estas oito condições. Nenhuma é *difícil* de estabelecer, mas o trabalho é um pouco laborioso. Por exemplo, eis como provar *associatividade*, no caso da adição:

Prova. Fixemos $i, j, k \in \mathbb{Z}$. Então existem $a_1, b_1, a_2, b_2, a_3, b_3 \in \mathbb{N}$ tais que $i = [a_1, b_1]$ e $j = [a_2, b_2]$ e $k = [a_3, b_3]$. (Por legibilidade, escrevemos “[x, y]” em vez de “[x, y] $\sim$ ”; faremos o mesmo ao longo desta secção.) Então:

$$\begin{aligned}
 i + (j + k) &= [a_1, b_1] + ([a_2, b_2] + [a_3, b_3]) \\
 &= [a_1, b_1] + [a_2 + a_3, b_2 + b_3] \\
 &= [a_1 + (a_2 + a_3), b_1 + (b_2 + b_3)] \\
 &= [(a_1 + a_2) + a_3, (b_1 + b_2) + b_3] \\
 &= [a_1 + a_2, b_1 + b_2] + [a_3, b_3] \\
 &= ([a_1, b_1] + [a_2, b_2]) + [a_3, b_3] \\
 &= (i + j) + k
 \end{aligned}$$

apoiando-nos livremente no comportamento da adição em $\mathbb{N}$. □

Do mesmo modo, eis uma prova do *simétrico aditivo*:

Prova. Fixemos $i \in \mathbb{Z}$, de modo que $i = [a, b]$ para alguns $a, b \in \mathbb{N}$. Seja $j = [b, a] \in \mathbb{Z}$. Apoiando-nos no comportamento dos naturais, $(a+b)+0 = 0+(a+b)$, logo $\langle a+b, b+a \rangle \sim_{\mathbb{Z}} \langle 0, 0 \rangle$ por definição, e portanto $[a+b, b+a] = [0, 0] = 0_{\mathbb{Z}}$. Assim $i + j = [a, b] + [b, a] = [a+b, b+a] = [0, 0] = 0_{\mathbb{Z}}$. □

E eis uma prova da *distributividade*:

Prova. Como acima, fixemos $i = [a_1, b_1]$ e $j = [a_2, b_2]$ e $k = [a_3, b_3]$. Então:

$$\begin{aligned}
 i \times (j + k) &= [a_1, b_1] \times ([a_2, b_2] + [a_3, b_3]) \\
 &= [a_1, b_1] \times [a_2 + a_3, b_2 + b_3] \\
 &= [a_1(a_2 + a_3) + b_1(b_2 + b_3), a_1(b_2 + b_3) + b_1(a_2 + a_3)] \\
 &= [a_1a_2 + a_1a_3 + b_1b_2 + b_1b_3, a_1b_2 + a_1b_3 + a_2b_1 + a_3b_1] \\
 &= [a_1a_2 + b_1b_2, a_1b_2 + a_2b_1] + [a_1a_3 + b_1b_3, a_1b_3 + a_3b_1] \\
 &= ([a_1, b_1] \times [a_2, b_2]) + ([a_1, b_1] \times [a_3, b_3]) \\
 &= (i \times j) + (i \times k)
 \end{aligned}$$

□

Deixamos como exercício provar as cinco condições restantes. Tendo feito isso, mostramos que $\mathbb{Z}$ constitui um anel comutativo, isto é, que a adição e a multiplicação (assim definidas) se comportam como devem.

Problema 5.4. Mostre que $\mathbb{Z}$ é um anel comutativo.

Mas a tarefa não termina. Para além de definir adição e multiplicação em $\mathbb{Z}$, definimos uma relação de ordem, $\leq$, e há que verificar que esta se comporta como deve. Com mais pormenor, há que mostrar que $\mathbb{Z}$ constitui um *anel ordenado*.³

Definição 5.8. Um *anel ordenado* é um anel comutativo que está também equipado com uma relação de ordem total, $\leq$, tal que:

$$\begin{aligned}
 a \leq b &\rightarrow a + c \leq b + c \\
 (a \leq b \wedge 0 \leq c) &\rightarrow a \times c \leq b \times c
 \end{aligned}$$

Problema 5.5. Mostre que $\mathbb{Z}$ é um anel ordenado.

Como antes, é laborioso mas rotineiro mostrar que $\mathbb{Z}$, tal como construído, é um anel ordenado. Deixamo-lo ao leitor.

Isto trata dos inteiros. Mas agora precisamos de mostrar coisas muito parecidas para os racionais. Em particular, há que mostrar que os racionais formam um *corpo ordenado*, com as nossas definições de $+$, $\times$ e $\leq$:

Definição 5.9. Um *corpo ordenado* é um anel ordenado que também satisfaz: sfr:arith:check:orderedfield

$$\text{Inverso multiplicativo} \quad (\forall a \in S \setminus \{0\})(\exists b \in S)a \times b = 1$$

Uma vez mostrado que $\mathbb{Z}$ constitui um anel ordenado, é fácil mas laborioso mostrar que $\mathbb{Q}$ constitui um corpo ordenado.

Problema 5.6. Mostre que $\mathbb{Q}$ é um corpo ordenado.

³Como na [Definição 2.16](#), uma ordem total é uma relação reflexiva, transitiva, anti-simétrica e conexa. No contexto de relações de ordem, a conexividade é por vezes chamada *tricotomia*, pois, para quaisquer a e b , temos $a \leq b \vee a = b \vee a \geq b$.

5.6. ANÉIS E CORPOS ORDENADOS

Tendo tratado dos inteiros e dos racionais, só restam os reais. Em particular, há que mostrar que $\mathbb{R}$ constitui um corpo ordenado *completo*, isto é, um corpo ordenado com a propriedade de completude. Agora, **Teorema 5.6** estabeleceu que $\mathbb{R}$ tem a propriedade de completude. Contudo, cumpre ainda percorrer a tediosa verificação de que $\mathbb{R}$ é um corpo ordenado.

Antes de nos lançarmos a *esse* exercício laborioso, há que confirmar algumas coisas mais “imediatas”. Por exemplo, precisamos de garantia de que $\alpha + \beta$, tal como definida, é de fato um *corte*, para quaisquer cortes α e β . Segue-se uma prova desse fato:

Prova. Como α e β são ambos cortes, $\alpha + \beta = \{p + q : p \in \alpha \wedge q \in \beta\}$ é um subconjunto próprio não vazio de $\mathbb{Q}$. Suponhamos agora $x < p + q$ para alguns $p \in \alpha$ e $q \in \beta$. Então $x - p < q$, logo $x - p \in \beta$, e $x = p + (x - p) \in \alpha + \beta$. Logo $\alpha + \beta$ é um segmento inicial de $\mathbb{Q}$. Finalmente, para qualquer $p + q \in \alpha + \beta$, como α e β são cortes, existem $p_1 \in \alpha$ e $q_1 \in \beta$ tais que $p < p_1$ e $q < q_1$; logo $p + q < p_1 + q_1 \in \alpha + \beta$; logo $\alpha + \beta$ não tem máximo. $\square$

Esforços semelhantes permitem verificar que $\alpha - \beta$ e $\alpha \times \beta$ e $\alpha \div \beta$ são cortes (no último caso, ignorando o caso em que β é o corte zero). De novo, deixamo-lo ao leitor.

Problema 5.7. Mostre que $\mathbb{R}$ é um corpo ordenado.

Mas eis um pequeno fio solto a atar. Em **Seção 5.4**, sugerimos que podemos tomar $\sqrt{2} = \{p \in \mathbb{Q} : p < 0 \text{ ou } p^2 < 2\}$. Mas há que mostrar que este conjunto é de fato um *corte*. Segue-se uma prova desse fato:

Prova. Claramente isto é um segmento inicial próprio não vazio dos racionais; basta pois mostrar que não tem máximo. Em particular, basta mostrar que, sendo p um racional positivo com $p^2 < 2$ e $q = \frac{2p+2}{p+2}$, se tem $p < q$ e $q^2 < 2$. Para ver que $p < q$, basta notar:

$$\begin{aligned} p^2 &< 2 \\ p^2 + 2p &< 2 + 2p \\ p(p+2) &< 2 + 2p \\ p &< \frac{2+2p}{p+2} = q \end{aligned}$$

Para ver que $q^2 < 2$, basta notar:

$$\begin{aligned} p^2 &< 2 \\ 2p^2 + 4p + 2 &< p^2 + 4p + 4 \\ 4p^2 + 8p + 4 &< 2(p^2 + 4p + 4) \\ (2p+2)^2 &< 2(p+2)^2 \\ \frac{(2p+2)^2}{(p+2)^2} &< 2 \\ q^2 &< 2 \end{aligned}$$

$\square$

`content/sets-functions-relations/arithmetization/cauchy.tex`

5.7 Apêndice: os Reais como Sequências de Cauchy

Na [Seção 5.4](#), construímos os reais como cortes de Dedekind. Nesta seção, explicamos uma construção alternativa. Apóia-se na definição, devida a Cauchy, do que hoje chamamos *sequência de Cauchy*; mas o uso desta definição para *construir* os reais deve-se a outros autores do século XIX, nomeadamente Weierstrass, Heine, Méray e Cantor. (Para uma boa história, ver [O'Connor and Robertson 2005](#).)

[sfr:arith:cauchy:sec](#)

Antes de chegarmos ao século XIX, vale a pena considerar Simon Stevin (1548–1620). Em resumo, Stevin apercebeu-se de que podemos pensar cada real em termos da sua expansão decimal. Assim, mesmo um número irracional, como $\sqrt{2}$, tem uma expansão decimal agradável, que começa:

1.41421356237...

É muito fácil modelar expansões decimais na teoria dos conjuntos: basta considerá-las como funções $d: \mathbb{N} \rightarrow \mathbb{N}$, onde $d(n)$ é o n -ésimo algarismo decimal que nos interessa. Depois precisamos de um pequeno ajuste, para tratar a parte do número real que vem antes da vírgula decimal (aqui, apenas 1). Precisaremos também de outro ajuste (uma relação de equivalência) para garantir que, por exemplo, $0.999\dots = 1$. Mas não é difícil apresentar uma construção perfeitamente rigorosa dos números reais, à maneira de Stevin, dentro da teoria dos conjuntos.

Stevin não é o nosso foco. (Para mais sobre Stevin, ver [Katz and Katz 2012](#).) Mas eis um pensamento estreitamente relacionado. Em vez de tratar diretamente a expansão decimal de $\sqrt{2}$, podemos antes considerar uma *sequência* de aproximações racionais cada vez mais precisas a $\sqrt{2}$, olhando para as expansões cada vez mais precisas:

1, 1.4, 1.414, 1.4142, 1.41421, ...

A ideia de que os reais podem ser considerados via “aproximações cada vez melhores” nos dá a base para outra sequência de insights, semelhante às constatações que retivemos ao construir $\mathbb{Q}$ a partir de $\mathbb{Z}$, ou $\mathbb{Z}$ a partir de $\mathbb{N}$. Os insights básicos são estes:

1. Todo o real pode escrever-se como uma expansão decimal (eventualmente infinita).
2. A informação codificada por uma expansão decimal (eventualmente infinita) pode igualmente codificar-se por uma sequência de números racionais.
3. Uma sequência de números racionais pode pensar-se como uma função de $\mathbb{N}$ para $\mathbb{Q}$; basta definir $f(n)$ como o n -ésimo racional na sequência.

5.7. APÊNDICE: OS REAIS COMO SEQUÊNCIAS DE CAUCHY

Claro que nem *qualquer* função de $\mathbb{N}$ para $\mathbb{Q}$ nos dará um número real. Por exemplo, considere-se esta função:

$$f(n) = \begin{cases} 1 & \text{se } n \text{ é ímpar} \\ 0 & \text{se } n \text{ é par} \end{cases}$$

Essencialmente, a preocupação é que a sequência $0, 1, 0, 1, 0, 1, 0, \dots$ não parece “convergir” para nenhum real. Logo: para garantir que consideramos sequências que de fato convergem para algum real, precisamos de restringir a atenção a sequências que tenham algum *limite*.

Já encontramos a ideia de limite em [Seção 77.2](#). Mas não podemos usar *exatamente* a mesma definição que ali. A expressão “ $(\forall \varepsilon > 0)$ ” envolvia tacitamente quantificação sobre os números reais; e estávamos a considerar limites de funções nos reais; logo, invocar essa definição seria valer-nos dos números reais — e eram precisamente eles que estávamos a tentar *construir*. Felizmente, podemos trabalhar com uma ideia estreitamente relacionada com a de limite.

sfr:arith:cauchy:
def:CauchySequence

Definição 5.10. Uma função $f : \mathbb{N} \rightarrow \mathbb{Q}$ é uma *sequência de Cauchy* sse, para qualquer $\varepsilon \in \mathbb{Q}$ positivo, se tem $(\exists \ell \in \mathbb{N})(\forall m, n > \ell)|f(m) - f(n)| < \varepsilon$.

A ideia geral de limite é a mesma de antes: se se quer um certo nível de precisão (medido por ε), há uma “região” onde olhar (qualquer entrada maior que ℓ). E é fácil ver que a nossa sequência $1, 1.4, 1.414, 1.4142, 1.41421, \dots$ tem limite: se se quer aproximar $\sqrt{2}$ com erro inferior a $1/10^n$, basta olhar para qualquer entrada depois da n -ésima.

O pensamento óbvio seria então dizer que um número real é simplesmente qualquer sequência de Cauchy. Mas, como nas construções de $\mathbb{Z}$ e $\mathbb{Q}$, isso seria demasiado ingênuo: para um dado número real, várias sequências de Cauchy diferentes “indicam” esse número. Uma forma simples de ver isto é a seguinte. Dada uma sequência de Cauchy f , defina-se g como exatamente a mesma função que f , exceto que $g(0) \neq f(0)$. Como as duas sequências coincidem em todo o lado depois do primeiro número, quereremos, em última análise, dizer que têm o mesmo limite, no sentido de [Definição 5.10](#), e portanto devem ser pensadas como “definindo” o mesmo real. Logo, devemos pensar nestas sequências de Cauchy como o mesmo número real.

Consequentemente, voltamos a precisar de definir uma relação de equivalência sobre as sequências de Cauchy, e identificar os números reais com *classes* de equivalência. Precisamos primeiro da ideia de uma função que tende para 0 no limite. Para qualquer função $h : \mathbb{N} \rightarrow \mathbb{Q}$, digamos que h *tende para* 0 sse, para qualquer $\varepsilon \in \mathbb{Q}$ positivo, se tem que $(\exists \ell \in \mathbb{N})(\forall n > \ell)|h(n)| < \varepsilon$.⁴ Além disso, sendo f e g funções $\mathbb{N} \rightarrow \mathbb{Q}$, definamos $(f - g)(n) = f(n) - g(n)$. Definimos agora:

$$f \approx g \text{ sse } (f - g) \text{ tende para } 0.$$

Há que verificar que $\approx$ é uma relação de equivalência; e de fato é. Podemos então, se quisermos, definir os reais como as classes de equivalência, sob $\approx$, de todas as sequências de Cauchy de $\mathbb{N} \rightarrow \mathbb{Q}$.

⁴Compare-se com a definição de $\lim_{x \rightarrow \infty} f(x) = 0$ em [Seção 77.2](#).

Problema 5.8. Seja $f(n) = 0$ para todo n . Seja $g(n) = \frac{1}{(n+1)^2}$. Mostre que ambas são sequências de Cauchy e que o limite de ambas as funções é 0, de modo que também $f \sim_{\mathbb{R}} g$.

Feito isto, como de costume escrevemos $[f]_{\sim}$ para a classe de equivalência que tem f como **um membro**. Porém, para manter a legibilidade, no que se segue omitimos o subscrito e escrevemos apenas $[f]$. Estipulamos também que, para cada $q \in \mathbb{Q}$, se tem $q_{\mathbb{R}} = [c_q]$, onde c_q é a função constante $c_q(n) = q$ para todo $n \in \mathbb{N}$. Definimos depois relações e operações básicas sobre os reais, por exemplo:

$$\begin{aligned}[f] + [g] &= [(f + g)] \\ [f] \times [g] &= [(f \times g)]\end{aligned}$$

onde $(f + g)(n) = f(n) + g(n)$ e $(f \times g)(n) = f(n) \times g(n)$. Claro que também há que verificar que cada uma de $(f + g)$, $(f - g)$ e $(f \times g)$ é sequência de Cauchy quando f e g o são; mas o são, e deixamo-lo ao leitor.

Por fim, definimos uma noção de ordem. Dizemos que $[f]$ é *positivo* sse tanto $[f] \neq 0_{\mathbb{Q}}$ como $(\exists \ell \in \mathbb{N})(\forall n > \ell) 0 < f(n)$. Depois dizemos $[f] < [g]$ sse $[(g - f)]$ é positivo. Há que verificar que isto está bem definido (isto é, que não depende da escolha da função “representante” na classe de equivalência). Mas tendo feito isto, é bastante fácil mostrar que obtemos as propriedades algébricas certas; isto é:

Teorema 5.11. *As sequências de Cauchy constituem um corpo ordenado.*

*sfr:arith:cauchy:
thm:cauchyorderedfield*

Prova. Exercício. □

Problema 5.9. Mostre que as sequências de Cauchy constituem um corpo ordenado.

É mais difícil provar que os reais, assim construídos, têm a propriedade de completude, pelo que daremos a prova.

Teorema 5.12. *Todo o conjunto não vazio de sequências de Cauchy com majorante tem supremo (menor majorante).*

Esboço de prova. Seja S um conjunto não vazio de sequências de Cauchy com majorante. Então existe algum $p \in \mathbb{Q}$ tal que $p_{\mathbb{R}}$ é majorante de S . Seja $r \in S$; então existe algum $q \in \mathbb{Q}$ tal que $q_{\mathbb{R}} < r$. Logo, se existe supremo de S , este está entre $q_{\mathbb{R}}$ e $p_{\mathbb{R}}$ (inclusive).

Nos aproximaremos do supremo, abordando-o simultaneamente por baixo e por cima. Em particular, definimos duas funções, $f, g: \mathbb{N} \rightarrow \mathbb{Q}$, com a ideia de que f se irá aproximar do supremo por cima e g por baixo. Começamos por definir:

$$\begin{aligned}f(0) &= p \\ g(0) &= q\end{aligned}$$

Então, sendo $a_n = \frac{f(n)+g(n)}{2}$, definamos:⁵

$$f(n+1) = \begin{cases} a_n & \text{se } (\forall h \in S)[h] \leq (a_n)_{\mathbb{R}} \\ f(n) & \text{caso contrário} \end{cases}$$

$$g(n+1) = \begin{cases} a_n & \text{se } (\exists h \in S)[h] \geq (a_n)_{\mathbb{R}} \\ g(n) & \text{caso contrário} \end{cases}$$

Tanto f como g são seqüências de Cauchy. (Isto verifica-se com relativa facilidade, mas deixamo-lo como exercício.) Note-se que a função $(f - g)$ tende para 0, pois a diferença entre f e g reduz-se à metade a cada passo. Logo $[f] = [g]$.

Mostramos primeiro que $[f]$ é majorante de S , isto é, que $(\forall h \in S)[h] \leq [f]$. (Invocaremos **Teorema 5.11** ao longo do argumento.) Seja $h \in S$ e suponhamos, por *reductio*, que $[f] < [h]$, de modo que $0_{\mathbb{R}} < [(h - f)]$. Como f é uma seqüência de Cauchy monotonamente decrescente, existe algum $n \in \mathbb{N}$ tal que $[(c_{f(n)} - f)] < [(h - f)]$. Logo:

$$(f(n))_{\mathbb{R}} = [c_{f(n)}] < [f] + [(h - f)] = [h],$$

o que contradiz o fato de que, por construção, $[h] \leq (f(n))_{\mathbb{R}}$.

Mostramos a seguir que $[f] = [g]$ é o supremo de S . Seja j qualquer seqüência de Cauchy e suponhamos $[j] < [g]$. Raciocinando como acima (usando o fato de g ser *crescente*), existe $n \in \mathbb{N}$ tal que $[j] < (g(n))_{\mathbb{R}}$. Mas, por construção, existe $h \in S$ tal que $(g(n))_{\mathbb{R}} \leq [h]$, logo $[j] < [h]$ e portanto $[j]$ não é majorante de S . $\square$

Capítulo 6

Conjuntos Infinitos

Este capítulo sobre conjuntos infinitos foi retirado de *Open Set Theory*, de Tim Button.

`content/sets-functions-relations/infinite/hilberts-hotel.tex`

⁵Isto é uma definição recursiva. Mas ainda *não* demos qualquer razão para crer que definições recursivas são legítimas.

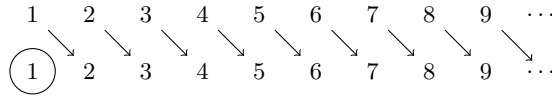
6.1 Hotel de Hilbert

O conjunto dos números naturais é obviamente infinito. Logo, se não quisermos *valer-nos* dos números naturais, o nosso primeiro passo é caracterizar um conjunto infinito em termos que não exijam mencionar os próprios números naturais. Eis uma abordagem agradável, apresentada por Hilbert numa conferência de 1924. Ele nos pede que imaginemos

sfr:infinite:hilbert:
sec

[...] um hotel com um número finito de quartos. Todos esses quartos devem estar ocupados por exatamente um hóspede. Se os hóspedes trocam de quarto de algum modo, [mas] de tal forma que cada quarto continua a conter no máximo uma pessoa, então nenhum quarto ficará livre, e o dono do hotel não pode, deste modo, criar um novo lugar para um hóspede que acaba de chegar [... ¶ ...]

Agora estipulamos que o hotel tem infinitos quartos numerados 1, 2, 3, 4, 5, ..., cada um ocupado por exatamente um hóspede. Logo que chega um novo hóspede, o dono só precisa de mudar cada um dos hóspedes antigos para o quarto associado ao número imediatamente superior, e o quarto 1 ficará livre para o recém-chegado.



(publicado em Hilbert 2013, 730; a nossa tradução)

O ponto crucial é que o Hotel de Hilbert tem infinitos quartos; e podemos tomar a sua explicação como definição do que significa dizer isto. De fato, foi esta a abordagem de Dedekind (apresentada aqui, claro, com enorme anacronismo; a definição de Dedekind é de 1888):

Definição 6.1. Um conjunto A é *infinito de Dedekind* sse existe uma *injeção* de A para um subconjunto próprio de A . Isto é, existe algum $o \in A$ e uma *injeção* $f: A \rightarrow A$ tal que $o \notin \text{Im}(f)$.

sfr:infinite:hilbert:
defn:DedekindInfinite

content/sets-functions-relations/infinite/dedekind-algebra.tex

6.2 Álgebras de Dedekind

Não queremos apenas que os números naturais sejam infinitos; queremos que tenham certas propriedades (algébricas): precisam de comportar-se bem sob adição, multiplicação, e assim por diante.

sfr:infinite:dedekind:
sec

A ideia de Dedekind foi tomar como primitiva a noção de *função sucessor* e depois caracterizar os números como aqueles que têm as seguintes propriedades:

6.2. ÁLGEBRAS DE DEDEKIND

1. Existe um número, 0, que não é sucessor de nenhum número
isto é, $0 \notin \text{Im}(s)$
isto é, $\forall x \ s(x) \neq 0$
2. Números distintos têm sucessores distintos
isto é, s é uma **injeção**
isto é, $\forall x \forall y (s(x) = s(y) \rightarrow x = y)$
3. Todo o número se obtém de 0 por aplicações repetidas da função sucessor.

sfr:infinite:dedekind:
repeatedapplication

As duas primeiras condições são fáceis de lidar com, usando lógica de primeira ordem (ver acima). Mas não podemos tratar (3) apenas com lógica de primeira ordem. O avanço de Dedekind foi reformular a condição (3), em termos de teoria dos conjuntos, assim:

- 3'. Os números naturais são o menor conjunto que é *fechado para a função sucessor*: isto é, se aplicarmos s a qualquer **membro** do conjunto, obtemos outro **membro** do conjunto.

Mas precisamos de explicitar isto com calma.

sfr:infinite:dedekind:
Closure

Definição 6.2. Para qualquer função f , o conjunto X é *f-fechado* sse $(\forall x \in X) f(x) \in X$. Definimos agora, para qualquer o :

$$\text{clo}_f(o) = \bigcap \{X : o \in X \text{ e } X \text{ é } f\text{-fechado}\}$$

Logo $\text{clo}_f(o)$ é a interseção de todos os conjuntos f -fechados que têm o como **um membro**. Intuitivamente, então, $\text{clo}_f(o)$ é o conjunto f -fechado *menor* que tem o como **um membro**. O resultado seguinte torna preciso esse pensamento intuitivo;

sfr:infinite:dedekind:
closureproperties

Lema 6.3. Para qualquer função f e qualquer $o \in A$:

sfr:infinite:dedekind:
closurehaselem

1. $o \in \text{clo}_f(o)$; e

sfr:infinite:dedekind:
closureclosed

2. $\text{clo}_f(o)$ é f -fechado; e

sfr:infinite:dedekind:
closuresmallest

3. se X é f -fechado e $o \in X$, então $\text{clo}_f(o) \subseteq X$

Prova. Note-se que existe pelo menos um conjunto f -fechado com o como **um membro**, nomeadamente $\text{Im}(f) \cup \{o\}$. Logo $\text{clo}_f(o)$, a interseção de *todos* os tais conjuntos, existe. Cumpre agora verificar (1)–(3).

Quanto a (1): $o \in \text{clo}_f(o)$ pois é uma interseção de conjuntos que todos têm o como **um membro**.

Quanto a (2): suponhamos $x \in \text{clo}_f(o)$. Se $o \in X$ e X é f -fechado, então $x \in X$, e agora $f(x) \in X$ pois X é f -fechado. Logo $f(x) \in \text{clo}_f(o)$.

Quanto a (3): de modo geral, se $X \in C$ então $\bigcap C \subseteq X$. □

Com isto, podemos dizer:

Definição 6.4. Uma *álgebra de Dedekind* é um conjunto A juntamente com uma função $f: A \rightarrow A$ e algum $o \in A$ tal que:

1. $o \notin \text{Im}(f)$
2. f é uma **injeção**
3. $A = \text{clo}_f(o)$

sfr:infinite:dedekind:
ded:proper
sfr:infinite:dedekind:
ded:injection
sfr:infinite:dedekind:
ded:closure

Como $A = \text{clo}_f(o)$, o resultado anterior nos diz que A é o menor conjunto f -fechado com o como **um membro**. Claramente uma álgebra de Dedekind é infinita de Dedekind; basta olhar para as cláusulas (1) e (2) da definição. Mas o fato mais relevante é que qualquer conjunto infinito de Dedekind pode ser transformado numa álgebra de Dedekind.

Teorema 6.5. *Se existe um conjunto infinito de Dedekind, então existe uma álgebra de Dedekind.*

sfr:infinite:dedekind:
thm:DedekindInfiniteAlgebra

Prova. Seja D infinito de Dedekind. Logo existe uma injeção $g: D \rightarrow D$ e um elemento $o \in D \setminus \text{Im}(g)$. Seja agora $A = \text{clo}_g(o)$; por **Lema 6.3**, A existe e $o \in A$. Seja $f = g|_A$. Mostraremos que A, f, o formam uma álgebra de Dedekind.

Quanto a (1): $o \notin \text{Im}(g)$ e $\text{Im}(f) \subseteq \text{Im}(g)$, logo $o \notin \text{Im}(f)$.

Quanto a (2): g é injetiva em D ; logo $f \subseteq g$ tem de ser injetiva.

Quanto a (3): por **Lema 6.3**, A é g -fechado; *a fortiori*, A é f -fechado. Logo $\text{clo}_f(o) \subseteq A$ por **Lema 6.3**. Como também $\text{clo}_f(o)$ é f -fechado e $f = g|_A$, segue-se que $\text{clo}_f(o)$ é g -fechado. Logo $A \subseteq \text{clo}_f(o)$ por **Lema 6.3**. $\square$

content/sets-functions-relations/infinite/dedekind-induction.tex

6.3 Álgebras de Dedekind e Indução Aritmética

Crucialmente, agora, uma álgebra de Dedekind—de fato, *qualquer* álgebra de Dedekind—servirá como substituto dos números naturais. Isto deve-se à seguinte consequência trivial:

sfr:infinite:induction:
sec

Teorema 6.6 (Indução aritmética). *Sejam N, s, o uma álgebra de Dedekind. Então, para qualquer conjunto X :*

sfr:infinite:induction:
thm:dedinfiniteinduction

$$\text{se } o \in X \text{ e } (\forall n \in N \cap X) s(n) \in X, \text{ então } N \subseteq X.$$

Prova. Pela definição de álgebra de Dedekind, $N = \text{clo}_s(o)$. Ora, se tanto $o \in X$ como $(\forall n \in N)(n \in X \rightarrow s(n) \in X)$, então $N = \text{clo}_s(o) \subseteq X$. $\square$

Como a indução é característica dos números naturais, o ponto é este. Dado qualquer conjunto infinito de Dedekind, podemos formar uma álgebra de Dedekind e usar essa álgebra como substituto dos números naturais.

Admitidamente, **Teorema 6.6** formula a indução em termos *de teoria dos conjuntos*. Mas podemos facilmente pôr o princípio em termos mais familiares:

6.4. “PROVA” DE DEDEKIND

sfr:infinite:induction:
natinductionscheme

Corolário 6.7. *Sejam N, s, o uma álgebra de Dedekind. Então, para qualquer fórmula $\varphi(x)$, que pode ter parâmetros:*

se $\varphi(o)$ e $(\forall n \in N)(\varphi(n) \rightarrow \varphi(s(n)))$, então $(\forall n \in N)\varphi(n)$

Prova. Seja $X = \{n \in N : \varphi(n)\}$, e use-se agora **Teorema 6.6** □

Neste resultado, falamos de uma fórmula “ter parâmetros”. O que isto significa, em traços grosseiros, é que, para quaisquer objetos $c_1, \dots, c_k$, podemos trabalhar com $\varphi(x, c_1, \dots, c_k)$. Com mais precisão, podemos enunciar o resultado sem mencionar “parâmetros” do seguinte modo. Para qualquer fórmula $\varphi(x, v_1, \dots, v_k)$, cujas variáveis livres são todas as exibidas, temos:

$$\begin{aligned} & \forall v_1 \dots \forall v_k ((\varphi(o, v_1, \dots, v_k) \wedge \\ & (\forall x \in N)(\varphi(x, v_1, \dots, v_k) \rightarrow \varphi(s(x), v_1, \dots, v_k))) \rightarrow \\ & (\forall x \in N)\varphi(x, v_1, \dots, v_k)) \end{aligned}$$

Evidentemente, falar em “ter parâmetros” pode tornar a leitura muito mais fácil. (Na **Parte XV**, usaremos este expediente com frequência.)

Voltando às álgebras de Dedekind: dada qualquer álgebra de Dedekind, podemos também definir as habituais funções aritméticas de adição, multiplicação e exponenciação. Isto não é trivial, porém, e envolve a técnica de *definição recursiva*. É uma técnica que introduziremos e justificaremos muito mais tarde, e num contexto bem mais geral. (Os entusiastas podem querer voltar a isto depois de **Capítulo 70**, ou talvez ler um tratamento alternativo, como **Potter 2004**, pp. 95–8.) Mas, onde N, s, o formam uma álgebra de Dedekind, acabaremos por poder estipular o seguinte:

$$\begin{array}{lll} a + o = a & a \times o = o & a^o = s(o) \\ a + s(b) = s(a + b) & a \times s(b) = (a \times b) + a & a^{s(b)} = a^b \times a \end{array}$$

e mostrar que estas definições se comportam como se espera.

content/sets-functions-relations/infinite/dedekinds-proof.tex

6.4 “Prova” de Dedekind da existência de um conjunto infinito

sfr:infinite:dedekindsproof:
sec

Neste capítulo, apresentamos um tratamento em teoria dos conjuntos dos números naturais, em termos de álgebras de Dedekind. Em **Seção 5.5**, refletimos sobre o significado filosófico da aritmetização da análise (entre outras coisas). Cumpre agora refletir sobre o significado do que aqui alcançamos.

Ao longo de **Capítulo 5**, tomamos os números naturais como dados e os usamos para construir os inteiros, os racionais e os reais, de modo explícito. Neste capítulo, não demos uma construção explícita dos números naturais.

Mostramos apenas que, *dado qualquer conjunto infinito de Dedekind*, podemos definir um conjunto que se comportará como queremos que $\mathbb{N}$ se comporte.

Obviamente, pois, não podemos pretender ter respondido a uma questão metafísica, como *quais os objetos que são os números naturais*. Mas isso é bom. Afinal, na [Seção 5.5](#), sublinhamos que estaríamos errados se pensássemos na definição de $\mathbb{R}$ como conjunto de cortes de Dedekind como uma *descoberta*, em vez de uma estipulação conveniente. A observação crucial é que os cortes de Dedekind exemplificam as propriedades matemáticas centrais dos números reais. Assim também aqui: a observação crucial é que *qualquer* álgebra de Dedekind exemplifica as propriedades matemáticas centrais dos números naturais. (De fato, Dedekind insistiu neste ponto ao provar que todas as álgebras de Dedekind são *isomorfas* ([1888](#), Theorems 132–3). Não surpreende, então, que muitos “estruturalistas” contemporâneos cite Dedekind como precursor.)

Além disso, mostramos como embutir a teoria dos números naturais numa teoria ingênua simples dos conjuntos, que ainda permanece bastante informal, mas que (aparentemente) não assume os números naturais como dados. Assim, podemos estar no caminho de realizar o próprio projeto ambicioso de Dedekind, que ele explicou assim:

Na ciência nada que seja provável deve ser acreditado sem prova. Embora esta exigência pareça razoável, não posso considerá-la satisfeita sequer nos métodos mais recentes de lançar os fundamentos da ciência mais simples; a saber, aquela parte da lógica que trata da teoria dos números. Ao falar de aritmética (álgebra, análise) como mera parte da lógica, quero insinuar que considero o conceito de número totalmente independente das noções ou intuições de espaço e tempo—que o considero antes um produto imediato das leis puras do pensamento. ([Dedekind, 1888](#), preface)

A ideia audaciosa de Dedekind é esta. Acabamos de mostrar como construir os números naturais usando só (ingenuamente) a teoria dos conjuntos. Em [Capítulo 5](#), vimos como construir os reais dados os números naturais e alguma teoria dos conjuntos. Logo, talvez, “a aritmética (álgebra, análise)” acabe por ser “meramente parte da lógica” (no sentido alargado que Dedekind dá à palavra “lógica”).

Essa é a ideia. Mas esperemos um momento. A nossa construção de uma álgebra de Dedekind (o nosso substituto dos números naturais) é condicional à existência de um conjunto infinito de Dedekind. (Basta olhar para trás, para [Teorema 6.5](#).) A menos que a existência de um conjunto infinito de Dedekind possa ser estabelecida via “lógica” ou “as leis puras do pensamento”, o projeto estagna.

Então, *pode* a existência de um conjunto infinito de Dedekind ser estabelecida pelas “leis puras do pensamento”? Eis o esforço de Dedekind:

O meu próprio reino de pensamentos, isto é, a totalidade S de todas as coisas que podem ser objetos do meu pensamento, é infinita. Pois, se s designa um elemento de S , então o pensamento s' de

6.5. APÊNDICE: PROVANDO SCHRÖDER-BERNSTEIN

que s pode ser objeto do meu pensamento é ele próprio um elemento de S . Se considerarmos isto como uma imagem $\varphi(s)$ do elemento s , então ... S é infinito de Dedekind, como queríamos ter provado. (Dedekind, 1888, §66)

Isto é algo de espantoso de encontrar no meio de um livro que em grande parte consiste em provas matemáticas altamente rigorosas. Duas observações valem a pena.

Primeiro: esta “prova” mal tem o que hoje reconheceríamos como caráter “matemático”. Fala de objetos psicológicos (pensamentos), e apenas de objetos *possíveis*.

Segundo: pelo menos tal como apresentamos as álgebras de Dedekind, esta “prova” tem uma lacuna técnica direta. Se o argumento de Dedekind é bem-sucedido, estabelece só que há infinitas coisas (nomeadamente, infinitos pensamentos). Mas Dedekind também precisa de nos dar uma razão para ver S como um único *conjunto*, com infinitos *membros*, em vez de pensar em S como *algumas coisas* (no plural).

O fato de Dedekind não ver aqui uma lacuna pode sugerir que o seu uso da palavra “totalidade” não corresponde exatamente ao *nosso* uso da palavra “conjunto”.¹ Mas isto não seria demasiado surpreendente. O projeto que perseguimos nos últimos dois capítulos—uma “construção” dos naturais e, a partir deles, uma “construção” dos inteiros, reais e racionais—foi todo ele feito de modo ingênuo. Nos apoiamos neste conjunto, ou naquele conjunto, sempre que precisamos, sem enunciar muitos princípios gerais sobre exatamente que conjuntos existem, e quando. Mas sabemos que precisamos de *alguns* princípios gerais, pois caso contrário caímos no Paradoxo de Russell.

Chegou a hora de superarmos a nossa ingenuidade.

content/sets-functions-relations/infinite/card-sb.tex

6.5 Apêndice: Provando Schröder-Bernstein

sfr:infinite:card-sb:
sec

Antes de deixarmos a teoria ingênua dos conjuntos, há ainda uma última prova ingênua (mas sofisticada!) a considerar. Trata-se de uma prova de Schröder-Bernstein (Teorema 4.25): se $A \preceq B$ e $B \preceq A$ então $A \approx B$; isto é, dados *injeções* $f: A \rightarrow B$ e $g: B \rightarrow A$, existe *uma bijeção* $h: A \rightarrow B$.

Neste capítulo, seguimos a noção dedekindiana de *fechos*. De fato, Dedekind deu uma prova muito elegante de Schröder-Bernstein com esta noção, e aqui a apresentamos. A prova segue de perto Potter (2004, pp. 157–8), se quiser um tratamento ligeiramente diferente mas essencialmente semelhante. Uma breve pesquisa na Internet também convencerá o leitor de que este é um teorema—tal como a irracionalidade de $\sqrt{2}$ —para o qual existem *muitas* provas interessantes e distintas.

¹De fato, temos outras razões para pensar que não correspondia; ver Potter (2004, p. 23).

CAPÍTULO 6. CONJUNTOS INFINITOS

Usando notação semelhante à de [Definição 6.2](#), seja

$$\text{Clo}_f(B) = \bigcap \{X : B \subseteq X \text{ e } X \text{ é } f\text{-fechado}\}$$

para cada conjunto B e função f . Assim definido, $\text{Clo}_f(B)$ é o menor conjunto f -fechado que contém B , no sentido de que:

Lema 6.8. *Para qualquer função f , e qualquer B :*

1. $B \subseteq \text{Clo}_f(B)$; e
2. $\text{Clo}_f(B)$ é f -fechado; e
3. se X é f -fechado e $B \subseteq X$, então $\text{Clo}_f(B) \subseteq X$.

*sfr:infinite:card-sb:
Closureprops
sfr:infinite:card-sb:
Closurehaselem
sfr:infinite:card-sb:
Closureclosed
sfr:infinite:card-sb:
Closuresmallest*

Prova. Exatamente como no [Lema 6.3](#). □

Precisamos de mais um fato para chegar a Schröder-Bernstein:

Proposição 6.9. *Se $A \subseteq B \subseteq C$ e $A \approx C$, então $A \approx B \approx C$.*

*sfr:infinite:card-sb:
sbhelper*

Prova. Dada uma [bijeção](#) $f: C \rightarrow A$, seja $F = \text{Clo}_f(C \setminus B)$ e defina-se uma função g com domínio C do seguinte modo:

$$g(x) = \begin{cases} f(x) & \text{se } x \in F \\ x & \text{caso contrário} \end{cases}$$

Mostraremos que g é [uma bijeção](#) de C para B , donde seguirá que $g \circ f^{-1}: A \rightarrow B$ é [uma bijeção](#), completando a prova.

Primeiro alegamos que, se $x \in F$ mas $y \notin F$, então $g(x) \neq g(y)$. Por reductio, suponhamos o contrário, de modo que $y = g(y) = g(x) = f(x)$. Como $x \in F$ e F é f -fechado por [Lema 6.8](#), temos $y = f(x) \in F$, contradição.

Suponhamos agora $g(x) = g(y)$. Então, pelo que precede, $x \in F$ sse $y \in F$. Se $x, y \in F$, então $f(x) = g(x) = g(y) = f(y)$, logo $x = y$ pois f é [uma bijeção](#). Se $x, y \notin F$, então $x = g(x) = g(y) = y$. Logo g é [uma injeção](#).

Resta mostrar que $\text{Im}(g) = B$. Fixemos $x \in B \subseteq C$. Se $x \notin F$, então $g(x) = x$. Se $x \in F$, então $x = f(y)$ para algum $y \in F$, pois caso contrário $F \setminus \{x\}$ seria f -fechado e estenderia $C \setminus B$, o que é impossível por [Lema 6.8](#); agora $g(y) = f(y) = x$. □

Finalmente, eis a prova do resultado principal. Recordemos que, dada uma função h e um conjunto D , definimos $h[D] = \{h(x) : x \in D\}$.

Prova de Schröder-Bernstein. Sejam $f: A \rightarrow B$ e $g: B \rightarrow A$ [injeções](#). Como $f[A] \subseteq B$, temos $g[f[A]] \subseteq g[B] \subseteq A$. Além disso, $g \circ f: A \rightarrow g[f[A]]$ é [uma injeção](#), pois tanto g como f o são; e de fato $g \circ f$ é [uma bijeção](#), apenas pela forma como definimos o seu contradomínio. Logo $g[f[A]] \approx A$, e portanto, por [Proposição 6.9](#), existe [uma bijeção](#) $h: A \rightarrow g[B]$. Além disso, g^{-1} é [uma bijeção](#) $g[B] \rightarrow B$. Logo $g^{-1} \circ h: A \rightarrow B$ é [uma bijeção](#). □

Parte II

Propositional Logic

This part contains material on classical propositional logic. The first chapter is relatively rudimentary and just lists definitions and results, many proofs are not carried out but are left as exercises. The material on proof systems and the completeness theorem is included from the part on first-order logic, with the “FOL” tag set to false. This leaves out everything related to predicates, terms, and quantifiers, and replaces talk of **estruturas** $\mathfrak{M}$ with talk about **valorações** v .

It is planned to expand this part to include more detail, and to add further topics and results, such as truth-functional completeness.

Capítulo 7

Syntax and Semantics

This is a very quick summary of definitions only. It should be expanded to provide a gentle intro to proofs by induction on formulas, with lots more examples.

-NoValue-

`content/propositional-logic/syntax-and-semantics/introduction.tex`

7.1 Introduction

pl:syn:int:
sec Propositional logic deals with **fórmulas** that are built from **variáveis proposicionais** using the propositional connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$. Intuitively, a **variável proposicional** p stands for a sentence or proposition that is true or

false. Whenever the “truth value” of the *variável proposicional* in a *fórmula* is determined, so is the truth value of any *fórmulas* formed from them using propositional connectives. We say that propositional logic is *truth functional*, because its semantics is given by functions of truth values. In particular, in propositional logic we leave out of consideration any further determination of truth and falsity, p.ex., whether something is necessarily true rather than just contingently true, or whether something is known to be true, or whether something is true now rather than was true or will be true. We only consider two truth values true ($\mathbb{T}$) and false ($\mathbb{F}$), and so exclude from discussion the possibility that a statement may be neither true nor false, or only half true. We also concentrate only on connectives where the truth value of a *fórmula* built from them is completely determined by the truth values of its parts (and not, say, on its meaning). In particular, whether the truth value of conditionals in English is truth functional in this sense is contentious. The material conditional $\rightarrow$ is; other logics deal with conditionals that are not truth functional.

In order to develop the theory and metatheory of truth-functional propositional logic, we must first define the syntax and semantics of its expressions. We will describe one way of constructing *fórmulas* from *variáveis proposicionais* using the connectives. Alternative definitions are possible. Other systems will choose different symbols, will select different sets of connectives as primitive, and will use parentheses differently (or even not at all, as in the case of so-called Polish notation). What all approaches have in common, though, is that the formation rules define the set of *fórmulas* *inductively*. If done properly, every expression can result essentially in only one way according to the formation rules. The inductive definition resulting in expressions that are *uniquely readable* means we can give meanings to these expressions using the same method—inductive definition.

Giving the meaning of expressions is the domain of semantics. The central concept in semantics for propositional logic is that of satisfaction in a *valoração*. A *valoração* $\mathbf{v}$ assigns truth values $\mathbb{T}$, $\mathbb{F}$ to the *variáveis proposicionais*. Any *valoração* determines a truth value $\bar{\mathbf{v}}(\varphi)$ for any *fórmula* φ . A *fórmula* is satisfied in a *valoração* $\mathbf{v}$ iff $\bar{\mathbf{v}}(\varphi) = \mathbb{T}$ —we write this as $\mathbf{v} \models \varphi$. This relation can also be defined by induction on the structure of φ , using the truth functions for the logical connectives to define, say, satisfaction of $\varphi \wedge \psi$ in terms of satisfaction (or not) of φ and ψ .

On the basis of the satisfaction relation $\mathbf{v} \models \varphi$ for sentences we can then define the basic semantic notions of tautology, entailment, and satisfiability. A *fórmula* is a tautology, $\models \varphi$, if every *valoração* satisfies it, O.s., $\bar{\mathbf{v}}(\varphi) = \mathbb{T}$ for any $\mathbf{v}$. It is entailed by a set of *fórmulas*, $\Gamma \models \varphi$, if every *valoração* that satisfies all the *fórmulas* in Γ also satisfies φ . And a set of *fórmulas* is satisfiable if some *valoração* satisfies all *fórmulas* in it at the same time. Because *fórmulas* are inductively defined, and satisfaction is in turn defined by induction on the structure of *fórmulas*, we can use induction to prove properties of our semantics and to relate the semantic notions defined.

-NoValue-

7.2. PROPOSITIONAL FÓRMULAS

content/propositional-logic/syntax-and-semantics/formulas.tex

7.2 Propositional Fórmulas

pl:syn:fml: **Fórmulas** of propositional logic are built up from *variáveis proposicionais*, the
sec propositional constant $\perp$ and the propositional constant $\top$ using *logical connectives*.

1. A infinito enumerável set At_0 of *variáveis proposicionais* $p_0, p_1, \dots$
2. The propositional constant for *falsidade* $\perp$.
3. The propositional constant for *verdade* $\top$.
4. The logical connectives: $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\rightarrow$ (*condicional*), $\leftrightarrow$ (*bicondicional*)
5. Punctuation marks: $(,)$, and the comma.

We denote this language of propositional logic by $\mathcal{L}_0$.

You may be familiar with different terminology and symbols than the ones [intro](#) we use above. Logic texts (and teachers) commonly use either $\sim$, $\neg$, and $!$ for “negation”, $\wedge$, $\cdot$, and $\&$ for “conjunction”. Commonly used symbols for the “conditional” or “implication” are $\rightarrow$, $\Rightarrow$, and $\supset$. Symbols for “biconditional,” “bi-implication,” or “(material) equivalence” are $\leftrightarrow$, $\Leftrightarrow$, and $\equiv$. The $\perp$ symbol is variously called “falsity,” “falsum,” “absurdity,” or “bottom.” The $\top$ symbol is variously called “truth,” “verum,” or “top.”

pl:syn:fml: **Definição 7.1 (Formula).** The set $\text{Frm}(\mathcal{L}_0)$ of *fórmulas* of propositional lo-
defn:formulas gic is defined inductively as follows:

1. $\perp$ is an atomic *fórmula*.
2. $\top$ is an atomic *fórmula*.
3. Every *variável proposicional* p_i is an atomic *fórmula*.
4. If φ is a *fórmula*, then $\neg\varphi$ is a *fórmula*.
5. If φ and ψ are *fórmulas*, then $(\varphi \wedge \psi)$ is a *fórmula*.
6. If φ and ψ are *fórmulas*, then $(\varphi \vee \psi)$ is a *fórmula*.
7. If φ and ψ are *fórmulas*, then $(\varphi \rightarrow \psi)$ is a *fórmula*.
8. If φ and ψ are *fórmulas*, then $(\varphi \leftrightarrow \psi)$ is a *fórmula*.
9. Nothing else is a *fórmula*.

explanation

The definition of *fórmulas* is an *inductive definition*. Essentially, we construct the set of *fórmulas* in infinitely many stages. In the initial stage, we pronounce all atomic formulas to be formulas; this corresponds to the first few cases of the definition, O.s., the cases for $\top$, $\perp$, p_i . “Atomic *fórmula*” thus means any *fórmula* of this form.

The other cases of the definition give rules for constructing new *fórmulas* out of *fórmulas* already constructed. At the second stage, we can use them to construct *fórmulas* out of atomic *fórmulas*. At the third stage, we construct new formulas from the atomic formulas and those obtained in the second stage, and so on. A *fórmula* is anything that is eventually constructed at such a stage, and nothing else.

When writing a formula $(\psi * \chi)$ constructed from ψ , χ using a two-place connective $*$, we will often leave out the outermost pair of parentheses and write simply $\psi * \chi$.

Definição 7.2 (Syntactic identity). The symbol $\equiv$ expresses syntactic identity between strings of symbols, O.s., $\varphi \equiv \psi$ iff φ and ψ are strings of symbols of the same length and which contain the same symbol in each place.

The $\equiv$ symbol may be flanked by strings obtained by concatenation, p.ex., $\varphi \equiv (\psi \vee \chi)$ means: the string of symbols φ is the same string as the one obtained by concatenating an opening parenthesis, the string ψ , the $\vee$ symbol, the string χ , and a closing parenthesis, in this order. If this is the case, then we know that the first symbol of φ is an opening parenthesis, φ contains ψ as a substring (starting at the second symbol), that substring is followed by $\vee$, etc.

-NoValue-

content/propositional-logic/syntax-and-semantics/preliminaries.tex

7.3 Preliminaries

Teorema 7.3 (Principle of induction on *fórmulas*). If some property P holds for all the atomic *fórmulas* and is such that

pl:syn:pre:
sec
pl:syn:pre:
thm:induction

1. it holds for $\neg\varphi$ whenever it holds for φ ;
2. it holds for $(\varphi \wedge \psi)$ whenever it holds for φ and ψ ;
3. it holds for $(\varphi \vee \psi)$ whenever it holds for φ and ψ ;
4. it holds for $(\varphi \rightarrow \psi)$ whenever it holds for φ and ψ ;
5. it holds for $(\varphi \leftrightarrow \psi)$ whenever it holds for φ and ψ ;

then P holds for all *fórmulas*.

7.3. PRELIMINARIES

Prova. Let S be the collection of all fórmulas with property P . Clearly $S \subseteq \text{Frm}(\mathcal{L}_0)$. S satisfies all the conditions of Definição 7.1: it contains all atomic fórmulas and is closed under the operadores lógicos. $\text{Frm}(\mathcal{L}_0)$ is the smallest such class, so $\text{Frm}(\mathcal{L}_0) \subseteq S$. So $\text{Frm}(\mathcal{L}_0) = S$, and every formula has property P . $\square$

pl:syn:pre: prop:balanced **Proposição 7.4.** Any fórmula in $\text{Frm}(\mathcal{L}_0)$ is balanced, in that it has as many left parentheses as right ones.

Problema 7.1. Prove Proposição 7.4

pl:syn:pre: prop:noinit **Proposição 7.5.** No proper initial segment of a fórmula is a fórmula.

Problema 7.2. Prove Proposição 7.5

Proposição 7.6 (Unique Readability). Any fórmula φ in $\text{Frm}(\mathcal{L}_0)$ has exactly one parsing as one of the following

1. $\perp$.
2. $\top$.
3. p_n for some $p_n \in \text{At}_0$.
4. $\neg\psi$ for some fórmula ψ .
5. $(\psi \wedge \chi)$ for some fórmulas ψ and χ .
6. $(\psi \vee \chi)$ for some fórmulas ψ and χ .
7. $(\psi \rightarrow \chi)$ for some fórmulas ψ and χ .
8. $(\psi \leftrightarrow \chi)$ for some fórmulas ψ and χ .

Moreover, this parsing is unique.

Prova. By induction on φ . For instance, suppose that φ has two distinct readings as $(\psi \rightarrow \chi)$ and $(\psi' \rightarrow \chi')$. Then ψ and ψ' must be the same (or else one would be a proper initial segment of the other); so if the two readings of φ are distinct it must be because χ and χ' are distinct readings of the same sequence of symbols, which is impossible by the inductive hypothesis. $\square$

Definição 7.7 (Uniform Substitution). If φ and ψ are fórmulas, and p_i is a variável proposicional, then $\varphi[\psi/p_i]$ denotes the result of replacing each occurrence of p_i by an occurrence of ψ in φ ; similarly, the simultaneous substitution of $p_1, \dots, p_n$ by fórmulas $\psi_1, \dots, \psi_n$ is denoted by $\varphi[\psi_1/p_1, \dots, \psi_n/p_n]$.

Problema 7.3. For each of the five fórmulas below determine whether the fórmula can be expressed as a substitution $\varphi[\psi/p_i]$ where φ is (i) p_0 ; (ii) $(\neg p_0 \wedge p_1)$; and (iii) $((\neg p_0 \rightarrow p_1) \wedge p_2)$. In each case specify the relevant substitution.

1. p_1
2. $(\neg p_0 \wedge p_0)$
3. $((p_0 \vee p_1) \wedge p_2)$
4. $\neg((p_0 \rightarrow p_1) \wedge p_2)$
5. $((\neg(p_0 \rightarrow p_1) \rightarrow (p_0 \vee p_1)) \wedge \neg(p_0 \wedge p_1))$

Problema 7.4. Give a mathematically rigorous definition of $\varphi[\psi/p]$ by induction.

-NoValue-

`content/propositional-logic/syntax-and-semantics/formation-sequences.tex`

7.4 Formation Sequences

Defining **fórmulas** via an inductive definition, and the complementary technique of proving properties of **fórmulas** via induction, is an elegant and efficient approach. However, it can also be useful to consider a more bottom-up, step-by-step approach to the construction of **fórmulas**, which we do here using the notion of a *formation sequence*.

pl:syn:fseq:
sec

Definição 7.8 (Formation sequences for formulas). A finite sequence $\langle \varphi_0, \varphi_1, \dots, \varphi_n \rangle$ of strings of symbols from the language $\mathcal{L}_0$ is a *formation sequence* for φ if $\varphi \equiv \varphi_n$ and for all $i \leq n$, either φ_i is an atomic formula or there exist $j, k < i$ such that one of the following holds:

pl:syn:fseq:
defn:fseq-frn

1. $\varphi_i \equiv \neg \varphi_j$.
2. $\varphi_i \equiv (\varphi_j \wedge \varphi_k)$.
3. $\varphi_i \equiv (\varphi_j \vee \varphi_k)$.
4. $\varphi_i \equiv (\varphi_j \rightarrow \varphi_k)$.
5. $\varphi_i \equiv (\varphi_j \leftrightarrow \varphi_k)$.

Exemplo 7.9.

$$\langle p_0, p_1, (p_1 \wedge p_0), \neg(p_1 \wedge p_0) \rangle$$

is a formation sequence of $\neg(p_1 \wedge p_0)$, as is

$$\langle p_0, p_1, p_0, (p_1 \wedge p_0), (p_0 \rightarrow p_1), \neg(p_1 \wedge p_0) \rangle.$$

As can be seen from the second example, formation sequences may contain ‘junk’: formulas which are redundant or do not contribute to the construction.

7.4. FORMATION SEQUENCES

*pl:syn:fseq:
prop:formed*

Proposição 7.10. Every *fórmula* φ in $\text{Frm}(\mathcal{L}_0)$ has a formation sequence.

Prova. Suppose φ is atomic. Then the sequence $\langle \varphi \rangle$ is a formation sequence for φ . Now suppose that ψ and χ have formation sequences $\langle \psi_0, \dots, \psi_n \rangle$ and $\langle \chi_0, \dots, \chi_m \rangle$ respectively.

1. If $\varphi \equiv \neg\psi$, then $\langle \psi_0, \dots, \psi_n, \neg\psi_n \rangle$ is a formation sequence for φ .
2. If $\varphi \equiv (\psi \wedge \chi)$, then $\langle \psi_0, \dots, \psi_n, \chi_0, \dots, \chi_m, (\psi_n \wedge \chi_m) \rangle$ is a formation sequence for φ .
3. If $\varphi \equiv (\psi \vee \chi)$, then $\langle \psi_0, \dots, \psi_n, \chi_0, \dots, \chi_m, (\psi_n \vee \chi_m) \rangle$ is a formation sequence for φ .
4. If $\varphi \equiv (\psi \rightarrow \chi)$, then $\langle \psi_0, \dots, \psi_n, \chi_0, \dots, \chi_m, (\psi_n \rightarrow \chi_m) \rangle$ is a formation sequence for φ .
5. If $\varphi \equiv (\psi \leftrightarrow \chi)$, then $\langle \psi_0, \dots, \psi_n, \chi_0, \dots, \chi_m, (\psi_n \leftrightarrow \chi_m) \rangle$ is a formation sequence for φ .

By the principle of induction on *fórmulas*, every *fórmula* has a formation sequence. $\square$

We can also prove the converse. This is important because it shows that our two ways of defining formulas are equivalent: they give the same results. It also means that we can prove theorems about formulas by using ordinary induction on the length of formation sequences.

*pl:syn:fseq:
lem:fseq-init*

Lema 7.11. Suppose that $\langle \varphi_0, \dots, \varphi_n \rangle$ is a formation sequence for φ_n , and that $k \leq n$. Then $\langle \varphi_0, \dots, \varphi_k \rangle$ is a formation sequence for φ_k .

Prova. Exercise. $\square$

*pl:syn:fseq:
thm:fseq-frm-equiv*

Teorema 7.12. $\text{Frm}(\mathcal{L}_0)$ is the set of all strings of symbols in the language $\mathcal{L}_0$ with a formation sequence.

Prova. Let F be the set of all strings of symbols in the language $\mathcal{L}_0$ that have a formation sequence. We have seen in **Proposição 7.10** that $\text{Frm}(\mathcal{L}_0) \subseteq F$, so now we prove the converse.

Suppose φ has a formation sequence $\langle \varphi_0, \dots, \varphi_n \rangle$. We prove that $\varphi \in \text{Frm}(\mathcal{L}_0)$ by strong induction on n . Our induction hypothesis is that every string of symbols with a formation sequence of length $m < n$ is in $\text{Frm}(\mathcal{L}_0)$. By the definition of a formation sequence, either φ_n is atomic or there must exist $j, k < n$ such that one of the following is the case:

1. $\varphi_n \equiv \neg\varphi_j$.
2. $\varphi_n \equiv (\varphi_j \wedge \varphi_k)$.
3. $\varphi_n \equiv (\varphi_j \vee \varphi_k)$.

$$4. \varphi_n \equiv (\varphi_j \rightarrow \varphi_k).$$

$$5. \varphi_n \equiv (\varphi_j \leftrightarrow \varphi_k).$$

Now we reason by cases. If φ_n is atomic then $\varphi_n \in \text{Frm}(\mathcal{L}_0)$. Suppose instead that $\varphi \equiv (\varphi_j \wedge \varphi_k)$. By [Lema 7.11](#), $\langle \varphi_0, \dots, \varphi_j \rangle$ and $\langle \varphi_0, \dots, \varphi_k \rangle$ are formation sequences for φ_j and φ_k respectively. Since these are proper initial subsequences of the formation sequence for φ , they both have length less than n . Therefore by the induction hypothesis, φ_j and φ_k are in $\text{Frm}(\mathcal{L}_0)$, and so by the definition of [a fórmula](#), so is $(\varphi_j \wedge \varphi_k)$. The other cases follow by parallel reasoning. $\square$

-NoValue-

[content/propositional-logic/syntax-and-semantics/valuations-sat.tex](#)

7.5 Valorações and Satisfaction

Definição 7.13 (Valorações). Let $\{\mathbb{T}, \mathbb{F}\}$ be the set of the two truth values, “true” and “false.” A [valoração](#) for $\mathcal{L}_0$ is a function $\mathbf{v}$ assigning either $\mathbb{T}$ or $\mathbb{F}$ to the [variáveis proposicionais](#) of the language, O.s., $\mathbf{v}: \text{At}_0 \rightarrow \{\mathbb{T}, \mathbb{F}\}$.

pl:syn:val:
sec

Definição 7.14. Given [a valoração](#) $\mathbf{v}$, define the evaluation function $\bar{\mathbf{v}}: \text{Frm}(\mathcal{L}_0) \rightarrow \{\mathbb{T}, \mathbb{F}\}$ inductively by:

$$\begin{aligned} \bar{\mathbf{v}}(\perp) &= \mathbb{F}; \\ \bar{\mathbf{v}}(\top) &= \mathbb{T}; \\ \bar{\mathbf{v}}(p_n) &= \mathbf{v}(p_n); \\ \bar{\mathbf{v}}(\neg\varphi) &= \begin{cases} \mathbb{T} & \text{if } \bar{\mathbf{v}}(\varphi) = \mathbb{F}; \\ \mathbb{F} & \text{otherwise.} \end{cases} \\ \bar{\mathbf{v}}(\varphi \wedge \psi) &= \begin{cases} \mathbb{T} & \text{if } \bar{\mathbf{v}}(\varphi) = \mathbb{T} \text{ and } \bar{\mathbf{v}}(\psi) = \mathbb{T}; \\ \mathbb{F} & \text{if } \bar{\mathbf{v}}(\varphi) = \mathbb{F} \text{ or } \bar{\mathbf{v}}(\psi) = \mathbb{F}. \end{cases} \\ \bar{\mathbf{v}}(\varphi \vee \psi) &= \begin{cases} \mathbb{T} & \text{if } \bar{\mathbf{v}}(\varphi) = \mathbb{T} \text{ or } \bar{\mathbf{v}}(\psi) = \mathbb{T}; \\ \mathbb{F} & \text{if } \bar{\mathbf{v}}(\varphi) = \mathbb{F} \text{ and } \bar{\mathbf{v}}(\psi) = \mathbb{F}. \end{cases} \\ \bar{\mathbf{v}}(\varphi \rightarrow \psi) &= \begin{cases} \mathbb{T} & \text{if } \bar{\mathbf{v}}(\varphi) = \mathbb{F} \text{ or } \bar{\mathbf{v}}(\psi) = \mathbb{T}; \\ \mathbb{F} & \text{if } \bar{\mathbf{v}}(\varphi) = \mathbb{T} \text{ and } \bar{\mathbf{v}}(\psi) = \mathbb{F}. \end{cases} \\ \bar{\mathbf{v}}(\varphi \leftrightarrow \psi) &= \begin{cases} \mathbb{T} & \text{if } \bar{\mathbf{v}}(\varphi) = \bar{\mathbf{v}}(\psi); \\ \mathbb{F} & \text{if } \bar{\mathbf{v}}(\varphi) \neq \bar{\mathbf{v}}(\psi). \end{cases} \end{aligned}$$

[explanation](#)

The clauses correspond to the following truth tables:

7.5. VALORAÇÕES AND SATISFACTION

φ	$\neg\varphi$
T	F
F	T

φ	ψ	$\varphi \wedge \psi$
T	T	T
T	F	F
F	T	F
F	F	F

φ	ψ	$\varphi \vee \psi$
T	T	T
T	F	T
F	T	T
F	F	F

φ	ψ	$\varphi \rightarrow \psi$
T	T	T
T	F	F
F	T	T
F	F	T

φ	ψ	$\varphi \leftrightarrow \psi$
T	T	T
T	F	F
F	T	F
F	F	T

Problema 7.5. Consider adding to $\mathcal{L}_0$ a ternary connective $\diamond$ with evaluation given by

$$\bar{\mathbf{v}}(\diamond(\varphi, \psi, \chi)) = \begin{cases} \bar{\mathbf{v}}(\psi) & \text{if } \bar{\mathbf{v}}(\varphi) = \mathbb{T}; \\ \bar{\mathbf{v}}(\chi) & \text{if } \bar{\mathbf{v}}(\varphi) = \mathbb{F}. \end{cases}$$

Write down the truth table for this connective.

pl:syn:val:
thm:LocalDetermination

Teorema 7.15 (Local Determination). Suppose that $\mathbf{v}_1$ and $\mathbf{v}_2$ are *valorações* that agree on the *variáveis proposicionais* occurring in φ , O.s., $\mathbf{v}_1(p_n) = \mathbf{v}_2(p_n)$ whenever p_n occurs in some *fórmula* φ . Then $\bar{\mathbf{v}}_1$ and $\bar{\mathbf{v}}_2$ also agree on φ , O.s., $\bar{\mathbf{v}}_1(\varphi) = \bar{\mathbf{v}}_2(\varphi)$.

Prova. By induction on φ . □

pl:syn:val:
defn:satisfaction

Definição 7.16 (Satisfaction). We can inductively define the notion of *satisfaction* of a *fórmula* φ by a *valoração* $\mathbf{v}$, $\mathbf{v} \models \varphi$, as follows. (We write $\mathbf{v} \not\models \varphi$ to mean “not $\mathbf{v} \models \varphi$.”)

1. $\varphi \equiv \perp$: $\mathbf{v} \not\models \varphi$.
2. $\varphi \equiv \top$: $\mathbf{v} \models \varphi$.
3. $\varphi \equiv p_i$: $\mathbf{v} \models \varphi$ iff $\mathbf{v}(p_i) = \mathbb{T}$.
4. $\varphi \equiv \neg\psi$: $\mathbf{v} \models \varphi$ iff $\mathbf{v} \not\models \psi$.
5. $\varphi \equiv (\psi \wedge \chi)$: $\mathbf{v} \models \varphi$ iff $\mathbf{v} \models \psi$ and $\mathbf{v} \models \chi$.
6. $\varphi \equiv (\psi \vee \chi)$: $\mathbf{v} \models \varphi$ iff $\mathbf{v} \models \psi$ or $\mathbf{v} \models \chi$ (or both).
7. $\varphi \equiv (\psi \rightarrow \chi)$: $\mathbf{v} \models \varphi$ iff $\mathbf{v} \not\models \psi$ or $\mathbf{v} \models \chi$ (or both).
8. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\mathbf{v} \models \varphi$ iff either both $\mathbf{v} \models \psi$ and $\mathbf{v} \models \chi$, or neither $\mathbf{v} \models \psi$ nor $\mathbf{v} \models \chi$.

If Γ is a set of *fórmulas*, $\mathbf{v} \models \Gamma$ iff $\mathbf{v} \models \varphi$ for every $\varphi \in \Gamma$.

Proposição 7.17. $\mathbf{v} \models \varphi$ iff $\bar{\mathbf{v}}(\varphi) = \mathbb{T}$.

*pl:syn:val:
prop:sat-value*

Prova. By induction on φ . □

Problema 7.6. Prove **Proposição 7.17**

-NoValue-

`content/propositional-logic/syntax-and-semantics/semantic-notions.tex`

7.6 Semantic Notions

We define the following semantic notions:

*pl:syn:sem:
sec*

Definição 7.18. 1. A fórmula φ is *satisfiable* if for some $\mathbf{v}$, $\mathbf{v} \models \varphi$; it is *unsatisfiable* if for no $\mathbf{v}$, $\mathbf{v} \models \varphi$;

2. A fórmula φ is a *tautology* if $\mathbf{v} \models \varphi$ for all *valorações* $\mathbf{v}$;

3. A fórmula φ is *contingent* if it is satisfiable but not a tautology;

4. If Γ is a set of fórmulas, $\Gamma \models \varphi$ (“ Γ entails φ ”) if and only if $\mathbf{v} \models \varphi$ for every *valoração* $\mathbf{v}$ for which $\mathbf{v} \models \Gamma$.

5. If Γ is a set of fórmulas, Γ is *satisfiable* if there is a *valoração* $\mathbf{v}$ for which $\mathbf{v} \models \Gamma$, and Γ is *unsatisfiable* otherwise.

Problema 7.7. For each of the following four fórmulas determine whether it is (a) satisfiable, (b) tautology, and (c) contingent.

1. $(p_0 \rightarrow (\neg p_1 \rightarrow \neg p_0))$.
2. $((p_0 \wedge \neg p_1) \rightarrow (\neg p_0 \wedge p_2)) \leftrightarrow ((p_2 \rightarrow p_0) \rightarrow (p_0 \rightarrow p_1))$.
3. $(p_0 \leftrightarrow p_1) \rightarrow (p_2 \leftrightarrow \neg p_1)$.
4. $((p_0 \leftrightarrow (\neg p_1 \wedge p_2)) \vee (p_2 \rightarrow (p_0 \leftrightarrow p_1)))$.

Proposição 7.19.

*pl:syn:sem:
prop:semanticalfacts*

1. φ is a tautology if and only if $\emptyset \models \varphi$;
2. If $\Gamma \models \varphi$ and $\Gamma \models \varphi \rightarrow \psi$ then $\Gamma \models \psi$;
3. If Γ is satisfiable then every finite subset of Γ is also satisfiable;
4. *Monotonicity:* if $\Gamma \subseteq \Delta$ and $\Gamma \models \varphi$ then also $\Delta \models \varphi$;
5. *Transitivity:* if $\Gamma \models \varphi$ and $\Delta \cup \{\varphi\} \models \psi$ then $\Gamma \cup \Delta \models \psi$.

*pl:syn:sem:
def:monotonicity
pl:syn:sem:
def:Cut*

Prova. Exercise. □

Problema 7.8. Prove [Proposição 7.19](#)

*pl:syn:sem:
prop:entails-unsat*

Proposição 7.20. $\Gamma \models \varphi$ if and only if $\Gamma \cup \{\neg\varphi\}$ is unsatisfiable.

Prova. Exercise. □

Problema 7.9. Prove [Proposição 7.20](#)

*pl:syn:sem:
thm:sem-deduction*

Teorema 7.21 (Semantic Deduction Theorem). $\Gamma \models \varphi \rightarrow \psi$ if and only if $\Gamma \cup \{\varphi\} \models \psi$.

Prova. Exercise. □

Problema 7.10. Prove [Teorema 7.21](#)

Capítulo 8

Sistemas de Derivação

Este capítulo reúne material geral sobre sistemas de [derivação](#). Um livro-texto que usa um sistema específico pode inserir a seção de introdução mais a seção de panorama relevante no início do capítulo que introduz esse sistema.

[content/propositional-logic/../../first-order-logic/proof-systems/introduction.tex](#)

8.1 Introdução

*pl:prf:int:
sec*

As lógicas comumente têm tanto uma semântica quanto um sistema de [derivação](#). A semântica trata de conceitos como verdade, satisfatibilidade, validade e consequência. O propósito dos sistemas de [derivação](#) é fornecer um método puramente sintático de estabelecer consequência e validade. Eles são puramente sintáticos no sentido de que uma [derivação](#) em tal sistema é um objeto sintático finito, geralmente uma sequência (ou outra disposição finita) de [sentenças](#) ou [fórmulas](#). Bons sistemas de [derivação](#) têm a propriedade de

que qualquer sequência ou disposição dada de *sentenças* ou *fórmulas* pode ser verificada mecanicamente como “correta”.

Os sistemas de *derivação* mais simples (e historicamente os primeiros) para a lógica de primeira ordem foram *axiomáticos*. Uma sequência de *fórmulas* conta como uma *derivação* em tal sistema se cada *fórmula* individual nela é ou está entre um conjunto fixo de “axiomas” ou segue de *fórmulas* anteriores na sequência por uma entre um número fixo de “regras de inferência”—e pode ser verificado mecanicamente se uma *fórmula* é um axioma e se segue corretamente de outras *fórmulas* por uma das regras de inferência. Sistemas de *derivação* axiomáticos são fáceis de descrever—e também fáceis de tratar meta-teoricamente—mas as *derivações* neles são difíceis de ler e entender, e também difíceis de produzir.

Outros sistemas de *derivação* foram desenvolvidos com o objetivo de facilitar a construção de *derivações* ou facilitar a compreensão de *derivações* uma vez completas. Exemplos são a dedução natural, árvores de verdade, também conhecidas como provas por tableaux, e o cálculo de seqüentes. Alguns sistemas de *derivação* são projetados especialmente com a mecanização em mente; por exemplo, o método da resolução é fácil de implementar em software (mas suas *derivações* são essencialmente impossíveis de entender). A maioria desses outros sistemas de *derivação* representa *derivações* como árvores de *fórmulas* em vez de seqüências. Isso facilita ver quais partes de uma *derivação* dependem de quais outras partes.

Assim, para uma dada lógica, como a lógica de primeira ordem, os diferentes sistemas de *derivação* darão diferentes explicações do que significa uma *sentença* ser um *teorema* e o que significa uma *sentença* ser *derivável* a partir de outras. Seja como for feito isso (via *derivações* axiomáticas, deduções naturais, *derivações* de seqüentes, árvores de verdade, refutações por resolução), queremos que essas relações correspondam às noções semânticas de validade e consequência. Escrevamos $\vdash \varphi$ para “ φ é um teorema” e “ $\Gamma \vdash \varphi$ ” para “ φ é *derivável* de Γ .” Seja como for $\vdash$ definido, queremos que corresponda a $\models$, isto é:

1. $\vdash \varphi$ se, e somente se, $\models \varphi$
2. $\Gamma \vdash \varphi$ se, e somente se, $\Gamma \models \varphi$

A direção “somente se” acima é chamada de *correção*. Um sistema de *derivação* é correto se a *derivabilidade* garante consequência (ou validade). Todo sistema de *derivação* decente tem que ser correto; sistemas de *derivação* incorretos não são úteis de modo algum. Afinal, o propósito inteiro de uma *derivação* é fornecer uma garantia sintática de validade ou consequência. Provaremos a correção para os sistemas de *derivação* que apresentamos.

A direção conversas “se” também é importante: é chamada de *completude*. Um sistema de *derivação* completo é forte o suficiente para mostrar que φ é um teorema sempre que φ é válida, e que $\Gamma \vdash \varphi$ sempre que $\Gamma \models \varphi$. A completude é mais difícil de estabelecer, e algumas lógicas não têm sistemas de *derivação* completos. A lógica de primeira ordem tem. Kurt Gödel foi o primeiro a provar

8.2. O CÁLCULO DE SEQUENTES

a completude para um sistema de **derivação** da lógica de primeira ordem em sua dissertação de 1929.

Outro conceito ligado aos sistemas de **derivação** é o de *consistência*. Um conjunto de **sentenças** é chamado inconsistente se qualquer coisa pode ser **derivada** dele, e consistente caso contrário. A inconsistência é a contrapartida sintática da insatisfatibilidade: como conjuntos insatisfatíveis, conjuntos inconsistentes de **sentenças** não fazem boas teorias; são defeituosos de maneira fundamental. Conjuntos consistentes de **sentenças** podem não ser verdadeiros ou úteis, mas pelo menos passam esse limiar mínimo de utilidade lógica. Para diferentes sistemas de **derivação**, a definição específica de consistência de conjuntos de **sentenças** pode diferir, mas como $\vdash$, queremos que a consistência coincida com sua contrapartida semântica, a satisfatibilidade. Queremos que seja sempre o caso que Γ é consistente se, e somente se, é satisfatível. Aqui, a direção “somente se” corresponde à completude (consistência garante satisfatibilidade), e a direção “se” corresponde à correção (satisfatibilidade garante consistência). De fato, para a lógica clássica de primeira ordem, as duas versões de correção e completude são equivalentes.

`content/propositional-logic/./first-order-logic/proof-systems/sequent-calculus.tex`

8.2 O Cálculo de Sequentes

pl:prf:seq:
sec

Enquanto muitos sistemas de **derivação** operam com disposições de **sentenças**, o cálculo de sequentes opera com *sequentes*. Um sequente é uma expressão da forma

$$\varphi_1, \dots, \varphi_m \Rightarrow \psi_1, \dots, \psi_n,$$

isto é, um par de seqüências de **sentenças**, separadas pelo símbolo de sequente $\Rightarrow$. Qualquer uma das seqüências pode ser vazia. Uma **derivação** no cálculo de sequentes é uma árvore de sequentes, em que os sequentes mais altos são de uma forma especial (são chamados de “sequentes iniciais” ou “axiomas”) e todo outro sequente segue dos sequentes imediatamente acima dele por uma das regras de inferência. As regras de inferência ou manipulam as **sentenças** nos sequentes (adicionando, removendo ou rearranjando-as à esquerda ou à direita), ou introduzem uma **fórmula** complexa na conclusão da regra. Por exemplo, a regra $\wedge L$ permite a inferência de $\varphi, \Gamma \Rightarrow \Delta$ para $\varphi \wedge \psi, \Gamma \Rightarrow \Delta$, e a $\rightarrow R$ permite a inferência de $\varphi, \Gamma \Rightarrow \Delta, \psi$ para $\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi$, para quaisquer Γ, Δ, φ e ψ . (Em particular, Γ e Δ podem ser vazios.)

A relação $\vdash$ baseada no cálculo de sequentes é definida da seguinte forma: $\Gamma \vdash \varphi$ se, e somente se, há alguma seqüência Γ_0 tal que toda φ em Γ_0 está em Γ e há uma **derivação** com o sequente $\Gamma_0 \Rightarrow \varphi$ em sua raiz. φ é um teorema no cálculo de sequentes se o sequente $\Rightarrow \varphi$ tem uma **derivação**. Por exemplo, aqui está uma **derivação** que mostra que $\vdash (\varphi \wedge \psi) \rightarrow \varphi$:

$$\frac{\frac{\varphi \Rightarrow \varphi}{\varphi \wedge \psi \Rightarrow \varphi} \wedge L}{\Rightarrow (\varphi \wedge \psi) \rightarrow \varphi} \rightarrow R$$

Um conjunto Γ é inconsistente no cálculo de seqüentes se há uma derivação de $\Gamma_0 \Rightarrow$ (em que toda $\varphi \in \Gamma_0$ está em Γ e o lado direito do seqüente é vazio). Usando a regra WR, qualquer sentença pode ser derivada a partir de um conjunto inconsistente.

O cálculo de seqüentes foi inventado na década de 1930 por Gerhard Gentzen. Por causa de seu desenho sistemático e simétrico, é um formalismo muito útil para desenvolver uma teoria de derivações. É relativamente fácil encontrar derivações no cálculo de seqüentes, mas essas derivações são frequentemente difíceis de ler e sua conexão com provas às vezes não é fácil de ver. Provou ser uma abordagem muito elegante a sistemas de derivação, no entanto, e muitas lógicas têm sistemas de cálculo de seqüentes.

`content/propositional-logic/..first-order-logic/proof-systems/natural-deduction.te`

8.3 Dedução Natural

A dedução natural é um sistema de derivação destinado a espelhar o raciocínio real (especialmente o tipo de raciocínio regimentado empregado por matemáticos). O raciocínio real procede por vários padrões “naturais”. Por exemplo, a prova por casos nos permite estabelecer uma conclusão com base em uma premissa disjuntiva, estabelecendo que a conclusão segue de qualquer um dos disjuntos. A prova indireta nos permite estabelecer uma conclusão mostrando que sua negação leva a uma contradição. A prova condicional estabelece uma afirmação condicional “se ... então ...” mostrando que o conseqüente segue do antecedente. A dedução natural é uma formalização de algumas dessas inferências naturais. Cada um dos conectivos lógicos e quantificadores vem com duas regras, uma de introdução e uma de eliminação, e cada uma corresponde a um desses padrões de inferência natural. Por exemplo, $\rightarrow$ Intro corresponde à prova condicional, e $\vee$ Elim à prova por casos. Uma regra particularmente simples é $\wedge$ Elim, que permite a inferência de $\varphi \wedge \psi$ para φ (ou ψ).

pl:prf:ntd:
sec

Uma característica que distingue a dedução natural de outros sistemas de derivação é o uso de hipóteses. Uma derivação em dedução natural é uma árvore de fórmulas. Uma única fórmula fica na raiz da árvore de fórmulas, e as “folhas” da árvore são fórmulas das quais a conclusão é derivada. Na dedução natural, algumas fórmulas folha desempenham um papel dentro da derivação, mas são “consumidas” quando a derivação atinge a conclusão. Isso corresponde à prática, no raciocínio real, de introduzir hipóteses que só permanecem em vigor por um curto período. Por exemplo, em uma prova por casos, assumimos a verdade de cada um dos disjuntos; na prova condicional, assumimos a verdade do antecedente; na prova indireta, assumimos a verdade da negação da conclusão. Essa maneira de introduzir hipóteses hipotéticas e depois descartá-las a serviço de estabelecer um passo intermediário é uma marca da dedução natural. As fórmulas nas folhas de uma derivação em dedução natural são chamadas de hipóteses, e algumas das regras de inferência podem descarregá-las. Por exemplo, se temos uma derivação de ψ a partir de algumas hipóteses que

8.4. TABLEAUX

incluem φ , então a regra $\rightarrow$ Intro nos permite inferir $\varphi \rightarrow \psi$ e descartar qualquer hipótese da forma φ . (Para acompanhar quais hipóteses são descartadas em quais inferências, rotulamos a inferência e as hipóteses que ela descarta com um número.) As hipóteses que permanecem **não descarregadas** no final da **derivação** são, juntas, suficientes para a verdade da conclusão, de modo que uma **derivação** estabelece que suas hipóteses **não descarregadas** implicam sua conclusão.

A relação $\Gamma \vdash \varphi$ baseada na dedução natural vale se, e somente se, há uma **derivação** em que φ é a última **sentença** na árvore, e toda folha que está **não descarregada** está em Γ . φ é um teorema na dedução natural se, e somente se, há uma **derivação** em que φ é a última sentença e todas as hipóteses são **descarregadas**. Por exemplo, aqui está uma **derivação** que mostra que $\vdash (\varphi \wedge \psi) \rightarrow \varphi$:

$$1 \frac{\frac{[\varphi \wedge \psi]^1}{\varphi} \wedge \text{Elim}}{(\varphi \wedge \psi) \rightarrow \varphi} \rightarrow \text{Intro}$$

O rótulo 1 indica que a hipótese $\varphi \wedge \psi$ é **descarregada** na inferência $\rightarrow$ Intro.

Um conjunto Γ é inconsistente se, e somente se, $\Gamma \vdash \perp$ na dedução natural. A regra $\perp_I$ faz com que, a partir de um conjunto inconsistente, qualquer **sentença** possa ser **derivada**.

Sistemas de dedução natural foram desenvolvidos por Gerhard Gentzen e Stanisław Jaśkowski na década de 1930, e depois por Dag Prawitz e Frederic Fitch. Porque suas inferências espelham métodos naturais de prova, é favorecida por filósofos. As versões desenvolvidas por Fitch são frequentemente usadas em livros introdutórios de lógica. Na filosofia da lógica, as regras da dedução natural às vezes foram tomadas como dando os significados dos operadores lógicos (“semântica da Teoria da Prova”).

[content/propositional-logic/./first-order-logic/proof-systems/tableaux.tex](#)

8.4 Tableaux

pl:prf:tab:
sec Enquanto muitos sistemas de **derivação** operam com disposições de **sentenças**, os **tableaux** operam com **fórmulas assinadas**. A **fórmula assinada** é um par consistindo em um sinal de valor de verdade ($\mathbb{T}$ ou $\mathbb{F}$) e a **sentença**

$$\mathbb{T}\varphi \text{ ou } \mathbb{F}\varphi.$$

A **tableau** consiste em **fórmulas assinadas** dispostas em uma árvore ramificada para baixo. Começa com várias **hipóteses** e continua com **fórmulas assinadas** que resultam de uma das **fórmulas assinadas** acima dela pela aplicação de uma das regras de inferência. Cada regra nos permite adicionar uma ou mais **fórmulas assinadas** ao final de um ramo, ou duas **fórmulas assinadas** lado a lado—neste caso um ramo se divide em dois, com as duas **fórmulas assinadas** adicionadas formando os finais dos dois ramos.

Uma regra aplicada a uma **fórmula assinada** complexa resulta na adição de **fórmulas assinadas** que são sub-fórmulas imediatas. Elas vêm em pares, uma regra para cada um dos dois sinais. Por exemplo, a regra $\wedge\mathbb{T}$ se aplica a $\mathbb{T}\varphi \wedge \psi$, e permite a adição tanto das duas **fórmulas assinadas** $\mathbb{T}\varphi$ quanto de $\mathbb{T}\psi$ ao final de qualquer ramo que contém $\mathbb{T}\varphi \wedge \psi$, e a regra $\varphi \wedge \psi\mathbb{F}$ permite que um ramo seja dividido adicionando $\mathbb{F}\varphi$ e $\mathbb{F}\psi$ lado a lado. Um **tableau** é fechado se cada um de seus ramos contém um par correspondente de **fórmulas assinadas** $\mathbb{T}\varphi$ e $\mathbb{F}\varphi$.

A relação $\vdash$ baseada em **tableaux** é definida da seguinte forma: $\Gamma \vdash \varphi$ se, e somente se, há algum conjunto finito $\Gamma_0 = \{\psi_1, \dots, \psi_n\} \subseteq \Gamma$ tal que há um **tableau** fechado para as hipóteses

$$\{\mathbb{F}\varphi, \mathbb{T}\psi_1, \dots, \mathbb{T}\psi_n\}$$

Por exemplo, aqui está um **tableau** fechado que mostra que $\vdash (\varphi \wedge \psi) \rightarrow \varphi$:

1.	$\mathbb{F}(\varphi \wedge \psi) \rightarrow \varphi$	Assumption
2.	$\mathbb{T}\varphi \wedge \psi$	$\rightarrow\mathbb{F}1$
3.	$\mathbb{F}\varphi$	$\rightarrow\mathbb{F}1$
4.	$\mathbb{T}\varphi$	$\rightarrow\mathbb{T}2$
5.	$\mathbb{T}\psi$	$\rightarrow\mathbb{T}2$
	$\otimes$	

Um conjunto Γ é inconsistente no cálculo de **tableau** se há um **tableau** fechado para hipóteses

$$\{\mathbb{T}\psi_1, \dots, \mathbb{T}\psi_n\}$$

para algum $\psi_i \in \Gamma$.

Os **Tableaux** foram inventados na década de 1950 independentemente por Evert Beth e Jaakko Hintikka, e simplificados e popularizados por Raymond Smullyan. São muito fáceis de usar, pois construir um **tableau** é um procedimento muito sistemático. Por causa da natureza sistemática dos **tableaux**, eles também se prestam à implementação por computador. No entanto, um **tableau** é frequentemente difícil de ler e sua conexão com provas às vezes não é fácil de ver. A abordagem também é bastante geral, e muitas lógicas diferentes têm sistemas de **tableau**. Os **Tableaux** também nos ajudam a encontrar **estruturas** que satisfazem **sentenças** (ou conjuntos de **sentenças**) dadas: se o conjunto é satisfatível, ele não terá um **tableau** fechado, isto é, qualquer **tableau** terá um ramo aberto. A **estrutura** satisfatória pode ser “lida” de um ramo aberto, desde que toda regra que seja possível aplicar tenha sido aplicada naquele ramo. Há também uma conexão muito estreita com o cálculo de seqüentes: essencialmente, um **tableau** fechado é uma **derivação** condensada no cálculo de seqüentes, escrita de cabeça para baixo.

8.5. DEDUÇÃO AXIOMÁTICA

8.5 Dedução Axiomática

pl:prf:axd:
sec

As **derivações** axiomáticas são os sistemas de **derivação** lógicos mais antigos e mais simples. Suas **derivações** são simplesmente sequências de **sentenças**. Uma sequência de **sentenças** conta como uma **derivação** correta se toda **sentença** φ nela satisfaz uma das seguintes condições:

1. φ é um axioma, ou
2. φ é um **membro** de um dado conjunto Γ de **sentenças**, ou
3. φ é justificada por uma regra de inferência.

Para ser um axioma, φ tem que ter a forma de um entre vários esquemas fixos de **sentença**. Há muitos conjuntos de esquemas de axiomas que fornecem um sistema de **derivação** satisfatório (correto e completo) para a lógica de primeira ordem. Alguns são organizados de acordo com os conectivos que governam, por exemplo, os esquemas

$$\varphi \rightarrow (\psi \rightarrow \varphi) \quad \psi \rightarrow (\psi \vee \chi) \quad (\psi \wedge \chi) \rightarrow \psi$$

são axiomas comuns que governam $\rightarrow$, $\vee$ e $\wedge$. Alguns sistemas de axiomas visam um número mínimo de axiomas. Dependendo dos conectivos tomados como primitivos, é até possível encontrar sistemas de axiomas que consistem em um único axioma.

Uma regra de inferência é uma afirmação condicional que dá uma condição suficiente para que uma **sentença** em uma **derivação** seja justificada. O modus ponens é uma regra muito comum desse tipo: diz que se φ e $\varphi \rightarrow \psi$ já estão justificados, então ψ está justificado. Isso significa que uma linha em uma **derivação** que contém a **sentença** ψ está justificada, desde que tanto φ quanto $\varphi \rightarrow \psi$ (para alguma **sentença** φ) apareçam na **derivação** antes de ψ .

A relação $\vdash$ baseada em **derivações** axiomáticas é definida da seguinte forma: $\Gamma \vdash \varphi$ se, e somente se, há uma **derivação** com a **sentença** φ como sua última fórmula (e Γ é tomado como o conjunto de **sentenças** nessa **derivação** que são justificadas por (2) acima). φ é um teorema se φ tem uma **derivação** em que Γ é vazio, isto é, toda **sentença** na **derivação** é justificada por (1) ou (3). Por exemplo, aqui está uma **derivação** que mostra que $\vdash \varphi \rightarrow (\psi \rightarrow (\psi \vee \varphi))$:

1. $\psi \rightarrow (\psi \vee \varphi)$
2. $(\psi \rightarrow (\psi \vee \varphi)) \rightarrow (\varphi \rightarrow (\psi \rightarrow (\psi \vee \varphi)))$
3. $\varphi \rightarrow (\psi \rightarrow (\psi \vee \varphi))$

A **sentença** na linha 1 tem a forma do axioma $\varphi \rightarrow (\varphi \vee \psi)$ (com os papéis de φ e ψ invertidos). A **sentença** na linha 2 tem a forma do axioma $\varphi \rightarrow (\psi \rightarrow \varphi)$. Assim, ambas as linhas estão justificadas. A linha 3 é justificada por modus ponens: se a abreviarmos como θ , então a linha 2 tem a forma $\chi \rightarrow \theta$, onde χ é $\psi \rightarrow (\psi \vee \varphi)$, isto é, a linha 1.

Um conjunto Γ é inconsistente se $\Gamma \vdash \perp$. Um sistema de axiomas completo também provará que $\perp \rightarrow \varphi$ para qualquer φ , e assim, se Γ é inconsistente, então $\Gamma \vdash \varphi$ para qualquer φ .

Sistemas de **derivações** axiomáticas para lógica foram dados pela primeira vez por Gottlob Frege em sua *Begriffsschrift* de 1879, que por essa razão é frequentemente considerada a primeira obra da lógica moderna. Foram aperfeiçoados em *Principia Mathematica* de Alfred North Whitehead e Bertrand Russell e por David Hilbert e seus alunos na década de 1920. São assim frequentemente chamados de “sistemas de Frege” ou “sistemas de Hilbert”. São muito versáteis no sentido de que é frequentemente fácil encontrar um sistema axiomático para uma lógica. Porque as **derivações** têm uma estrutura muito simples e apenas uma ou duas regras de inferência, também é relativamente fácil provar coisas *sobre* eles. No entanto, são muito difíceis de usar na prática, isto é, é difícil encontrar e escrever provas.

Capítulo 9

O Cálculo de Sequentes

Este capítulo apresenta o cálculo de sequentes padrão LK de Gentzen para a lógica clássica de primeira ordem. Ele poderia ter mais exemplos e exercícios. Para incluir ou excluir material relevante ao cálculo de sequentes como sistema de prova, use a tag “prfLK”.

[content/propositional-logic/./first-order-logic/sequent-calculus/rules-and-proofs.](#)

9.1 Regras e Derivações

Para o que segue, sejam $\Gamma, \Delta, \Pi, \Lambda$ representando sequências finitas de **sentenças**. pl:seq:rul:sec

Definição 9.1 (Sequente). Um *sequente* é uma expressão da forma

$$\Gamma \Rightarrow \Delta$$

9.2. REGRAS PROPOSICIONAIS

em que Γ e Δ são sequências finitas (possivelmente vazias) de **sentenças** da linguagem $\mathcal{L}$. Γ é chamado de *antecedente*, enquanto Δ é o *sucedente*.

A ideia intuitiva por trás de um sequente é: se todas as **sentenças** no antecedente valem, então ao menos uma das **sentenças** no sucedente vale. Isto é, se $\Gamma = \langle \varphi_1, \dots, \varphi_m \rangle$ e $\Delta = \langle \psi_1, \dots, \psi_n \rangle$, então $\Gamma \Rightarrow \Delta$ vale se, e somente se, explanation

$$(\varphi_1 \wedge \dots \wedge \varphi_m) \rightarrow (\psi_1 \vee \dots \vee \psi_n)$$

vale. Há dois casos especiais: quando Γ é vazio e quando Δ é vazio. Quando Γ é vazio, isto é, $m = 0$, $\Rightarrow \Delta$ vale se, e somente se, $\psi_1 \vee \dots \vee \psi_n$ vale. Quando Δ é vazio, isto é, $n = 0$, $\Gamma \Rightarrow$ vale se, e somente se, $\neg(\varphi_1 \wedge \dots \wedge \varphi_m)$ vale. Dizemos que um sequente é válido se, e somente se, a **sentença** correspondente é válida.

Se Γ é uma sequência de **sentenças**, escrevemos Γ, φ para o resultado de acrescentar φ à extremidade direita de Γ (e φ, Γ para o resultado de acrescentar φ à extremidade esquerda de Γ). Se Δ também é uma sequência de **sentenças**, então Γ, Δ é a concatenação das duas sequências.

Definição 9.2 (Sequente Inicial). Um *sequente inicial* é um sequente de uma das seguintes formas:

1. $\varphi \Rightarrow \varphi$
2. $\Rightarrow \top$
3. $\perp \Rightarrow$

para qualquer **sentença** φ na linguagem.

Derivações no cálculo de sequentes são certas árvores de sequentes, em que os sequentes no topo são sequentes iniciais e, se um sequente está abaixo de um ou dois outros sequentes, ele deve seguir corretamente por uma regra de inferência. As regras para **LK** dividem-se em dois tipos principais: regras *lógicas* e regras *estruturais*. As regras lógicas recebem o nome do **operador principal** da **sentença** que contém φ e/ou ψ no sequente inferior. Cada uma vem em duas versões, uma para inferir um sequente com a **sentença** que contém o **operador lógico** à esquerda, e uma com a **sentença** à direita.

[content/propositional-logic/./first-order-logic/sequent-calculus/propositional-rules.tex](#)

9.2 Regras Proposicionais

pl:seq:prl:
sec **Regras para $\neg$**

$\frac{\Gamma \Rightarrow \Delta, \varphi}{\neg \varphi, \Gamma \Rightarrow \Delta} \neg L$	$\frac{\varphi, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg \varphi} \neg R$

Regras para $\wedge$

$$\frac{\varphi, \Gamma \Rightarrow \Delta}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta} \wedge L \qquad \frac{\Gamma \Rightarrow \Delta, \varphi \quad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi} \wedge R$$

Regras para $\vee$

$$\frac{\varphi, \Gamma \Rightarrow \Delta \quad \psi, \Gamma \Rightarrow \Delta}{\varphi \vee \psi, \Gamma \Rightarrow \Delta} \vee L \qquad \frac{\Gamma \Rightarrow \Delta, \varphi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi} \vee R$$

$$\frac{\Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi} \vee R$$

Regras para $\rightarrow$

$$\frac{\Gamma \Rightarrow \Delta, \varphi \quad \psi, \Pi \Rightarrow \Lambda}{\varphi \rightarrow \psi, \Gamma, \Pi \Rightarrow \Delta, \Lambda} \rightarrow L \qquad \frac{\varphi, \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi} \rightarrow R$$

[content/propositional-logic/./first-order-logic/sequent-calculus/structural-rules.](#)

9.3 Regras Estruturais

Precisamos também de algumas regras que nos permitam reorganizar **sentenças** nos lados esquerdo e direito de um sequente. Como as regras lógicas exigem que as **sentenças** na premissa sobre as quais a regra atua estejam ou na extremidade esquerda ou na extremidade direita, precisamos de uma regra de “troca” que nos permita mover **sentenças** para a posição correta. Às vezes também é importante poder combinar duas **sentenças** idênticas em uma só, e acrescentar **a sentença** em qualquer um dos lados.

Enfraquecimento

$$\frac{\Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta} WL \qquad \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \varphi} WR$$

9.4. DERIVAÇÕES

Contração

$$\frac{\varphi, \varphi, \Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta} \text{CL} \qquad \frac{\Gamma \Rightarrow \Delta, \varphi, \varphi}{\Gamma \Rightarrow \Delta, \varphi} \text{CR}$$

Troca

$$\frac{\Gamma, \varphi, \psi, \Pi \Rightarrow \Delta}{\Gamma, \psi, \varphi, \Pi \Rightarrow \Delta} \text{XL} \qquad \frac{\Gamma \Rightarrow \Delta, \varphi, \psi, \Lambda}{\Gamma \Rightarrow \Delta, \psi, \varphi, \Lambda} \text{XR}$$

Uma série de inferências de enfraquecimento, contração e troca será frequentemente indicada por linhas de inferência duplas.

A regra a seguir, chamada de “corte”, não é estritamente necessária, mas torna muito mais fácil reutilizar e combinar **derivações**.

$$\frac{\Gamma \Rightarrow \Delta, \varphi \quad \varphi, \Pi \Rightarrow \Lambda}{\Gamma, \Pi \Rightarrow \Delta, \Lambda} \text{Cut}$$

`content/propositional-logic/./first-order-logic/sequent-calculus/derivations.tex`

9.4 Derivações

pl:seq;der: sec Já dissemos como é um sequente inicial e demos as regras de inferência. De- explanation **Derivações** no cálculo de sequentes são geradas indutivamente a partir disso: cada **derivação** ou é um sequente inicial por si só, ou consiste em uma ou duas **derivações** seguidas de uma inferência.

Definição 9.3 (LK-derivação). Uma **LK-derivação** de um sequente S é uma árvore finita de sequentes que satisfaz as seguintes condições:

1. Os sequentes no topo da árvore são sequentes iniciais.
2. O sequente na base da árvore é S .
3. Todo sequente na árvore exceto S é uma premissa de uma aplicação correta de uma regra de inferência cuja conclusão está diretamente abaixo daquele sequente na árvore.

Dizemos então que S é o *sequente final* da **derivação** e que S é **derivável em LK** (ou **LK-derivável**).

CAPÍTULO 9. O CÁLCULO DE SEQUENTES

Exemplo 9.4. Todo sequente inicial, por exemplo, $\chi \Rightarrow \chi$, é a **derivação**. Podemos obter uma nova **derivação** a partir disso aplicando, digamos, a regra WL,

$$\frac{\Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta} \text{ WL}$$

A regra, contudo, deve ser geral: podemos substituir o φ na regra por qualquer **sentença**, por exemplo, também por θ . Se a premissa corresponde ao nosso sequente inicial $\chi \Rightarrow \chi$, isso significa que tanto Γ quanto Δ são apenas χ , e a conclusão seria então $\theta, \chi \Rightarrow \chi$. Assim, o seguinte é a **derivação**:

$$\frac{\chi \Rightarrow \chi}{\theta, \chi \Rightarrow \chi} \text{ WL}$$

Podemos agora aplicar outra regra, digamos XL, que nos permite trocar duas **sentenças** à esquerda. Assim, o seguinte também é uma **derivação** correta:

$$\frac{\frac{\chi \Rightarrow \chi}{\theta, \chi \Rightarrow \chi} \text{ WL}}{\chi, \theta \Rightarrow \chi} \text{ XL}$$

Nesta aplicação da regra, que foi dada como

$$\frac{\Gamma, \varphi, \psi, \Pi \Rightarrow \Delta}{\Gamma, \psi, \varphi, \Pi \Rightarrow \Delta,} \text{ XL}$$

tanto Γ quanto Π eram vazios, Δ é χ , e os papéis de φ e ψ são desempenhados por θ e χ , respectivamente. De modo muito semelhante, vemos também que

$$\frac{\theta \Rightarrow \theta}{\chi, \theta \Rightarrow \theta} \text{ WL}$$

é a **derivação**. Agora podemos pegar essas duas **derivações** e combiná-las usando $\wedge R$. Aquela regra era

$$\frac{\Gamma \Rightarrow \Delta, \varphi \quad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi} \wedge R$$

No nosso caso, as premissas devem corresponder aos últimos sequentes das **derivações** que terminam nas premissas. Isso significa que Γ é χ, θ , Δ é vazio, φ é χ e ψ é θ . Assim, a conclusão, se a inferência for correta, é $\chi, \theta \Rightarrow \chi \wedge \theta$.

$$\frac{\frac{\frac{\chi \Rightarrow \chi}{\theta, \chi \Rightarrow \chi} \text{ WL}}{\chi, \theta \Rightarrow \chi} \text{ XL} \quad \frac{\theta \Rightarrow \theta}{\chi, \theta \Rightarrow \theta} \text{ WL}}{\chi, \theta \Rightarrow \chi \wedge \theta} \wedge R$$

É claro que também podemos inverter as premissas; então φ seria θ e ψ seria χ .

9.5. EXEMPLOS DE DERIVAÇÕES

$$\frac{\frac{\theta \Rightarrow \theta}{\chi, \theta \Rightarrow \theta} \text{WL} \quad \frac{\frac{\chi \Rightarrow \chi}{\theta, \chi \Rightarrow \chi} \text{WL} \quad \frac{\theta, \chi \Rightarrow \chi}{\chi, \theta \Rightarrow \chi} \text{XL}}{\chi, \theta \Rightarrow \theta \wedge \chi} \text{XR}$$

content/propositional-logic/./first-order-logic/sequent-calculus/proving-things.tex

9.5 Exemplos de Derivações

pl:seq:pro:
sec

Exemplo 9.5. Dê uma **LK-derivação** para o sequente $\varphi \wedge \psi \Rightarrow \varphi$.

Começamos escrevendo o sequente final desejado na base da **derivação**.

$$\overline{\varphi \wedge \psi \Rightarrow \varphi}$$

Em seguida, precisamos descobrir que tipo de inferência poderia ter um sequente inferior dessa forma. Poderia ser uma regra estrutural, mas é uma boa ideia começar procurando uma regra lógica. O único conectivo lógico que ocorre no sequente inferior é $\wedge$, então procuramos uma regra de $\wedge$ e, como o símbolo $\wedge$ ocorre no antecedente, estamos diante da regra $\wedge\text{L}$.

$$\overline{\varphi \wedge \psi \Rightarrow \varphi} \wedge\text{L}$$

Há duas opções para o que poderia ter sido o sequente superior da inferência $\wedge\text{L}$: poderíamos ter um sequente superior $\varphi \Rightarrow \varphi$ ou $\psi \Rightarrow \varphi$. Claramente, $\varphi \Rightarrow \varphi$ é um sequente inicial (o que é bom), enquanto $\psi \Rightarrow \varphi$ não é derivável em geral. Preenchemos o sequente superior:

$$\frac{\varphi \Rightarrow \varphi}{\varphi \wedge \psi \Rightarrow \varphi} \wedge\text{L}$$

Agora temos uma **LK-derivação** correta do sequente $\varphi \wedge \psi \Rightarrow \varphi$.

Exemplo 9.6. Dê uma **LK-derivação** para o sequente $\neg\varphi \vee \psi \Rightarrow \varphi \rightarrow \psi$.

Comece escrevendo o sequente final desejado na base da **derivação**.

$$\overline{\neg\varphi \vee \psi \Rightarrow \varphi \rightarrow \psi}$$

Para encontrar uma regra lógica que pudesse nos dar esse sequente final, olhamos para os conectivos lógicos no sequente final: $\neg$, $\vee$ e $\rightarrow$. No momento, só nos importam $\vee$ e $\rightarrow$, porque são **operadores principais** de **sentenças** no sequente final, enquanto $\neg$ está dentro do escopo de outro conectivo, de modo que cuidaremos dele depois. Nossas opções de regras lógicas para a inferência final são, portanto, a regra $\vee\text{L}$ e a regra $\rightarrow\text{R}$. Poderíamos escolher qualquer uma das regras, na verdade, mas vamos escolher a regra $\rightarrow\text{R}$ (quando não fosse por outro motivo, porque ela nos permite adiar a divisão em dois ramos). De acordo com a forma das inferências $\rightarrow\text{R}$ que podem produzir o sequente inferior, isto deve ser assim:

CAPÍTULO 9. O CÁLCULO DE SEQUENTES

$$\frac{\overline{\varphi, \neg\varphi \vee \psi \Rightarrow \psi}}{\neg\varphi \vee \psi \Rightarrow \varphi \rightarrow \psi} \rightarrow R$$

Se movermos $\neg\varphi \vee \psi$ para a parte externa do antecedente, podemos aplicar a regra $\vee L$. De acordo com o esquema, isto deve se dividir em dois sequentes superiores como segue:

$$\frac{\frac{\overline{\neg\varphi, \varphi \Rightarrow \psi} \quad \overline{\psi, \varphi \Rightarrow \psi}}{\neg\varphi \vee \psi, \varphi \Rightarrow \psi} \vee L \quad \frac{\overline{\varphi, \neg\varphi \vee \psi \Rightarrow \psi}}{\neg\varphi \vee \psi \Rightarrow \varphi \rightarrow \psi} \rightarrow R \quad \text{XR}$$

Lembre-se de que estamos tentando subir até sequentes iniciais; parece que estamos bem perto! O ramo direito está a apenas um enfraquecimento e uma troca de um sequente inicial, e então estará pronto:

$$\frac{\frac{\overline{\neg\varphi, \varphi \Rightarrow \psi} \quad \frac{\frac{\psi \Rightarrow \psi}{\varphi, \psi \Rightarrow \psi} \text{WL}}{\psi, \varphi \Rightarrow \psi} \text{XL}}{\neg\varphi \vee \psi, \varphi \Rightarrow \psi} \vee L \quad \frac{\overline{\varphi, \neg\varphi \vee \psi \Rightarrow \psi}}{\neg\varphi \vee \psi \Rightarrow \varphi \rightarrow \psi} \rightarrow R \quad \text{XR}$$

Agora, olhando para o ramo esquerdo, o único conectivo lógico em qualquer **sentença** é o símbolo $\neg$ nas **sentenças** do antecedente, então estamos diante de uma instância da regra $\neg L$.

$$\frac{\frac{\overline{\varphi \Rightarrow \psi, \varphi}}{\neg\varphi, \varphi \Rightarrow \psi} \neg L \quad \frac{\frac{\psi \Rightarrow \psi}{\varphi, \psi \Rightarrow \psi} \text{WL}}{\psi, \varphi \Rightarrow \psi} \text{XL}}{\neg\varphi \vee \psi, \varphi \Rightarrow \psi} \vee L \quad \frac{\overline{\varphi, \neg\varphi \vee \psi \Rightarrow \psi}}{\neg\varphi \vee \psi \Rightarrow \varphi \rightarrow \psi} \rightarrow R \quad \text{XR}$$

De modo semelhante a como finalizamos o ramo direito, estamos a apenas um enfraquecimento e uma troca de finalizar também este ramo esquerdo.

$$\frac{\frac{\frac{\varphi \Rightarrow \varphi}{\varphi \Rightarrow \varphi, \psi} \text{WR}}{\varphi \Rightarrow \psi, \varphi} \text{XR} \quad \frac{\frac{\psi \Rightarrow \psi}{\varphi, \psi \Rightarrow \psi} \text{WL}}{\psi, \varphi \Rightarrow \psi} \text{XL}}{\neg\varphi, \varphi \Rightarrow \psi} \neg L \quad \frac{\overline{\neg\varphi \vee \psi, \varphi \Rightarrow \psi}}{\neg\varphi \vee \psi \Rightarrow \varphi \rightarrow \psi} \rightarrow R \quad \text{XR}$$

Exemplo 9.7. Dê uma **LK-derivação** do sequente $\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi)$

Usando as técnicas acima, começamos escrevendo o sequente final desejado na base.

9.5. EXEMPLOS DE DERIVAÇÕES

$$\overline{\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi)}$$

Os conectivos principais disponíveis das **sentenças** no sequente final são o símbolo $\vee$ e o símbolo $\neg$. Funcionaria aplicar aqui ou a regra $\vee L$ ou a regra $\neg R$, mas começamos com a regra $\neg R$ porque ela evita, por ora, a divisão em dois ramos:

$$\frac{\overline{\varphi \wedge \psi, \neg\varphi \vee \neg\psi \Rightarrow}}{\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi)} \neg R$$

Agora temos a escolha de olhar para a regra $\wedge L$ ou a regra $\vee L$. Vejamos o que acontece quando aplicamos a regra $\wedge L$: temos a escolha de começar com o sequente $\varphi, \neg\varphi \vee \neg\psi \Rightarrow$ ou com o sequente $\psi, \neg\varphi \vee \neg\psi \Rightarrow$. Como a **derivação** é simétrica em relação a φ e ψ , vamos com o primeiro:

$$\frac{\frac{\overline{\varphi, \neg\varphi \vee \neg\psi \Rightarrow}}{\varphi \wedge \psi, \neg\varphi \vee \neg\psi \Rightarrow} \wedge L}{\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi)} \neg R$$

Continuando a preencher a **derivação**, vemos que esbarramos em um problema:

$$\frac{\frac{\frac{\overline{\varphi \Rightarrow \varphi}}{\neg\varphi, \varphi \Rightarrow} \neg L \quad \frac{\frac{\overline{\varphi \Rightarrow \psi} \quad ?}{\neg\psi, \varphi \Rightarrow} \neg L}{\neg\varphi \vee \neg\psi, \varphi \Rightarrow} \vee L}{\frac{\frac{\neg\varphi \vee \neg\psi, \varphi \Rightarrow}{\varphi, \neg\varphi \vee \neg\psi \Rightarrow} XL}{\varphi \wedge \psi, \neg\varphi \vee \neg\psi \Rightarrow} \wedge L}{\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi)} \neg R$$

O topo do ramo direito não pode ser reduzido mais, e ele não pode ser levado, por meio de inferências estruturais, a um sequente inicial, então este não é o caminho certo a tomar. Assim, claramente, foi um erro aplicar a regra $\wedge L$ acima. Voltando ao que tínhamos antes e executando a regra $\vee L$ em vez disso, obtemos

$$\frac{\frac{\overline{\neg\varphi, \varphi \wedge \psi \Rightarrow} \quad \overline{\neg\psi, \varphi \wedge \psi \Rightarrow}}{\neg\varphi \vee \neg\psi, \varphi \wedge \psi \Rightarrow} \vee L}{\frac{\frac{\neg\varphi \vee \neg\psi, \varphi \wedge \psi \Rightarrow}{\varphi \wedge \psi, \neg\varphi \vee \neg\psi \Rightarrow} XL}{\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi)} \neg R$$

Completando cada ramo como fizemos antes, obtemos

$$\frac{\frac{\frac{\overline{\varphi \Rightarrow \varphi}}{\varphi \wedge \psi \Rightarrow \varphi} \wedge L \quad \frac{\overline{\psi \Rightarrow \psi}}{\varphi \wedge \psi \Rightarrow \psi} \wedge L}{\neg\varphi, \varphi \wedge \psi \Rightarrow} \neg L \quad \frac{\frac{\overline{\psi \Rightarrow \psi}}{\varphi \wedge \psi \Rightarrow \psi} \wedge L}{\neg\psi, \varphi \wedge \psi \Rightarrow} \neg L}{\neg\varphi \vee \neg\psi, \varphi \wedge \psi \Rightarrow} \vee L}{\frac{\frac{\neg\varphi \vee \neg\psi, \varphi \wedge \psi \Rightarrow}{\varphi \wedge \psi, \neg\varphi \vee \neg\psi \Rightarrow} XL}{\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi)} \neg R$$

CAPÍTULO 9. O CÁLCULO DE SEQUENTES

(Poderíamos ter executado as regras de $\wedge$ mais abaixo do que as regras de $\neg$ nesses passos e ainda assim obtido uma **derivação** correta).

Exemplo 9.8. Até agora não usamos a regra de contração, mas às vezes ela é necessária. Eis um exemplo em que isso acontece. Suponha que queiramos provar $\Rightarrow \varphi \vee \neg\varphi$. Aplicar $\vee R$ de trás para frente nos daria uma destas duas **derivações**:

$$\frac{\frac{}{\Rightarrow \varphi}}{\Rightarrow \varphi \vee \neg\varphi} \vee R \qquad \frac{\frac{\frac{}{\varphi \Rightarrow}}{\Rightarrow \neg\varphi} \neg R}{\Rightarrow \varphi \vee \neg\varphi} \vee R$$

Nenhuma destas, é claro, termina em um sequente inicial. O truque é perceber que a regra de contração nos permite combinar duas cópias de **a sentença** em uma só—e quando estamos buscando uma prova, isto é, indo de baixo para cima, podemos manter uma cópia de $\varphi \vee \neg\varphi$ na premissa, por exemplo,

$$\frac{\frac{\frac{}{\Rightarrow \varphi \vee \neg\varphi, \varphi}}{\Rightarrow \varphi \vee \neg\varphi, \varphi \vee \neg\varphi} \vee R}{\Rightarrow \varphi \vee \neg\varphi} CR$$

Agora podemos aplicar $\vee R$ uma segunda vez, e também obter $\neg\varphi$, o que leva a uma **derivação** completa.

$$\frac{\frac{\frac{\frac{\frac{}{\varphi \Rightarrow \varphi}}{\Rightarrow \varphi, \neg\varphi} \neg R}{\Rightarrow \varphi, \varphi \vee \neg\varphi} \vee R}{\Rightarrow \varphi \vee \neg\varphi, \varphi} XR}{\Rightarrow \varphi \vee \neg\varphi, \varphi \vee \neg\varphi} \vee R}{\Rightarrow \varphi \vee \neg\varphi} CR$$

Problema 9.1. Dê **derivações** dos seguintes sequentes:

1. $\varphi \wedge (\psi \wedge \chi) \Rightarrow (\varphi \wedge \psi) \wedge \chi$.
2. $\varphi \vee (\psi \vee \chi) \Rightarrow (\varphi \vee \psi) \vee \chi$.
3. $\varphi \rightarrow (\psi \rightarrow \chi) \Rightarrow \psi \rightarrow (\varphi \rightarrow \chi)$.
4. $\varphi \Rightarrow \neg\neg\varphi$.

Problema 9.2. Dê **derivações** dos seguintes sequentes:

1. $(\varphi \vee \psi) \rightarrow \chi \Rightarrow \varphi \rightarrow \chi$.
2. $(\varphi \rightarrow \chi) \wedge (\psi \rightarrow \chi) \Rightarrow (\varphi \vee \psi) \rightarrow \chi$.
3. $\Rightarrow \neg(\varphi \wedge \neg\varphi)$.
4. $\psi \rightarrow \varphi \Rightarrow \neg\varphi \rightarrow \neg\psi$.
5. $\Rightarrow (\varphi \rightarrow \neg\varphi) \rightarrow \neg\varphi$.

9.6. NOÇÕES DA TEORIA DA PROVA

6. $\Rightarrow \neg(\varphi \rightarrow \psi) \rightarrow \neg\psi$.
7. $\varphi \rightarrow \chi \Rightarrow \neg(\varphi \wedge \neg\chi)$.
8. $\varphi \wedge \neg\chi \Rightarrow \neg(\varphi \rightarrow \chi)$.
9. $\varphi \vee \psi, \neg\psi \Rightarrow \varphi$.
10. $\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi)$.
11. $\Rightarrow (\neg\varphi \wedge \neg\psi) \rightarrow \neg(\varphi \vee \psi)$.
12. $\Rightarrow \neg(\varphi \vee \psi) \rightarrow (\neg\varphi \wedge \neg\psi)$.

Problema 9.3. Dê *derivações* dos seguintes sequentes:

1. $\neg(\varphi \rightarrow \psi) \Rightarrow \varphi$.
2. $\neg(\varphi \wedge \psi) \Rightarrow \neg\varphi \vee \neg\psi$.
3. $\varphi \rightarrow \psi \Rightarrow \neg\varphi \vee \psi$.
4. $\Rightarrow \neg\neg\varphi \rightarrow \varphi$.
5. $\varphi \rightarrow \psi, \neg\varphi \rightarrow \psi \Rightarrow \psi$.
6. $(\varphi \wedge \psi) \rightarrow \chi \Rightarrow (\varphi \rightarrow \chi) \vee (\psi \rightarrow \chi)$.
7. $(\varphi \rightarrow \psi) \rightarrow \varphi \Rightarrow \varphi$.
8. $\Rightarrow (\varphi \rightarrow \psi) \vee (\psi \rightarrow \chi)$.

(Todos esses requerem a regra CR.)

Esta seção reúne as definições da relação de demonstrabilidade e de consistência para a dedução natural.

`content/propositional-logic/./first-order-logic/sequent-calculus/proof-theoretic-notions.tex`

9.6 Noções da Teoria da Prova

Assim como definimos uma série de noções semânticas importantes (validade, pl:seq:ptn: explanation consequência, satisfatibilidade), definimos agora as *noções da teoria da prova* correspondentes. Elas não são definidas por apelo à satisfação de *sentenças* em *estruturas*, mas por apelo à *derivabilidade* ou à *não derivabilidade* de certos sequentes. Foi uma descoberta importante que essas noções coincidem. Que elas coincidem é o conteúdo dos teoremas da *correção* e da *completude*.

CAPÍTULO 9. O CÁLCULO DE SEQUENTES

Definição 9.9 (Teoremas). A sentença φ é um *teorema* se há a derivação em **LK** do sequente $\Rightarrow \varphi$. Escrevemos $\vdash \varphi$ se φ é um teorema e $\nvdash \varphi$ se não é.

Definição 9.10 (Derivabilidade). A sentença φ é *derivável* de um conjunto de sentenças Γ , $\Gamma \vdash \varphi$, se, e somente se, há um subconjunto finito $\Gamma_0 \subseteq \Gamma$ e uma sequência Γ'_0 das sentenças em Γ_0 tal que **LK** deriva $\Gamma'_0 \Rightarrow \varphi$. Se φ não é derivável de Γ , escrevemos $\Gamma \nvdash \varphi$.

Por causa das regras de contração, enfraquecimento e troca, a ordem e o número de sentenças em Γ'_0 não importam: se um sequente $\Gamma'_0 \Rightarrow \varphi$ é derivável, então também o é $\Gamma''_0 \Rightarrow \varphi$ para qualquer Γ''_0 que contenha as mesmas sentenças que Γ'_0 . Por exemplo, se $\Gamma_0 = \{\psi, \chi\}$, então tanto $\Gamma'_0 = \langle \psi, \psi, \chi \rangle$ quanto $\Gamma''_0 = \langle \chi, \chi, \psi \rangle$ são sequências que contêm apenas as sentenças em Γ_0 . Se um sequente contendo uma delas é derivável, o outro também é, por exemplo:

$$\begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \hline \psi, \psi, \chi \Rightarrow \varphi \\ \hline \psi, \chi \Rightarrow \varphi \quad \text{CL} \\ \hline \chi, \psi \Rightarrow \varphi \quad \text{XL} \\ \hline \chi, \chi, \psi \Rightarrow \varphi \quad \text{WL} \end{array}$$

De agora em diante, diremos que, se Γ_0 é um conjunto finito de sentenças, então $\Gamma_0 \Rightarrow \varphi$ é qualquer sequente em que o antecedente é uma sequência de sentenças em Γ_0 , e incluiremos tacitamente contrações, trocas e enfraquecimentos se necessário.

Definição 9.11 (Consistência). Um conjunto de sentenças Γ é *inconsistente* se, e somente se, há um subconjunto finito $\Gamma_0 \subseteq \Gamma$ tal que **LK** deriva $\Gamma_0 \Rightarrow$. Se Γ não é inconsistente, isto é, se para todo $\Gamma_0 \subseteq \Gamma$ finito, **LK** não deriva $\Gamma_0 \Rightarrow$, dizemos que é *consistente*.

Proposição 9.12 (Reflexividade). Se $\varphi \in \Gamma$, então $\Gamma \vdash \varphi$.

*pl:seq:ptn:
prop:reflexivity*

Prova. O sequente inicial $\varphi \Rightarrow \varphi$ é derivável, e $\{\varphi\} \subseteq \Gamma$. □

Proposição 9.13 (Monotonicidade). Se $\Gamma \subseteq \Delta$ e $\Gamma \vdash \varphi$, então $\Delta \vdash \varphi$.

*pl:seq:ptn:
prop:monotonicity*

Prova. Suponha $\Gamma \vdash \varphi$, isto é, há um $\Gamma_0 \subseteq \Gamma$ finito tal que $\Gamma_0 \Rightarrow \varphi$ é derivável. Como $\Gamma \subseteq \Delta$, então Γ_0 também é um subconjunto finito de Δ . A derivação de $\Gamma_0 \Rightarrow \varphi$ mostra, assim, também que $\Delta \vdash \varphi$. □

Proposição 9.14 (Transitividade). Se $\Gamma \vdash \varphi$ e $\{\varphi\} \cup \Delta \vdash \psi$, então $\Gamma \cup \Delta \vdash \psi$.

*pl:seq:ptn:
prop:transitivity*

9.7. DERIVABILIDADE E CONSISTÊNCIA

Prova. Se $\Gamma \vdash \varphi$, há um $\Gamma_0 \subseteq \Gamma$ finito e a derivação π_0 de $\Gamma_0 \Rightarrow \varphi$. Se $\{\varphi\} \cup \Delta \vdash \psi$, então, para algum subconjunto finito $\Delta_0 \subseteq \Delta$, há a derivação π_1 de $\varphi, \Delta_0 \Rightarrow \psi$. Considere a seguinte derivação:

$$\frac{\begin{array}{c} \vdots \\ \pi_0 \\ \vdots \\ \Gamma_0 \Rightarrow \varphi \end{array} \quad \begin{array}{c} \vdots \\ \pi_1 \\ \vdots \\ \varphi, \Delta_0 \Rightarrow \psi \end{array}}{\Gamma_0, \Delta_0 \Rightarrow \psi} \text{Cut}$$

Como $\Gamma_0 \cup \Delta_0 \subseteq \Gamma \cup \Delta$, isso mostra $\Gamma \cup \Delta \vdash \psi$. $\square$

Note que isso significa, em particular, que se $\Gamma \vdash \varphi$ e $\varphi \vdash \psi$, então $\Gamma \vdash \psi$. Segue-se também que se $\varphi_1, \dots, \varphi_n \vdash \psi$ e $\Gamma \vdash \varphi_i$ para cada i , então $\Gamma \vdash \psi$.

Proposição 9.15. *Γ é inconsistente se, e somente se, $\Gamma \vdash \varphi$ para toda sentença φ .*

Prova. Exercício. $\square$

Problema 9.4. Prove [Proposição 19.16](#)

Proposição 9.16 (Compacidade).

1. Se $\Gamma \vdash \varphi$, então há um subconjunto finito $\Gamma_0 \subseteq \Gamma$ tal que $\Gamma_0 \vdash \varphi$.
2. Se todo subconjunto finito de Γ é consistente, então Γ é consistente.

Prova. 1. Se $\Gamma \vdash \varphi$, então há um subconjunto finito $\Gamma_0 \subseteq \Gamma$ tal que o sequente $\Gamma_0 \Rightarrow \varphi$ tem a derivação. Consequentemente, $\Gamma_0 \vdash \varphi$.

2. Se Γ é inconsistente, há um subconjunto finito $\Gamma_0 \subseteq \Gamma$ tal que **LK** deriva $\Gamma_0 \Rightarrow$. Mas então Γ_0 é um subconjunto finito de Γ que é inconsistente. $\square$

[content/propositional-logic/./first-order-logic/sequent-calculus/provability-consistency.tex](#)

9.7 Derivabilidade e Consistência

Estabeleceremos agora uma série de propriedades da relação de derivabilidade. Elas são interessantes por si mesmas, mas cada uma terá um papel na prova do teorema da completude.

Proposição 9.17. *Se $\Gamma \vdash \varphi$ e $\Gamma \cup \{\varphi\}$ é inconsistente, então Γ é inconsistente.*

Prova. Há Γ_0 e $\Gamma_1 \subseteq \Gamma$ finitos tais que **LK** deriva $\Gamma_0 \Rightarrow \varphi$ e $\varphi, \Gamma_1 \Rightarrow$. Seja π_0 a **LK-derivação** de $\Gamma_0 \Rightarrow \varphi$ e π_1 a **LK-derivação** de $\Gamma_1, \varphi \Rightarrow$. Podemos então derivar

CAPÍTULO 9. O CÁLCULO DE SEQUENTES

$$\frac{\frac{\vdots \pi_0}{\Gamma_0 \Rightarrow \varphi} \quad \frac{\vdots \pi_1}{\varphi, \Gamma_1 \Rightarrow}}{\Gamma_0, \Gamma_1 \Rightarrow} \text{Cut}$$

Como $\Gamma_0 \subseteq \Gamma$ e $\Gamma_1 \subseteq \Gamma$, $\Gamma_0 \cup \Gamma_1 \subseteq \Gamma$; portanto, Γ é inconsistente. $\square$

Proposição 9.18. $\Gamma \vdash \varphi$ se, e somente se, $\Gamma \cup \{\neg\varphi\}$ é inconsistente.

*pl:seq:prv:
prop:prov-incons*

Prova. Primeiro, suponha $\Gamma \vdash \varphi$, isto é, há a derivação π_0 de $\Gamma \Rightarrow \varphi$. Acrescentando uma regra $\neg$ L, obtemos a derivação de $\neg\varphi, \Gamma \Rightarrow$, isto é, $\Gamma \cup \{\neg\varphi\}$ é inconsistente.

Se $\Gamma \cup \{\neg\varphi\}$ é inconsistente, há a derivação π_1 de $\neg\varphi, \Gamma \Rightarrow$. O seguinte é a derivação de $\Gamma \Rightarrow \varphi$:

$$\frac{\frac{\varphi \Rightarrow \varphi}{\Rightarrow \varphi, \neg\varphi} \neg R \quad \frac{\vdots \pi_1}{\neg\varphi, \Gamma \Rightarrow}}{\Gamma \Rightarrow \varphi} \text{Cut}$$

$\square$

Problema 9.5. Prove que $\Gamma \vdash \neg\varphi$ se, e somente se, $\Gamma \cup \{\varphi\}$ é inconsistente.

Proposição 9.19. Se $\Gamma \vdash \varphi$ e $\neg\varphi \in \Gamma$, então Γ é inconsistente.

*pl:seq:prv:
prop:explicit-inc*

Prova. Suponha $\Gamma \vdash \varphi$ e $\neg\varphi \in \Gamma$. Então há a derivação π de um sequente $\Gamma_0 \Rightarrow \varphi$. O sequente $\neg\varphi, \Gamma_0 \Rightarrow$ também é derivável:

$$\frac{\frac{\vdots \pi}{\Gamma_0 \Rightarrow \varphi} \quad \frac{\frac{\varphi \Rightarrow \varphi}{\neg\varphi, \varphi \Rightarrow} \neg L}{\frac{\varphi, \neg\varphi \Rightarrow}{\neg\varphi, \neg\varphi \Rightarrow} \text{XL}}{\Gamma_0, \neg\varphi \Rightarrow} \text{Cut}$$

Como $\neg\varphi \in \Gamma$ e $\Gamma_0 \subseteq \Gamma$, isso mostra que Γ é inconsistente. $\square$

Proposição 9.20. Se $\Gamma \cup \{\varphi\}$ e $\Gamma \cup \{\neg\varphi\}$ são ambos inconsistentes, então Γ é inconsistente.

*pl:seq:prv:
prop:provability-exhaustive*

Prova. Há conjuntos finitos $\Gamma_0 \subseteq \Gamma$ e $\Gamma_1 \subseteq \Gamma$ e **LK-derivações** π_0 e π_1 de $\varphi, \Gamma_0 \Rightarrow$ e $\neg\varphi, \Gamma_1 \Rightarrow$, respectivamente. Podemos então derivar

$$\frac{\frac{\frac{\vdots \pi_0}{\varphi, \Gamma_0 \Rightarrow}}{\Gamma_0 \Rightarrow \neg\varphi} \neg R \quad \frac{\vdots \pi_1}{\neg\varphi, \Gamma_1 \Rightarrow}}{\Gamma_0, \Gamma_1 \Rightarrow} \text{Cut}$$

9.8. DERIVABILIDADE E OS CONECTIVOS PROPOSICIONAIS

Como $\Gamma_0 \subseteq \Gamma$ e $\Gamma_1 \subseteq \Gamma$, $\Gamma_0 \cup \Gamma_1 \subseteq \Gamma$. Portanto, Γ é inconsistente. $\square$

[content/propositional-logic/./first-order-logic/sequent-calculus/provability-propositional.t](#)

9.8 Derivabilidade e os Conectivos Proposicionais

pl:seq:ppr: Estabelecemos que a relação de **derivabilidade** $\vdash$ do cálculo de seqüentes é **forte** o suficiente para estabelecer alguns fatos básicos envolvendo os conectivos **sec** proposicionais, tais como $\varphi \wedge \psi \vdash \varphi$ e $\varphi, \varphi \rightarrow \psi \vdash \psi$ (modus ponens). Esses fatos são necessários para a prova do teorema da completude. **explanation**

Proposição 9.21.

1. Tanto $\varphi \wedge \psi \vdash \varphi$ quanto $\varphi \wedge \psi \vdash \psi$.
2. $\varphi, \psi \vdash \varphi \wedge \psi$.

Prova. 1. Ambos os seqüentes $\varphi \wedge \psi \Rightarrow \varphi$ e $\varphi \wedge \psi \Rightarrow \psi$ são **deriváveis**:

$$\frac{\varphi \Rightarrow \varphi}{\varphi \wedge \psi \Rightarrow \varphi} \wedge L \quad \frac{\psi \Rightarrow \psi}{\varphi \wedge \psi \Rightarrow \psi} \wedge L$$

2. Eis a **derivação** do seqüente $\varphi, \psi \Rightarrow \varphi \wedge \psi$:

$$\frac{\varphi \Rightarrow \varphi \quad \psi \Rightarrow \psi}{\varphi, \psi \Rightarrow \varphi \wedge \psi} \wedge R$$

$\square$

Proposição 9.22.

1. $\varphi \vee \psi, \neg\varphi, \neg\psi$ é inconsistente.
2. Tanto $\varphi \vdash \varphi \vee \psi$ quanto $\psi \vdash \varphi \vee \psi$.

Prova. 1. Damos a **derivação** do seqüente $\varphi \vee \psi, \neg\varphi, \neg\psi \Rightarrow$:

$$\frac{\frac{\frac{\varphi \Rightarrow \varphi}{\neg\varphi, \varphi \Rightarrow} \neg L}{\varphi, \neg\varphi, \neg\psi \Rightarrow} \quad \frac{\frac{\psi \Rightarrow \psi}{\neg\psi, \psi \Rightarrow} \neg L}{\psi, \neg\varphi, \neg\psi \Rightarrow} \vee L$$

(Lembre-se de que linhas de inferência duplas indicam várias inferências de enfraquecimento, contração e troca.)

2. Ambos os seqüentes $\varphi \Rightarrow \varphi \vee \psi$ e $\psi \Rightarrow \varphi \vee \psi$ têm **derivações**:

$$\frac{\varphi \Rightarrow \varphi}{\varphi \Rightarrow \varphi \vee \psi} \vee R \quad \frac{\psi \Rightarrow \psi}{\psi \Rightarrow \varphi \vee \psi} \vee R$$

$\square$

CAPÍTULO 9. O CÁLCULO DE SEQUENTES

Proposição 9.23.

1. $\varphi, \varphi \rightarrow \psi \vdash \psi$.
2. Tanto $\neg\varphi \vdash \varphi \rightarrow \psi$ quanto $\psi \vdash \varphi \rightarrow \psi$.

*pl:seq:ppr:
prop:provability-lif
pl:seq:ppr:
prop:provability-lif-left
pl:seq:ppr:
prop:provability-lif-right*

Prova. 1. O sequente $\varphi \rightarrow \psi, \varphi \Rightarrow \psi$ é **derivável**:

$$\frac{\varphi \Rightarrow \varphi \quad \psi \Rightarrow \psi}{\varphi \rightarrow \psi, \varphi \Rightarrow \psi} \rightarrow L$$

2. Ambos os sequentes $\neg\varphi \Rightarrow \varphi \rightarrow \psi$ e $\psi \Rightarrow \varphi \rightarrow \psi$ são **deriváveis**:

$$\frac{\frac{\frac{\varphi \Rightarrow \varphi}{\neg\varphi, \varphi \Rightarrow} \neg L \quad \frac{\varphi, \neg\varphi \Rightarrow}{\varphi, \neg\varphi \Rightarrow \psi} XL}{\varphi, \neg\varphi \Rightarrow \psi} WR \quad \frac{\psi \Rightarrow \psi}{\varphi, \psi \Rightarrow \psi} WL \quad \frac{\varphi, \psi \Rightarrow \psi}{\psi \Rightarrow \varphi \rightarrow \psi} \rightarrow R \quad \square$$

content/propositional-logic/./first-order-logic/sequent-calculus/soundness.tex

9.9 Correção

explanation Um sistema de **derivação**, tal como o cálculo de sequentes, é *correto* se não pode derivar coisas que de fato não valem. A correção é, assim, uma espécie de propriedade de segurança garantida para sistemas de **derivação**. Dependendo de qual propriedade da teoria da prova está em questão, gostaríamos de saber, por exemplo, que

*pl:seq:sou:
sec*

1. todo φ **derivável** é uma tautologia;
2. se a **sentença** é **derivável** de algumas outras, ela também é consequência delas;
3. se um conjunto de **sentenças** é inconsistente, ele é insatisfatível.

Essas são propriedades importantes de um sistema de **derivação**. Se alguma delas não vale, o sistema de **derivação** é deficiente—ele **deriva** demais. Consequentemente, estabelecer a correção de um sistema de **derivação** é da máxima importância.

Como todas essas propriedades da teoria da prova são definidas via **derivabilidade** no cálculo de sequentes de certos sequentes, provar (1)–(3) acima requer provar algo sobre as propriedades semânticas dos sequentes **deriváveis**. Primeiro definiremos o que significa um sequente ser *válido*, e então mostraremos que todo sequente **derivável** é válido. (1)–(3) seguem-se então como corolários desse resultado.

9.9. CORREÇÃO

Definição 9.24. A **valoração** $\mathbf{v}$ *satisfaz* um sequente $\Gamma \Rightarrow \Delta$ se, e somente se, ou $\mathbf{v} \not\models \varphi$ para algum $\varphi \in \Gamma$ ou $\mathbf{v} \models \varphi$ para algum $\varphi \in \Delta$.

Um sequente é *válido* se, e somente se, toda **valoração** $\mathbf{v}$ o satisfaz.

Teorema 9.25 (Correção). Se LK *deriva* $\Theta \Rightarrow \Xi$, então $\Theta \Rightarrow \Xi$ é válido.

pl:seq:sou:
thm:sequent-soundness

Prova. Seja π a **derivação** de $\Theta \Rightarrow \Xi$. Procedemos por indução sobre o número de inferências n em π .

Se o número de inferências é 0, então π consiste apenas em um sequente inicial. Todo sequente inicial $\varphi \Rightarrow \varphi$ é obviamente válido, pois para toda $\mathbf{v}$, ou $\mathbf{v} \not\models \varphi$ ou $\mathbf{v} \models \varphi$.

Se o número de inferências é maior que 0, distinguimos casos de acordo com o tipo da inferência mais inferior. Pela hipótese de indução, podemos supor que as premissas dessa inferência são válidas, já que o número de inferências na **derivação** de qualquer premissa é menor que n .

Primeiro, consideramos as inferências possíveis com apenas uma premissa.

1. A última inferência é um enfraquecimento. Então $\Theta \Rightarrow \Xi$ é $\varphi, \Gamma \Rightarrow \Delta$ (se a última inferência é WL) ou $\Gamma \Rightarrow \Delta, \varphi$ (se é WR), e a **derivação** termina em uma de

$$\frac{\begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \Gamma \Rightarrow \Delta \end{array}}{\varphi, \Gamma \Rightarrow \Delta} \text{WL} \qquad \frac{\begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \Gamma \Rightarrow \Delta \end{array}}{\Gamma \Rightarrow \Delta, \varphi} \text{WR}$$

Pela hipótese de indução, $\Gamma \Rightarrow \Delta$ é válido, isto é, para toda **valoração** $\mathbf{v}$, ou há algum $\chi \in \Gamma$ tal que $\mathbf{v} \not\models \chi$ ou há algum $\chi \in \Delta$ tal que $\mathbf{v} \models \chi$.

Se $\mathbf{v} \not\models \chi$ para algum $\chi \in \Gamma$, então $\chi \in \Theta$ também, já que $\Theta = \varphi, \Gamma$, e assim $\mathbf{v} \not\models \chi$ para algum $\chi \in \Theta$. Analogamente, se $\mathbf{v} \models \chi$ para algum $\chi \in \Delta$, como $\chi \in \Xi$, $\mathbf{v} \models \chi$ para algum $\chi \in \Xi$. Consequentemente, $\Theta \Rightarrow \Xi$ é válido.

2. A última inferência é $\neg$ L: Então a premissa da última inferência é $\Gamma \Rightarrow \Delta, \varphi$ e a conclusão é $\neg\varphi, \Gamma \Rightarrow \Delta$, isto é, a **derivação** termina em

$$\frac{\begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \Gamma \Rightarrow \Delta, \varphi \end{array}}{\neg\varphi, \Gamma \Rightarrow \Delta} \neg\text{L}$$

e $\Theta = \neg\varphi, \Gamma$ enquanto $\Xi = \Delta$.

A hipótese de indução nos diz que $\Gamma \Rightarrow \Delta, \varphi$ é válido, isto é, para toda $\mathbf{v}$, ou (a) para algum $\chi \in \Gamma$, $\mathbf{v} \not\models \chi$, ou (b) para algum $\chi \in \Delta$, $\mathbf{v} \models \chi$, ou (c) $\mathbf{v} \models \varphi$. Queremos mostrar que $\Theta \Rightarrow \Xi$ também é válido. Seja

CAPÍTULO 9. O CÁLCULO DE SEQUENTES

v a valoração. Se (a) vale, então há $\chi \in \Gamma$ tal que $\mathbf{v} \not\models \chi$, mas $\chi \in \Theta$ também. Se (b) vale, há $\chi \in \Delta$ tal que $\mathbf{v} \models \chi$, mas $\chi \in \Xi$ também. Por fim, se $\mathbf{v} \models \varphi$, então $\mathbf{v} \not\models \neg\varphi$. Como $\neg\varphi \in \Theta$, há $\chi \in \Theta$ tal que $\mathbf{v} \not\models \chi$. Consequentemente, $\Theta \Rightarrow \Xi$ é válido.

3. A última inferência é $\neg R$: Exercício.
4. A última inferência é $\wedge L$: Há duas variantes: $\varphi \wedge \psi$ pode ser inferido à esquerda a partir de φ ou de ψ no lado esquerdo da premissa. No primeiro caso, π termina em

$$\frac{\begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \varphi, \Gamma \Rightarrow \Delta \end{array}}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta} \wedge L$$

e $\Theta = \varphi \wedge \psi, \Gamma$ enquanto $\Xi = \Delta$. Considere **a valoração v**. Como, pela hipótese de indução, $\varphi, \Gamma \Rightarrow \Delta$ é válido, (a) $\mathbf{v} \not\models \varphi$, (b) $\mathbf{v} \not\models \chi$ para algum $\chi \in \Gamma$, ou (c) $\mathbf{v} \models \chi$ para algum $\chi \in \Delta$. No caso (a), $\mathbf{v} \not\models \varphi \wedge \psi$, então há $\chi \in \Theta$ (a saber, $\varphi \wedge \psi$) tal que $\mathbf{v} \not\models \chi$. No caso (b), há $\chi \in \Gamma$ tal que $\mathbf{v} \not\models \chi$, e $\chi \in \Theta$ também. No caso (c), há $\chi \in \Delta$ tal que $\mathbf{v} \models \chi$, e $\chi \in \Xi$ também, já que $\Xi = \Delta$. Assim, em cada caso, $\mathbf{v}$ satisfaz $\varphi \wedge \psi, \Gamma \Rightarrow \Delta$. Como $\mathbf{v}$ era arbitrário, $\Gamma \Rightarrow \Delta$ é válido. O caso em que $\varphi \wedge \psi$ é inferido a partir de ψ é tratado da mesma forma, mudando φ para ψ .

5. A última inferência é $\vee R$: Há duas variantes: $\varphi \vee \psi$ pode ser inferido à direita a partir de φ ou de ψ no lado direito da premissa. No primeiro caso, π termina em

$$\frac{\begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \Gamma \Rightarrow \Delta, \varphi \end{array}}{\Gamma \Rightarrow \Delta, \varphi \vee \psi} \vee R$$

Agora $\Theta = \Gamma$ e $\Xi = \Delta, \varphi \vee \psi$. Considere **a valoração v**. Como $\Gamma \Rightarrow \Delta, \varphi$ é válido, (a) $\mathbf{v} \models \varphi$, (b) $\mathbf{v} \not\models \chi$ para algum $\chi \in \Gamma$, ou (c) $\mathbf{v} \models \chi$ para algum $\chi \in \Delta$. No caso (a), $\mathbf{v} \models \varphi \vee \psi$. No caso (b), há $\chi \in \Gamma$ tal que $\mathbf{v} \not\models \chi$. No caso (c), há $\chi \in \Delta$ tal que $\mathbf{v} \models \chi$. Assim, em cada caso, $\mathbf{v}$ satisfaz $\Gamma \Rightarrow \Delta, \varphi \vee \psi$, isto é, $\Theta \Rightarrow \Xi$. Como $\mathbf{v}$ era arbitrário, $\Theta \Rightarrow \Xi$ é válido. O caso em que $\varphi \vee \psi$ é inferido a partir de ψ é tratado da mesma forma, mudando φ para ψ .

6. A última inferência é $\rightarrow R$: Então π termina em

9.9. CORREÇÃO

$$\frac{\begin{array}{c} \vdots \\ \varphi, \Gamma \Rightarrow \Delta, \psi \end{array}}{\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi} \rightarrow R$$

Novamente, a hipótese de indução diz que a premissa é válida; queremos mostrar que a conclusão também é válida. Seja $\mathbf{v}$ arbitrário. Como $\varphi, \Gamma \Rightarrow \Delta, \psi$ é válido, ao menos um dos seguintes casos ocorre: (a) $\mathbf{v} \not\models \varphi$, (b) $\mathbf{v} \models \psi$, (c) $\mathbf{v} \not\models \chi$ para algum $\chi \in \Gamma$, ou (d) $\mathbf{v} \models \chi$ para algum $\chi \in \Delta$. Nos casos (a) e (b), $\mathbf{v} \models \varphi \rightarrow \psi$ e assim há um $\chi \in \Delta, \varphi \rightarrow \psi$ tal que $\mathbf{v} \models \chi$. No caso (c), para algum $\chi \in \Gamma$, $\mathbf{v} \not\models \chi$. No caso (d), para algum $\chi \in \Delta$, $\mathbf{v} \models \chi$. Em cada caso, $\mathbf{v}$ satisfaz $\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi$. Como $\mathbf{v}$ era arbitrário, $\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi$ é válido.

Agora consideremos as inferências possíveis com duas premissas.

1. A última inferência é um corte: então π termina em

$$\frac{\begin{array}{c} \vdots \\ \Gamma \Rightarrow \Delta, \varphi \end{array} \quad \begin{array}{c} \vdots \\ \varphi, \Pi \Rightarrow \Lambda \end{array}}{\Gamma, \Pi \Rightarrow \Delta, \Lambda} \text{Cut}$$

Seja $\mathbf{v}$ a **avaliação**. Pela hipótese de indução, as premissas são válidas, então $\mathbf{v}$ satisfaz ambas as premissas. Distinguímos dois casos: (a) $\mathbf{v} \not\models \varphi$ e (b) $\mathbf{v} \models \varphi$. No caso (a), para que $\mathbf{v}$ satisfaça a premissa esquerda, ela deve satisfazer $\Gamma \Rightarrow \Delta$. Mas então também satisfaz a conclusão. No caso (b), para que $\mathbf{v}$ satisfaça a premissa direita, ela deve satisfazer $\Pi \setminus \Lambda$. Novamente, $\mathbf{v}$ satisfaz a conclusão.

2. A última inferência é $\wedge R$. Então π termina em

$$\frac{\begin{array}{c} \vdots \\ \Gamma \Rightarrow \Delta, \varphi \end{array} \quad \begin{array}{c} \vdots \\ \Gamma \Rightarrow \Delta, \psi \end{array}}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi} \wedge R$$

Considere a **avaliação** $\mathbf{v}$. Se $\mathbf{v}$ satisfaz $\Gamma \Rightarrow \Delta$, terminamos. Então suponha que não. Como $\Gamma \Rightarrow \Delta, \varphi$ é válido pela hipótese de indução, $\mathbf{v} \models \varphi$. Analogamente, como $\Gamma \Rightarrow \Delta, \psi$ é válido, $\mathbf{v} \models \psi$. Mas então $\mathbf{v} \models \varphi \wedge \psi$.

3. A última inferência é $\vee L$: Exercício.
4. A última inferência é $\rightarrow L$. Então π termina em

$$\frac{\begin{array}{c} \vdots \\ \Gamma \Rightarrow \Delta, \varphi \end{array} \quad \begin{array}{c} \vdots \\ \psi, \Pi \Rightarrow \Lambda \end{array}}{\varphi \rightarrow \psi, \Gamma, \Pi \Rightarrow \Delta, \Lambda} \rightarrow L$$

Novamente, considere a **avaliação** $\mathbf{v}$ e suponha que $\mathbf{v}$ não satisfaça $\Gamma, \Pi \Rightarrow \Delta, \Lambda$. Temos de mostrar que $\mathbf{v} \not\models \varphi \rightarrow \psi$. Se $\mathbf{v}$ não satisfaz $\Gamma, \Pi \Rightarrow \Delta, \Lambda$, não satisfaz nem $\Gamma \Rightarrow \Delta$ nem $\Pi \Rightarrow \Lambda$. Como $\Gamma \Rightarrow \Delta, \varphi$ é válido, temos $\mathbf{v} \models \varphi$. Como $\psi, \Pi \Rightarrow \Lambda$ é válido, temos $\mathbf{v} \not\models \psi$. Mas então $\mathbf{v} \not\models \varphi \rightarrow \psi$, que é o que queríamos mostrar. $\square$

Problema 9.6. Complete a prova de **Teorema 9.25**.

Corolário 9.26. Se $\vdash \varphi$, então φ é uma tautologia.

*pl:seq:sou:
cor:weak-soundness*

Corolário 9.27. Se $\Gamma \vdash \varphi$, então $\Gamma \models \varphi$.

*pl:seq:sou:
cor:entailment-soundness*

Prova. Se $\Gamma \vdash \varphi$, então, para algum subconjunto finito $\Gamma_0 \subseteq \Gamma$, há a **derivação** de $\Gamma_0 \Rightarrow \varphi$. Por **Teorema 9.25**, toda **avaliação** $\mathbf{v}$ ou torna algum $\psi \in \Gamma_0$ falso ou torna φ verdadeiro. Logo, se $\mathbf{v} \models \Gamma$, então também $\mathbf{v} \models \varphi$. $\square$

Corolário 9.28. Se Γ é satisfatível, então é consistente.

*pl:seq:sou:
cor:consistency-soundness*

Prova. Provamos a contrapositiva. Suponha que Γ não seja consistente. Então há um $\Gamma_0 \subseteq \Gamma$ finito e a **derivação** de $\Gamma_0 \Rightarrow \perp$. Por **Teorema 9.25**, $\Gamma_0 \Rightarrow \perp$ é válido. Em outras palavras, para toda **avaliação** $\mathbf{v}$, há $\chi \in \Gamma_0$ tal que $\mathbf{v} \not\models \chi$, e como $\Gamma_0 \subseteq \Gamma$, esse χ também está em Γ . Assim, nenhuma $\mathbf{v}$ satisfaz Γ , e Γ não é satisfatível. $\square$

Capítulo 10

Dedução Natural

10.1. REGRAS E DERIVAÇÕES

Este capítulo apresenta um sistema de dedução natural no estilo de Gentzen/Prawitz.

Para incluir ou excluir material relevante à dedução natural como sistema de prova, use a tag “prfND”.

content/propositional-logic/./first-order-logic/natural-deduction/rules-and-proofs.tex

10.1 Regras e Derivações

pl:ntd:rul:
sec

Sistemas de dedução natural pretendem ser um paralelo próximo do raciocínio informal usado em provas matemáticas (daí serem, de certo modo, “naturais”). Provas em dedução natural começam com hipóteses. Em seguida, aplicam-se regras de inferência. As hipóteses são “descarregadas” pelas regras de inferência $\neg$ -Intro, $\rightarrow$ -Intro, e $\vee$ -Elim, e o rótulo da hipótese *descarregada* é colocado ao lado da inferência para maior clareza.

explanation

Definição 10.1 (Hipótese). Uma *hipótese* é qualquer *sentença* na posição mais alta de qualquer ramo.

Derivações em dedução natural são certas árvores de *sentenças*, em que as *sentenças* no topo são hipóteses e, se a *sentença* está abaixo de um, dois ou três outros sequentes, ela deve seguir corretamente por uma regra de inferência. As *sentenças* no topo da inferência são chamadas de *premissas* e a *sentença* abaixo, de *conclusão* da inferência. As regras vêm em pares, uma regra de introdução e uma de eliminação para cada *operador lógico*. Elas introduzem a *operador lógico* na conclusão ou removem a *operador lógico* de uma premissa da regra. Algumas das regras permitem que uma hipótese de certo tipo seja *descarregada*. Para indicar qual hipótese é *descarregada* por qual inferência, também atribuímos rótulos tanto à hipótese quanto à inferência. Isso é indicado escrevendo-se a hipótese como “[φ]ⁿ.”

É costume considerar regras para todos os *operadores lógicos* $\wedge$, $\vee$, $\rightarrow$, $\neg$ e $\perp$, mesmo que alguns deles sejam definidos.

content/propositional-logic/./first-order-logic/natural-deduction/propositional-rules.tex

10.2 Regras Proposicionais

pl:ntd:prl:
sec

Regras para $\wedge$

$$\begin{array}{c}
 \frac{\varphi \quad \psi}{\varphi \wedge \psi} \wedge \text{Intro}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{\varphi \wedge \psi}{\varphi} \wedge \text{Elim} \\
 \frac{\varphi \wedge \psi}{\psi} \wedge \text{Elim}
 \end{array}$$

Regras para $\vee$

$$\begin{array}{c}
 \frac{\varphi}{\varphi \vee \psi} \vee \text{Intro} \\
 \frac{\psi}{\varphi \vee \psi} \vee \text{Intro}
 \end{array}
 \qquad
 \begin{array}{c}
 [\varphi]^n \quad [\psi]^n \\
 \vdots \quad \vdots \\
 \varphi \vee \psi \quad \chi \quad \chi \\
 \chi \vee \text{Elim}
 \end{array}$$

Regras para $\rightarrow$

$$\begin{array}{c}
 [\varphi]^n \\
 \vdots \\
 \psi \\
 n \frac{\psi}{\varphi \rightarrow \psi} \rightarrow \text{Intro}
 \end{array}
 \qquad
 \frac{\varphi \rightarrow \psi \quad \varphi}{\psi} \rightarrow \text{Elim}$$

Regras para $\neg$

$$\begin{array}{c}
 [\varphi]^n \\
 \vdots \\
 \perp \\
 n \frac{\perp}{\neg \varphi} \neg \text{Intro}
 \end{array}
 \qquad
 \frac{\neg \varphi \quad \varphi}{\perp} \neg \text{Elim}$$

Regras para $\perp$

$$\frac{\perp}{\varphi} \perp_I
 \qquad
 \begin{array}{c}
 [\neg \varphi]^n \\
 \vdots \\
 \perp \\
 n \frac{\perp}{\varphi} \perp_C
 \end{array}$$

10.3. DERIVAÇÕES

Note que $\neg$ Intro e $\perp_C$ são muito semelhantes: a diferença é que $\neg$ Intro deriva uma **sentença** negada $\neg\varphi$, mas $\perp_C$, uma **sentença** positiva φ .

Sempre que uma regra indica que alguma hipótese pode ser descarregada, nós tomamos isso como uma permissão, mas não uma exigência. Por exemplo, na regra $\rightarrow$ Intro, podemos descarregar qualquer número de hipóteses da forma φ na **derivação** da premissa ψ , inclusive zero.

`content/propositional-logic/./first-order-logic/natural-deduction/derivations.tex`

10.3 Derivações

pl:ntd:der:
sec Já dissemos o que é uma hipótese e demos as regras de inferência. **Derivações** explanation em dedução natural são geradas indutivamente a partir disso: cada **derivação** ou é uma hipótese por si só, ou consiste em uma, duas ou três **derivações** seguidas de uma inferência correta.

Definição 10.2 (Derivação). A **derivação** de a **sentença** φ a partir de hipóteses Γ é uma árvore finita de **sentenças** que satisfaz as seguintes condições:

1. As **sentenças** no topo da árvore ou estão em Γ ou são **descarregadas** por uma inferência na árvore.
2. A **sentença** na base da árvore é φ .
3. Toda **sentença** na árvore exceto a sentença φ na base é uma premissa de uma aplicação correta de uma regra de inferência cuja conclusão está diretamente abaixo daquela **sentença** na árvore.

Dizemos então que φ é a **conclusão** da **derivação** e Γ , suas hipóteses **não descarregadas**.

Se existe a **derivação** de φ a partir de Γ , dizemos que φ é **derivável** a partir de Γ , ou, em símbolos: $\Gamma \vdash \varphi$. Se há a **derivação** de φ em que toda hipótese é **descarregada**, escrevemos $\vdash \varphi$.

Exemplo 10.3. Toda hipótese por si só é a **derivação**. Assim, por exemplo, φ por si só é a **derivação**, e ψ por si só também. Podemos obter uma nova **derivação** a partir delas aplicando, digamos, a regra $\wedge$ Intro,

$$\frac{\varphi \quad \psi}{\varphi \wedge \psi} \wedge \text{Intro}$$

Essas regras devem ser gerais: podemos substituir o φ e ψ nela por quaisquer **sentenças**, por exemplo, por χ e θ . Então a conclusão seria $\chi \wedge \theta$, e assim

$$\frac{\chi \quad \theta}{\chi \wedge \theta} \wedge \text{Intro}$$

é uma **derivação** correta. É claro que também podemos trocar as hipóteses, de modo que θ desempenhe o papel de φ e χ o de ψ . Assim,

$$\frac{\theta \quad \chi}{\theta \wedge \chi} \wedge \text{Intro}$$

também é uma **derivação** correta.

Podemos agora aplicar outra regra, digamos, $\rightarrow$ Intro, que nos permite concluir um condicional e nos permite descarregar qualquer hipótese que seja idêntica ao antecedente desse condicional. Assim, ambas as seguintes seriam **derivações** corretas:

$$1 \frac{\frac{[\chi]^1 \quad \theta}{\chi \wedge \theta} \wedge \text{Intro}}{\chi \rightarrow (\chi \wedge \theta)} \rightarrow \text{Intro} \quad 1 \frac{\frac{\chi \quad [\theta]^1}{\chi \wedge \theta} \wedge \text{Intro}}{\theta \rightarrow (\chi \wedge \theta)} \rightarrow \text{Intro}$$

Elas mostram, respectivamente, que $\theta \vdash \chi \rightarrow (\chi \wedge \theta)$ e $\chi \vdash \theta \rightarrow (\chi \wedge \theta)$.

Lembre-se de que descarregar hipóteses é uma permissão, não uma exigência: não temos de descarregar as hipóteses. Em particular, podemos aplicar uma regra mesmo que as hipóteses não estejam presentes na **derivação**. Por exemplo, o seguinte é válido, embora não haja hipótese φ a ser **descarregada**:

$$1 \frac{\psi}{\varphi \rightarrow \psi} \rightarrow \text{Intro}$$

`content/propositional-logic/.../first-order-logic/natural-deduction/proving-things.t`

10.4 Exemplos de Derivações

Exemplo 10.4. Vamos dar a **derivação** da **sentença** $(\varphi \wedge \psi) \rightarrow \varphi$.

pl:ntd:pro:
sec

Começamos escrevendo a conclusão desejada na parte inferior da **derivação**.

$$\overline{(\varphi \wedge \psi) \rightarrow \varphi}$$

Em seguida, precisamos descobrir que tipo de inferência poderia resultar em a **sentença** desta forma. O **operador principal** da conclusão é $\rightarrow$, então tentaremos chegar à conclusão usando a regra $\rightarrow$ Intro. O melhor é anotar as suposições envolvidas e rotular as regras de inferência conforme avançamos, para que seja fácil ver se todas as suposições foram **descarregadas** ao final da prova.

$$1 \frac{\begin{array}{c} [\varphi \wedge \psi]^1 \\ \vdots \\ \vdots \\ \varphi \end{array}}{(\varphi \wedge \psi) \rightarrow \varphi} \rightarrow \text{Intro}$$

10.4. EXEMPLOS DE DERIVAÇÕES

Agora precisamos preencher os passos da suposição $\varphi \wedge \psi$ até φ . Como só temos um conectivo com o qual lidar, $\wedge$, devemos usar a regra $\wedge$ elim. Isso nos dá a seguinte prova:

$$1 \frac{\frac{[\varphi \wedge \psi]^1}{\varphi} \wedge \text{Elim}}{(\varphi \wedge \psi) \rightarrow \varphi} \rightarrow \text{Intro}$$

Agora temos uma **derivação** correta de $(\varphi \wedge \psi) \rightarrow \varphi$.

Exemplo 10.5. Agora vamos dar a **derivação** de $(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)$.

Começamos escrevendo a conclusão desejada na parte inferior da **derivação**.

$$\overline{(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)}$$

Para encontrar uma regra lógica que possa nos dar esta conclusão, observamos os conectivos lógicos na conclusão: $\neg$, $\vee$ e $\rightarrow$. Por enquanto, só nos interessa a primeira ocorrência de $\rightarrow$, porque é o **operador principal** da **sentença** na conclusão, enquanto $\neg$, $\vee$ e a segunda ocorrência de $\rightarrow$ estão dentro do escopo de outro conectivo, então cuidaremos deles depois. Começamos, portanto, com a regra $\rightarrow$ Intro. Uma aplicação correta deve ser assim:

$$1 \frac{\begin{array}{c} [\neg\varphi \vee \psi]^1 \\ \vdots \\ \varphi \rightarrow \psi \end{array}}{(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)} \rightarrow \text{Intro}$$

Isso nos deixa com duas possibilidades para continuar. Podemos continuar trabalhando de baixo para cima e procurar outra aplicação da regra $\rightarrow$ Intro, ou podemos trabalhar de cima para baixo e aplicar a regra $\vee$ Elim. Vamos adotar a segunda opção. Usaremos a suposição $\neg\varphi \vee \psi$ como a premissa mais à esquerda de $\vee$ Elim. Para uma aplicação válida de $\vee$ Elim, as outras duas premissas devem ser idênticas à conclusão $\varphi \rightarrow \psi$, mas cada uma pode ser derivada, por sua vez, de outra suposição, a saber, um dos dois disjuntos de $\neg\varphi \vee \psi$. Nossa **derivação** ficará, então, assim:

$$2 \frac{\begin{array}{c} [\neg\varphi]^2 \\ \vdots \\ \varphi \rightarrow \psi \end{array} \quad \begin{array}{c} [\psi]^2 \\ \vdots \\ \varphi \rightarrow \psi \end{array}}{\varphi \rightarrow \psi} \vee \text{Elim} \\ 1 \frac{\frac{[\neg\varphi \vee \psi]^1}{\varphi \rightarrow \psi} \vee \text{Elim}}{(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)} \rightarrow \text{Intro}$$

Em cada um dos dois ramos à direita, queremos derivar $\varphi \rightarrow \psi$, o que é melhor feito usando $\rightarrow$ Intro.

$$\begin{array}{c}
 \begin{array}{c}
 [\neg\varphi]^2, [\varphi]^3 \\
 \vdots \\
 \psi \\
 3 \frac{}{\varphi \rightarrow \psi} \rightarrow\text{Intro}
 \end{array}
 \quad
 \begin{array}{c}
 [\psi]^2, [\varphi]^4 \\
 \vdots \\
 \psi \\
 4 \frac{}{\varphi \rightarrow \psi} \rightarrow\text{Intro}
 \end{array} \\
 2 \frac{[\neg\varphi \vee \psi]^1}{\varphi \rightarrow \psi} \vee\text{Elim} \\
 1 \frac{}{(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)} \rightarrow\text{Intro}
 \end{array}$$

Nas duas partes que faltam na **derivação**, precisamos de **derivações** de ψ a partir de $\neg\varphi$ e φ no meio, e a partir de φ e ψ à esquerda. Começemos pelo primeiro caso. $\neg\varphi$ e φ são as duas premissas de $\neg\text{Elim}$:

$$\begin{array}{c}
 \frac{[\neg\varphi]^2 \quad [\varphi]^3}{\perp} \neg\text{Elim} \\
 \vdots \\
 \psi
 \end{array}$$

Usando $\perp_I$, podemos obter ψ como conclusão e completar o ramo.

$$\begin{array}{c}
 \begin{array}{c}
 [\neg\varphi]^2 \quad [\varphi]^3 \\
 \frac{}{\perp} \neg\text{Elim} \\
 \perp_I \\
 \psi \\
 3 \frac{}{\varphi \rightarrow \psi} \rightarrow\text{Intro}
 \end{array}
 \quad
 \begin{array}{c}
 [\psi]^2, [\varphi]^4 \\
 \vdots \\
 \psi \\
 4 \frac{}{\varphi \rightarrow \psi} \rightarrow\text{Intro}
 \end{array} \\
 2 \frac{[\neg\varphi \vee \psi]^1}{\varphi \rightarrow \psi} \vee\text{Elim} \\
 1 \frac{}{(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)} \rightarrow\text{Intro}
 \end{array}$$

Vejamos agora o ramo mais à direita. Aqui é importante perceber que a definição de **derivação** *permite que hipóteses sejam descarregadas* mas *não exige* que o sejam. Em outras palavras, se pudermos derivar ψ a partir de uma das suposições φ e ψ sem usar a outra, tudo bem. E para derivar ψ a partir de ψ é trivial: ψ por si só é tal **a derivação**, e nenhuma inferência é necessária. Portanto, podemos simplesmente eliminar a suposição φ .

$$\begin{array}{c}
 \begin{array}{c}
 [\neg\varphi]^2 \quad [\varphi]^3 \\
 \frac{}{\perp} \neg\text{Elim} \\
 \perp_I \\
 \psi \\
 3 \frac{}{\varphi \rightarrow \psi} \rightarrow\text{Intro}
 \end{array}
 \quad
 \begin{array}{c}
 [\psi]^2 \\
 \frac{}{\varphi \rightarrow \psi} \rightarrow\text{Intro}
 \end{array} \\
 2 \frac{[\neg\varphi \vee \psi]^1}{\varphi \rightarrow \psi} \vee\text{Elim} \\
 1 \frac{}{(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)} \rightarrow\text{Intro}
 \end{array}$$

Observe que, na **derivação** concluída, a inferência $\rightarrow\text{Intro}$ mais à direita na verdade não descarrega nenhuma hipótese.

10.4. EXEMPLOS DE DERIVAÇÕES

Exemplo 10.6. Até agora não precisamos da regra $\perp_C$. Ela é especial porque nos permite descarregar uma hipótese que não é uma sub-fórmula da conclusão da regra. Está intimamente relacionada à regra $\perp_I$. Na verdade, a regra $\perp_I$ é um caso especial da regra $\perp_C$ —existe uma lógica chamada “lógica intuicionista” na qual apenas $\perp_I$ é permitida. A regra $\perp_C$ é um último recurso quando nada mais funciona. Por exemplo, suponha que queiramos derivar $\varphi \vee \neg\varphi$. Nossa estratégia usual seria tentar derivar $\varphi \vee \neg\varphi$ usando $\vee\text{Intro}$. Mas isso exigiria derivar φ ou $\neg\varphi$ a partir de nenhuma suposição, e isso não pode ser feito. $\perp_C$ ao resgate!

$$\begin{array}{c} [\neg(\varphi \vee \neg\varphi)]^1 \\ \vdots \\ 1 \frac{\perp}{\varphi \vee \neg\varphi} \perp_C \end{array}$$

Agora procuramos a derivação de $\perp$ a partir de $\neg(\varphi \vee \neg\varphi)$. Como $\perp$ é a conclusão de $\neg\text{Elim}$, podemos tentar isso:

$$\begin{array}{c} [\neg(\varphi \vee \neg\varphi)]^1 \quad [\neg(\varphi \vee \neg\varphi)]^1 \\ \vdots \quad \vdots \\ \neg\varphi \quad \varphi \\ \hline 1 \frac{\perp}{\varphi \vee \neg\varphi} \perp_C \quad \neg\text{Elim} \end{array}$$

Nossa estratégia para encontrar a derivação de $\neg\varphi$ exige uma aplicação de $\neg\text{Intro}$:

$$\begin{array}{c} [\neg(\varphi \vee \neg\varphi)]^1, [\varphi]^2 \quad [\neg(\varphi \vee \neg\varphi)]^1 \\ \vdots \quad \vdots \\ 2 \frac{\perp}{\neg\varphi} \neg\text{Intro} \quad \varphi \\ \hline 1 \frac{\perp}{\varphi \vee \neg\varphi} \perp_C \quad \neg\text{Elim} \end{array}$$

Aqui, podemos obter $\perp$ facilmente aplicando $\neg\text{Elim}$ à suposição $\neg(\varphi \vee \neg\varphi)$ e a $\varphi \vee \neg\varphi$, que segue da nossa nova suposição φ por $\vee\text{Intro}$:

$$\begin{array}{c} [\neg(\varphi \vee \neg\varphi)]^1 \quad \frac{[\varphi]^2}{\varphi \vee \neg\varphi} \vee\text{Intro} \quad [\neg(\varphi \vee \neg\varphi)]^1 \\ \vdots \quad \vdots \\ 2 \frac{\perp}{\neg\varphi} \neg\text{Intro} \quad \varphi \\ \hline 1 \frac{\perp}{\varphi \vee \neg\varphi} \perp_C \quad \neg\text{Elim} \end{array}$$

No lado direito usamos a mesma estratégia, exceto que obtemos φ por $\perp_C$:

$$\frac{\frac{\frac{[\neg(\varphi \vee \neg\varphi)]^1}{\perp} \neg\text{Intro}}{\frac{[\varphi]^2}{\varphi \vee \neg\varphi} \vee\text{Intro}} \neg\text{Elim} \quad \frac{\frac{[\neg\varphi]^3}{\varphi \vee \neg\varphi} \vee\text{Intro}}{\frac{[\neg(\varphi \vee \neg\varphi)]^1}{\perp} \neg\text{Intro}} \neg\text{Elim}}{\frac{\perp}{\varphi \vee \neg\varphi} \perp_C} \neg\text{Elim}$$

Problema 10.1. Dê **derivações** que mostrem o seguinte:

1. $\varphi \wedge (\psi \wedge \chi) \vdash (\varphi \wedge \psi) \wedge \chi$.
2. $\varphi \vee (\psi \vee \chi) \vdash (\varphi \vee \psi) \vee \chi$.
3. $\varphi \rightarrow (\psi \rightarrow \chi) \vdash \psi \rightarrow (\varphi \rightarrow \chi)$.
4. $\varphi \vdash \neg\neg\varphi$.

Problema 10.2. Dê **derivações** que mostrem o seguinte:

1. $(\varphi \vee \psi) \rightarrow \chi \vdash \varphi \rightarrow \chi$.
2. $(\varphi \rightarrow \chi) \wedge (\psi \rightarrow \chi) \vdash (\varphi \vee \psi) \rightarrow \chi$.
3. $\vdash \neg(\varphi \wedge \neg\varphi)$.
4. $\psi \rightarrow \varphi \vdash \neg\varphi \rightarrow \neg\psi$.
5. $\vdash (\varphi \rightarrow \neg\varphi) \rightarrow \neg\varphi$.
6. $\vdash \neg(\varphi \rightarrow \psi) \rightarrow \neg\psi$.
7. $\varphi \rightarrow \chi \vdash \neg(\varphi \wedge \neg\chi)$.
8. $\varphi \wedge \neg\chi \vdash \neg(\varphi \rightarrow \chi)$.
9. $\varphi \vee \psi, \neg\psi \vdash \varphi$.
10. $\neg\varphi \vee \neg\psi \vdash \neg(\varphi \wedge \psi)$.
11. $\vdash (\neg\varphi \wedge \neg\psi) \rightarrow \neg(\varphi \vee \psi)$.
12. $\vdash \neg(\varphi \vee \psi) \rightarrow (\neg\varphi \wedge \neg\psi)$.

Problema 10.3. Dê **derivações** que mostrem o seguinte:

1. $\neg(\varphi \rightarrow \psi) \vdash \varphi$.
2. $\neg(\varphi \wedge \psi) \vdash \neg\varphi \vee \neg\psi$.
3. $\varphi \rightarrow \psi \vdash \neg\varphi \vee \psi$.
4. $\vdash \neg\neg\varphi \rightarrow \varphi$.
5. $\varphi \rightarrow \psi, \neg\varphi \rightarrow \psi \vdash \psi$.

10.5. NOÇÕES DA TEORIA DA PROVA

6. $(\varphi \wedge \psi) \rightarrow \chi \vdash (\varphi \rightarrow \chi) \vee (\psi \rightarrow \chi)$.

7. $(\varphi \rightarrow \psi) \rightarrow \varphi \vdash \varphi$.

8. $\vdash (\varphi \rightarrow \psi) \vee (\psi \rightarrow \chi)$.

(Todos exigem a regra $\perp_C$.)

[content/propositional-logic/./first-order-logic/natural-deduction/proof-theoretic-notions.tex](#)

10.5 Noções da Teoria da Prova

pl:ntd:ptn:
sec

Esta seção reúne as definições da relação de demonstrabilidade e de consistência para a dedução natural.

Assim como definimos várias noções semânticas importantes (validade, consequência, satisfatibilidade), agora definimos noções *da Teoria da Prova* correspondentes. Elas não são definidas por apelo à satisfação de **sentenças** em **estruturas**, mas por apelo à **derivabilidade** ou **não derivabilidade** de certas **sentenças** a partir de outras. Foi uma descoberta importante que essas noções coincidem. Que coincidem é o conteúdo dos teoremas de *correção* e *completude*.

explanation

Definição 10.7 (Teoremas). Uma **sentença** φ é um *teorema* se há uma **derivação** de φ em dedução natural na qual todas as hipóteses são **descarregadas**. Escrevemos $\vdash \varphi$ se φ é um teorema e $\nvdash \varphi$ se não o é.

Definição 10.8 (Derivabilidade). A **sentença** φ é **derivável** de um conjunto de **sentenças** Γ , $\Gamma \vdash \varphi$, se há uma **derivação** com conclusão φ e na qual toda hipótese é ou **descarregada** ou está em Γ . Se φ não é **derivável** de Γ , escrevemos $\Gamma \nvdash \varphi$.

Definição 10.9 (Consistência). Um conjunto de **sentenças** Γ é *inconsistente* se, e somente se, $\Gamma \vdash \perp$. Se Γ não é inconsistente, isto é, se $\Gamma \nvdash \perp$, dizemos que é *consistente*.

pl:ntd:ptn:
prop:reflexivity

Proposição 10.10 (Reflexividade). Se $\varphi \in \Gamma$, então $\Gamma \vdash \varphi$.

Prova. A hipótese φ por si só é uma **derivação** de φ em que toda hipótese **não descarregada** (isto é, φ) está em Γ . $\square$

pl:ntd:ptn:
prop:monotonicity

Proposição 10.11 (Monotonicidade). Se $\Gamma \subseteq \Delta$ e $\Gamma \vdash \varphi$, então $\Delta \vdash \varphi$.

Prova. Qualquer **derivação** de φ a partir de Γ é também uma **derivação** de φ a partir de Δ . $\square$

Proposição 10.12 (Transitividade). *Se $\Gamma \vdash \varphi$ e $\{\varphi\} \cup \Delta \vdash \psi$, então $\Gamma \cup \Delta \vdash \psi$.* pl:ntd;ptn:
prop:transitivity

Prova. Se $\Gamma \vdash \varphi$, há uma **derivação** δ_0 de φ com todas as hipóteses **não descarregadas** em Γ . Se $\{\varphi\} \cup \Delta \vdash \psi$, então há uma **derivação** δ_1 de ψ com todas as hipóteses **não descarregadas** em $\{\varphi\} \cup \Delta$. Considere agora:

$$\begin{array}{c}
 \Delta, [\varphi]^1 \\
 \vdots \delta_1 \\
 \vdots \\
 \psi \\
 \hline
 1 \frac{\varphi \rightarrow \psi}{\varphi \rightarrow \psi} \rightarrow \text{Intro} \quad \begin{array}{c} \Gamma \\ \vdots \\ \delta_0 \\ \vdots \\ \varphi \end{array} \\
 \hline
 \psi \rightarrow \text{Elim}
 \end{array}$$

As hipóteses **não descarregadas** estão agora todas entre $\Gamma \cup \Delta$, de modo que isto mostra $\Gamma \cup \Delta \vdash \psi$. $\square$

Quando $\Gamma = \{\varphi_1, \varphi_2, \dots, \varphi_k\}$ é um conjunto finito, podemos usar a notação simplificada $\varphi_1, \varphi_2, \dots, \varphi_k \vdash \psi$ para $\Gamma \vdash \psi$; em particular, $\varphi \vdash \psi$ significa que $\{\varphi\} \vdash \psi$.

Note que se $\Gamma \vdash \varphi$ e $\varphi \vdash \psi$, então $\Gamma \vdash \psi$. Segue também que se $\varphi_1, \dots, \varphi_n \vdash \psi$ e $\Gamma \vdash \varphi_i$ para cada i , então $\Gamma \vdash \psi$.

Proposição 10.13. *As seguintes afirmações são equivalentes.* pl:ntd;ptn:
prop:incons

1. Γ é inconsistente.
2. $\Gamma \vdash \varphi$ para toda **sentença** φ .
3. $\Gamma \vdash \varphi$ e $\Gamma \vdash \neg \varphi$ para alguma **sentença** φ .

Prova. Exercício. $\square$

Problema 10.4. Prove **Proposição 10.13**

Proposição 10.14 (Compacidade). pl:ntd;ptn:
prop:proves-compact

1. Se $\Gamma \vdash \varphi$, então há um subconjunto finito $\Gamma_0 \subseteq \Gamma$ tal que $\Gamma_0 \vdash \varphi$.
2. Se todo subconjunto finito de Γ é consistente, então Γ é consistente.

Prova. 1. Se $\Gamma \vdash \varphi$, então há uma **derivação** δ de φ a partir de Γ . Seja Γ_0 o conjunto de hipóteses **não descarregadas** de δ . Como toda **derivação** é finita, Γ_0 só pode conter finitamente muitas **sentenças**. Assim, δ é uma **derivação** de φ a partir de um Γ_0 finito $\subseteq \Gamma$.

2. Isto é a contrapositiva de (1) no caso especial $\varphi \equiv \perp$. $\square$

10.6 Derivabilidade e Consistência

pl:ntd:prv:sec Estabeleceremos agora várias propriedades da relação de **derivabilidade**. Elas são independentemente interessantes, mas cada uma desempenhará um papel na prova do teorema da completude.

pl:ntd:prv:prop:provability-contr **Proposição 10.15.** *Se $\Gamma \vdash \varphi$ e $\Gamma \cup \{\varphi\}$ é inconsistente, então Γ é inconsistente.*

Prova. Seja a **derivação** de φ a partir de Γ igual a δ_1 e a **derivação** de $\perp$ a partir de $\Gamma \cup \{\varphi\}$ igual a δ_2 . Podemos então derivar:

$$\frac{\begin{array}{c} \Gamma, [\varphi]^1 \\ \vdots \\ \delta_2 \\ \vdots \\ \perp \\ 1 \frac{}{\neg\varphi} \neg\text{Intro} \end{array}}{\neg\varphi} \neg\text{Elim} \quad \frac{\begin{array}{c} \Gamma \\ \vdots \\ \delta_1 \\ \vdots \\ \varphi \end{array}}{\varphi} \neg\text{Elim} \quad \frac{}{\perp}$$

Na nova **derivação**, a hipótese φ é **descarregada**, de modo que é uma **derivação** a partir de Γ . □

pl:ntd:prv:prop:prov-incons **Proposição 10.16.** *$\Gamma \vdash \varphi$ se, e somente se, $\Gamma \cup \{\neg\varphi\}$ é inconsistente.*

Prova. Primeiro suponha que $\Gamma \vdash \varphi$, isto é, há uma **derivação** δ_0 de φ a partir de hipóteses **não descarregadas** Γ . Obtemos uma **derivação** de $\perp$ a partir de $\Gamma \cup \{\neg\varphi\}$ da seguinte forma:

$$\frac{\begin{array}{c} \Gamma \\ \vdots \\ \delta_0 \\ \vdots \\ \varphi \end{array}}{\varphi} \neg\text{Elim} \quad \frac{}{\neg\varphi} \neg\text{Intro} \quad \frac{}{\perp}$$

Agora suponha que $\Gamma \cup \{\neg\varphi\}$ é inconsistente, e seja δ_1 a **derivação** correspondente de $\perp$ a partir de hipóteses **não descarregadas** em $\Gamma \cup \{\neg\varphi\}$. Obtemos uma **derivação** de φ apenas a partir de Γ usando $\perp_C$:

$$\frac{\begin{array}{c} \Gamma, [\neg\varphi]^1 \\ \vdots \\ \delta_1 \\ \vdots \\ \perp \\ 1 \frac{}{\varphi} \perp_C \end{array}}{\varphi} \perp_C \quad \square$$

Problema 10.5. Prove que $\Gamma \vdash \neg\varphi$ se, e somente se, $\Gamma \cup \{\varphi\}$ é inconsistente.

pl:ntd:prv:prop:explicit-inc **Proposição 10.17.** *Se $\Gamma \vdash \varphi$ e $\neg\varphi \in \Gamma$, então Γ é inconsistente.*

Prova. Suponha que $\Gamma \vdash \varphi$ e $\neg\varphi \in \Gamma$. Então há uma **derivação** δ de φ a partir de Γ . Considere esta aplicação simples da regra $\neg$ Elim:

$$\frac{\begin{array}{c} \Gamma \\ \vdots \\ \delta \\ \vdots \\ \varphi \end{array} \quad \neg\varphi}{\perp} \neg\text{Elim}$$

Como $\neg\varphi \in \Gamma$, todas as hipóteses **não descarregadas** estão em Γ ; isto mostra que $\Gamma \vdash \perp$. $\square$

Proposição 10.18. *Se $\Gamma \cup \{\varphi\}$ e $\Gamma \cup \{\neg\varphi\}$ são ambos inconsistentes, então Γ é inconsistente.*

*pl:ntd:prv:
prop:provability-exhaustive*

Prova. Há **derivações** δ_1 e δ_2 de $\perp$ a partir de $\Gamma \cup \{\varphi\}$ e de $\perp$ a partir de $\Gamma \cup \{\neg\varphi\}$, respectivamente. Podemos então derivar

$$\frac{\begin{array}{c} \Gamma, [\neg\varphi]^2 \\ \vdots \\ \delta_2 \\ \vdots \\ \perp \end{array} \quad \begin{array}{c} \Gamma, [\varphi]^1 \\ \vdots \\ \delta_1 \\ \vdots \\ \perp \end{array}}{\frac{2 \quad \frac{\perp}{\neg\neg\varphi} \neg\text{Intro} \quad 1 \quad \frac{\perp}{\neg\varphi} \neg\text{Intro}}{\perp} \neg\text{Elim}}$$

Como as hipóteses φ e $\neg\varphi$ são **descarregadas**, isto é uma **derivação** de $\perp$ apenas a partir de Γ . Logo Γ é inconsistente. $\square$

<content/propositional-logic/.../first-order-logic/natural-deduction/provability-prop>

10.7 Derivabilidade e os Conectivos Proposicionais

explanation Estabelecemos que a relação de **derivabilidade** $\vdash$ da dedução natural é forte o suficiente para estabelecer alguns fatos básicos envolvendo os conectivos proposicionais, tais como $\varphi \wedge \psi \vdash \varphi$ e $\varphi, \varphi \rightarrow \psi \vdash \psi$ (modus ponens). Esses fatos são necessários para a prova do teorema da completude.

*pl:ntd:ppr:
sec*

Proposição 10.19.

*pl:ntd:ppr:
prop:provability-land
pl:ntd:ppr:
prop:provability-land-left
pl:ntd:ppr:
prop:provability-land-right*

1. Tanto $\varphi \wedge \psi \vdash \varphi$ quanto $\varphi \wedge \psi \vdash \psi$.

2. $\varphi, \psi \vdash \varphi \wedge \psi$.

Prova. 1. Podemos derivar ambos:

$$\frac{\varphi \wedge \psi}{\varphi} \wedge\text{Elim} \quad \frac{\varphi \wedge \psi}{\psi} \wedge\text{Elim}$$

10.7. DERIVABILIDADE E OS CONECTIVOS PROPOSICIONAIS

2. Podemos derivar:

$$\frac{\varphi \quad \psi}{\varphi \wedge \psi} \wedge \text{Intro} \quad \square$$

*pl:ntd:ppr:
prop:provability-lor*

Proposição 10.20.

1. $\varphi \vee \psi, \neg\varphi, \neg\psi$ é inconsistente.
2. Tanto $\varphi \vdash \varphi \vee \psi$ quanto $\psi \vdash \varphi \vee \psi$.

Prova. 1. Considere a seguinte **derivação**:

$$\begin{array}{c} \frac{\frac{\frac{\neg\varphi}{\perp} \quad [\varphi]^1}{\neg\text{Elim}} \quad \frac{\frac{\neg\psi}{\perp} \quad [\psi]^1}{\neg\text{Elim}}}{\frac{\varphi \vee \psi \quad \perp}{\perp} \vee\text{Elim}} \end{array}$$

Esta é a **derivação** de $\perp$ a partir das hipóteses não descarregadas $\varphi \vee \psi$, $\neg\varphi$ e $\neg\psi$.

2. Podemos derivar ambos:

$$\frac{\varphi}{\varphi \vee \psi} \vee\text{Intro} \quad \frac{\psi}{\varphi \vee \psi} \vee\text{Intro} \quad \square$$

*pl:ntd:ppr:
prop:provability-lif
pl:ntd:ppr:
prop:provability-lif-left
pl:ntd:ppr:
prop:provability-lif-right*

Proposição 10.21.

1. $\varphi, \varphi \rightarrow \psi \vdash \psi$.
2. Tanto $\neg\varphi \vdash \varphi \rightarrow \psi$ quanto $\psi \vdash \varphi \rightarrow \psi$.

Prova. 1. Podemos derivar:

$$\frac{\varphi \rightarrow \psi \quad \varphi}{\psi} \rightarrow\text{Elim}$$

2. Isso é mostrado pelas duas **derivações** seguintes:

$$\begin{array}{c} \frac{\frac{\neg\varphi}{\perp} \quad [\varphi]^1}{\perp} \neg\text{Elim} \\ \frac{\frac{\psi}{\varphi \rightarrow \psi} \quad \perp_I}{\varphi \rightarrow \psi} \rightarrow\text{Intro} \end{array} \quad \frac{\psi}{\varphi \rightarrow \psi} \rightarrow\text{Intro}$$

Note que $\rightarrow\text{Intro}$ pode, mas não precisa, descarregar a hipótese φ . $\square$

10.8 Correção

explanation Um sistema de **derivação**, como a dedução natural, é *correto* se não pode derivar coisas que de fato não seguem. A correção é assim uma espécie de propriedade de segurança garantida para sistemas de **derivação**. Dependendo de qual propriedade da Teoria da Prova está em questão, gostaríamos de saber, por exemplo, que

pl:ntd:sou:
sec

1. toda **sentença derivável** é uma tautologia;
2. se uma **sentença** é **derivável** a partir de outras, ela também é uma consequência delas;
3. se um conjunto de **sentenças** é inconsistente, ele é insatisfatível.

Estas são propriedades importantes de um sistema de **derivação**. Se alguma delas não vale, o sistema de **derivação** é deficiente—ele **deriva** demais. Consequentemente, estabelecer a correção de um sistema de **derivação** é da maior importância.

Teorema 10.22 (Correção). *Se φ é **derivável** a partir das hipóteses **não descarregadas** Γ , então $\Gamma \models \varphi$.*

pl:ntd:sou:
thm:soundness

Prova. Seja δ uma **derivação** de φ . Procedemos por indução no número de inferências em δ .

Para a base da indução, mostramos a afirmação quando o número de inferências é 0. Neste caso, δ consiste apenas de uma única **sentença** φ , isto é, uma hipótese. Essa hipótese é **não descarregada**, pois hipóteses só podem ser **descarregadas** por inferências, e não há inferências. Assim, qualquer **valoração** $\mathbf{v}$ que satisfaz todas as hipóteses **não descarregadas** da prova também satisfaz φ .

Agora o passo indutivo. Suponha que δ contém n inferências. A(s) premissa(s) da inferência mais baixa são **derivadas** usando sub-**derivações**, cada uma das quais contém menos de n inferências. Assumimos a hipótese de indução: as premissas da inferência mais baixa seguem das hipóteses **não descarregadas** das sub-**derivações** que terminam nessas premissas. Temos que mostrar que a conclusão φ segue das hipóteses **não descarregadas** de toda a prova.

Distinguimos casos de acordo com o tipo da inferência mais baixa. Primeiro, consideramos as possíveis inferências com apenas uma premissa.

1. Suponha que a última inferência é $\neg$ -Intro: a **derivação** tem a forma

$$\begin{array}{c} \Gamma, [\varphi]^n \\ \vdots \\ \delta_1 \\ \vdots \\ \perp \\ n \frac{}{\neg\varphi} \neg\text{Intro} \end{array}$$

10.8. CORREÇÃO

Pela hipótese de indução, $\perp$ segue das hipóteses **não descarregadas** $\Gamma \cup \{\varphi\}$ de δ_1 . Considere a **valoração** $\mathbf{v}$. Precisamos mostrar que, se $\mathbf{v} \models \Gamma$, então $\mathbf{v} \models \neg\varphi$. Suponha por redução ao absurdo que $\mathbf{v} \models \Gamma$, mas $\mathbf{v} \not\models \neg\varphi$, isto é, $\mathbf{v} \models \varphi$. Isto significaria que $\mathbf{v} \models \Gamma \cup \{\varphi\}$. Isto é contrário à nossa hipótese de indução. Logo, $\mathbf{v} \models \neg\varphi$.

2. A última inferência é $\wedge$ Elim: há duas variantes: φ ou ψ pode ser inferido da premissa $\varphi \wedge \psi$. Considere o primeiro caso. A **derivação** δ tem este aspecto:

$$\frac{\begin{array}{c} \Gamma \\ \vdots \\ \delta_1 \\ \vdots \\ \varphi \wedge \psi \end{array}}{\varphi} \wedge\text{Elim}$$

Pela hipótese de indução, $\varphi \wedge \psi$ segue das hipóteses **não descarregadas** Γ de δ_1 . Considere a **estrutura** $\mathbf{v}$. Precisamos mostrar que, se $\mathbf{v} \models \Gamma$, então $\mathbf{v} \models \varphi$. Suponha $\mathbf{v} \models \Gamma$. Pela nossa hipótese de indução ($\Gamma \models \varphi \wedge \psi$), sabemos que $\mathbf{v} \models \varphi \wedge \psi$. Por definição, $\mathbf{v} \models \varphi \wedge \psi$ se, e somente se, $\mathbf{v} \models \varphi$ e $\mathbf{v} \models \psi$. (O caso em que ψ é inferido de $\varphi \wedge \psi$ é tratado de forma similar.)

3. A última inferência é $\vee$ Intro: há duas variantes: $\varphi \vee \psi$ pode ser inferido da premissa φ ou da premissa ψ . Considere o primeiro caso. A **derivação** tem a forma

$$\frac{\begin{array}{c} \Gamma \\ \vdots \\ \delta_1 \\ \vdots \\ \varphi \end{array}}{\varphi \vee \psi} \vee\text{Intro}$$

Pela hipótese de indução, φ segue das hipóteses **não descarregadas** Γ de δ_1 . Considere a **valoração** $\mathbf{v}$. Precisamos mostrar que, se $\mathbf{v} \models \Gamma$, então $\mathbf{v} \models \varphi \vee \psi$. Suponha $\mathbf{v} \models \Gamma$; então $\mathbf{v} \models \varphi$, pois $\Gamma \models \varphi$ (a hipótese de indução). Logo também deve ser o caso que $\mathbf{v} \models \varphi \vee \psi$. (O caso em que $\varphi \vee \psi$ é inferido de ψ é tratado de forma similar.)

4. A última inferência é $\rightarrow$ Intro: $\varphi \rightarrow \psi$ é inferido de uma subprova com hipótese φ e conclusão ψ , isto é,

$$\frac{\begin{array}{c} \Gamma, [\varphi]^n \\ \vdots \\ \delta_1 \\ \vdots \\ \psi \end{array}}{\varphi \rightarrow \psi} \rightarrow\text{Intro}$$

Pela hipótese de indução, ψ segue das hipóteses **não descarregadas** de δ_1 , isto é, $\Gamma \cup \{\varphi\} \models \psi$. Considere a **valoração** $\mathbf{v}$. As hipóteses **não descarregadas** de δ são apenas Γ , pois φ é descartada na última inferência. Precisamos mostrar que $\Gamma \models \varphi \rightarrow \psi$. Por redução ao absurdo, suponha que para alguma **valoração** $\mathbf{v}$, $\mathbf{v} \models \Gamma$ mas $\mathbf{v} \not\models \varphi \rightarrow \psi$. Assim, $\mathbf{v} \models \varphi$ e $\mathbf{v} \not\models \psi$. Mas por hipótese, ψ é uma consequência de $\Gamma \cup \{\varphi\}$, isto é, $\mathbf{v} \models \psi$, o que é uma contradição. Logo, $\Gamma \models \varphi \rightarrow \psi$.

5. A última inferência é $\perp_I$: aqui, δ termina em

$$\frac{\begin{array}{c} \Gamma \\ \vdots \\ \delta_1 \\ \vdots \\ \perp \end{array}}{\varphi} \perp_I$$

Pela hipótese de indução, $\Gamma \models \perp$. Temos que mostrar que $\Gamma \models \varphi$. Suponha o contrário; então para algum $\mathbf{v}$ temos $\mathbf{v} \models \Gamma$ e $\mathbf{v} \not\models \varphi$. Mas sempre temos $\mathbf{v} \models \perp$, de modo que isto significaria que $\Gamma \not\models \perp$, contrariando a hipótese de indução.

6. A última inferência é $\perp_C$: Exercício.

Agora consideremos as possíveis inferências com várias premissas: $\vee$ Elim, $\wedge$ Intro, e $\rightarrow$ Elim.

1. A última inferência é $\wedge$ Intro. $\varphi \wedge \psi$ é inferido das premissas φ e ψ e δ tem a forma

$$\frac{\begin{array}{c} \Gamma_1 \\ \vdots \\ \delta_1 \\ \vdots \\ \varphi \end{array} \quad \begin{array}{c} \Gamma_2 \\ \vdots \\ \delta_2 \\ \vdots \\ \psi \end{array}}{\varphi \wedge \psi} \wedge \text{Intro}$$

Pela hipótese de indução, φ segue das hipóteses **não descarregadas** Γ_1 de δ_1 e ψ segue das hipóteses **não descarregadas** Γ_2 de δ_2 . As hipóteses **não descarregadas** de δ são $\Gamma_1 \cup \Gamma_2$, de modo que temos que mostrar que $\Gamma_1 \cup \Gamma_2 \models \varphi \wedge \psi$. Considere a **valoração** $\mathbf{v}$ com $\mathbf{v} \models \Gamma_1 \cup \Gamma_2$. Como $\mathbf{v} \models \Gamma_1$, deve ser o caso que $\mathbf{v} \models \varphi$, pois $\Gamma_1 \models \varphi$, e como $\mathbf{v} \models \Gamma_2$, $\mathbf{v} \models \psi$, pois $\Gamma_2 \models \psi$. Juntas, $\mathbf{v} \models \varphi \wedge \psi$.

2. A última inferência é $\vee$ Elim: Exercício.

3. A última inferência é $\rightarrow$ Elim. ψ é inferido das premissas $\varphi \rightarrow \psi$ e φ . A **derivação** δ tem este aspecto:

$$\begin{array}{c}
\begin{array}{c} \Gamma_1 \\ \vdots \\ \delta_1 \\ \vdots \\ \varphi \rightarrow \psi \end{array} \quad \begin{array}{c} \Gamma_2 \\ \vdots \\ \delta_2 \\ \vdots \\ \varphi \end{array} \\
\hline
\psi \quad \rightarrow\text{Elim}
\end{array}
\quad \square$$

Pela hipótese de indução, $\varphi \rightarrow \psi$ segue das hipóteses **não descarregadas** Γ_1 de δ_1 e φ segue das hipóteses **não descarregadas** Γ_2 de δ_2 . Considere a **valoração** $\mathbf{v}$. Precisamos mostrar que, se $\mathbf{v} \models \Gamma_1 \cup \Gamma_2$, então $\mathbf{v} \models \psi$. Suponha $\mathbf{v} \models \Gamma_1 \cup \Gamma_2$. Como $\Gamma_1 \models \varphi \rightarrow \psi$, $\mathbf{v} \models \varphi \rightarrow \psi$. Como $\Gamma_2 \models \varphi$, temos $\mathbf{v} \models \varphi$. Isto significa que $\mathbf{v} \models \psi$ (pois se $\mathbf{v} \not\models \psi$, como $\mathbf{v} \models \varphi$, teríamos $\mathbf{v} \not\models \varphi \rightarrow \psi$, contradizendo $\mathbf{v} \models \varphi \rightarrow \psi$).

4. A última inferência é $\neg$ Elim: Exercício.

Problema 10.6. Complete a prova de **Teorema 10.22**.

pl:ntd:sou: cor:weak-soundness **Corolário 10.23.** Se $\vdash \varphi$, então φ é uma tautologia.

pl:ntd:sou: cor:consistency-soundness **Corolário 10.24.** Se Γ é satisfável, então é consistente.

Prova. Provamos a contrapositiva. Suponha que Γ não é consistente. Então $\Gamma \vdash \perp$, isto é, há uma **derivação** de $\perp$ a partir de hipóteses **não descarregadas** em Γ . Por **Teorema 10.22**, qualquer **valoração** $\mathbf{v}$ que satisfaz Γ deve satisfazer $\perp$. Como $\mathbf{v} \not\models \perp$ para toda **valoração** $\mathbf{v}$, nenhuma $\mathbf{v}$ pode satisfazer Γ , isto é, Γ não é satisfável. $\square$

Capítulo 11

Tableaux

This chapter presents a signed analytic tableaux system.
To include or exclude material relevant to natural deduction as a proof system, use the “prfTab” tag.

-NoValue-

 content/propositional-logic/./first-order-logic/tableaux/rules-and-proofs.tex

11.1 Rules and Tableaux

A **tableau** is a systematic survey of the possible ways a **sentença** can be true or false in a **estrutura**. The building blocks of a tableau are **fórmulas assinadas**: **sentenças** plus a truth value “sign,” either $\mathbb{T}$ or $\mathbb{F}$. These signed **fórmulas** are arranged in a (downward growing) tree. pl:tab:rul:
sec

Definição 11.1. A **fórmula assinada** is a pair consisting of a truth value and a **sentença**, O.s., either:

$$\mathbb{T}\varphi \text{ or } \mathbb{F}\varphi.$$

Intuitively, we might read $\mathbb{T}\varphi$ as “ φ might be true” and $\mathbb{F}\varphi$ as “ φ might be false” (in some **estrutura**).

Each **fórmula assinada** in the tree is either an *assumption* (which are listed at the very top of the tree), or it is obtained from a **fórmula assinada** above it by one of a number of rules of inference. There are two rules for each possible **operador principal** of the preceding **fórmula**, one for the case where the sign is $\mathbb{T}$, and one for the case where the sign is $\mathbb{F}$. Some rules allow the tree to branch, and some only add **fórmulas assinadas** to the branch. A rule may be (and often must be) applied not to the immediately preceding **fórmula assinada**, but to any **fórmula assinada** in the branch from the root to the place the rule is applied.

A branch is *closed* when it contains both $\mathbb{T}\varphi$ and $\mathbb{F}\varphi$. A closed **tableau** is one where every branch is closed. Under the intuitive interpretation, any branch describes a joint possibility, but $\mathbb{T}\varphi$ and $\mathbb{F}\varphi$ are not jointly possible. In other words, if a branch is closed, the possibility it describes has been ruled out. In particular, that means that a closed **tableau** rules out all possibilities of simultaneously making every assumption of the form $\mathbb{T}\varphi$ true and every assumption of the form $\mathbb{F}\varphi$ false.

A closed **tableau for** φ is a closed **tableau** with root $\mathbb{F}\varphi$. If such a closed **tableau** exists, all possibilities for φ being false have been ruled out; O.s., φ must be true in every **estrutura**.

-NoValue-

 content/propositional-logic/./first-order-logic/tableaux/propositional-rules.tex

11.2 Propositional Rules

pl:tab:prl:
sec

11.2. PROPOSITIONAL RULES

Rules for $\neg$

$$\frac{\mathbb{T} \neg \varphi}{\mathbb{F} \varphi} \neg \mathbb{T} \qquad \frac{\mathbb{F} \neg \varphi}{\mathbb{T} \varphi} \neg \mathbb{F}$$

Rules for $\wedge$

$$\frac{\mathbb{T} \varphi \wedge \psi}{\mathbb{T} \varphi} \wedge \mathbb{T} \qquad \frac{\mathbb{F} \varphi \wedge \psi}{\mathbb{F} \varphi \mid \mathbb{F} \psi} \wedge \mathbb{F}$$

Rules for $\vee$

$$\frac{\mathbb{T} \varphi \vee \psi}{\mathbb{T} \varphi \mid \mathbb{T} \psi} \vee \mathbb{T} \qquad \frac{\mathbb{F} \varphi \vee \psi}{\mathbb{F} \varphi} \vee \mathbb{F}$$

Rules for $\rightarrow$

$$\frac{\mathbb{T} \varphi \rightarrow \psi}{\mathbb{F} \varphi \mid \mathbb{T} \psi} \rightarrow \mathbb{T} \qquad \frac{\mathbb{F} \varphi \rightarrow \psi}{\mathbb{T} \varphi} \rightarrow \mathbb{F}$$

The Cut Rule

$$\frac{}{\mathbb{T} \varphi \mid \mathbb{F} \varphi} \text{Cut}$$

The Cut rule is not applied “to” a previous *fórmula assinada*; rather, it allows every branch in a *tableau* to be split in two, one branch containing $\mathbb{T} \varphi$, the other $\mathbb{F} \varphi$. It is not necessary—any set of *fórmulas assinadas* with a closed *tableau* has one not using Cut—but it allows us to combine *tableaux* in a convenient way.

-NoValue-

`content/propositional-logic/./first-order-logic/tableaux/derivations.tex`

11.3 Tableaux

explanation We've said what an assumption is, and we've given the rules of inference. pl:tab:der:sec **Tableaux** are inductively generated from these: each **tableau** either is a single branch consisting of one or more assumptions, or it results from a **tableau** by applying one of the rules of inference on a branch.

Definição 11.2 (Tableau). A **tableau** for assumptions $S_1\varphi_1, \dots, S_n\varphi_n$ (where each S_i is either $\mathbb{T}$ or $\mathbb{F}$) is a finite tree of **fórmulas assinadas** satisfying the following conditions:

1. The n topmost **fórmulas assinadas** of the tree are $S_i\varphi_i$, one below the other.
2. Every **fórmula assinada** in the tree that is not one of the assumptions results from a correct application of an inference rule to a **fórmula assinada** in the branch above it.

A branch of a **tableau** is *closed* iff it contains both $\mathbb{T}\varphi$ and $\mathbb{F}\varphi$, and *open* otherwise. A **tableau** in which every branch is closed is a *closed tableau* (for its set of assumptions). If a **tableau** is not closed, O.s., if it contains at least one open branch, it is *open*.

Exemplo 11.3. Every set of assumptions on its own is a **tableau**, but it will generally not be closed. (Obviously, it is closed only if the assumptions already contain a pair of **fórmulas assinadas** $\mathbb{T}\varphi$ and $\mathbb{F}\varphi$.)

From a **tableau** (open or closed) we can obtain a new, larger one by applying one of the rules of inference to a **fórmula assinada** φ in it. The rule will append one or more **fórmulas assinadas** to the end of any branch containing the occurrence of φ to which we apply the rule.

For instance, consider the assumption $\mathbb{T}\varphi \wedge \neg\varphi$. Here is the (open) **tableau** consisting of just that assumption:

1. $\mathbb{T}\varphi \wedge \neg\varphi$ Assumption

We obtain a new **tableau** from it by applying the $\wedge\mathbb{T}$ rule to the assumption. That rule allows us to add two new lines to the **tableau**, $\mathbb{T}\varphi$ and $\mathbb{T}\neg\varphi$:

1. $\mathbb{T}\varphi \wedge \neg\varphi$ Assumption
2. $\mathbb{T}\varphi$ $\wedge\mathbb{T} 1$
3. $\mathbb{T}\neg\varphi$ $\wedge\mathbb{T} 1$

When we write down **tableaux**, we record the rules we've applied on the right (p.ex., $\wedge\mathbb{T}1$ means that the **fórmula assinada** on that line is the result of applying the $\wedge\mathbb{T}$ rule to the **fórmula assinada** on line 1). This new **tableau** now contains additional **fórmulas assinadas**, but to only one ($\mathbb{T}\neg\varphi$) can we apply a rule (in this case, the $\neg\mathbb{T}$ rule). This results in the closed **tableau**

11.4. EXAMPLES OF TABLEAUX

1.	$\mathbb{T}\varphi \wedge \neg\varphi$	Assumption
2.	$\mathbb{T}\varphi$	$\wedge\mathbb{T} 1$
3.	$\mathbb{T}\neg\varphi$	$\wedge\mathbb{T} 1$
4.	$\mathbb{F}\varphi$	$\neg\mathbb{T} 3$
	$\otimes$	

-NoValue-

content/propositional-logic/./first-order-logic/tableaux/proving-things.tex

11.4 Examples of Tableaux

pl:tab:pro:
sec

Exemplo 11.4. Let's find a closed **tableau** for the **sentença** $(\varphi \wedge \psi) \rightarrow \varphi$.

We begin by writing the corresponding assumption at the top of the **tableau**.

1.	$\mathbb{F}(\varphi \wedge \psi) \rightarrow \varphi$	Assumption

There is only one assumption, so only one **fórmula assinada** to which we can apply a rule. (For every **fórmula assinada**, there is always at most one rule that can be applied: it's the rule for the corresponding sign and **operador principal** of the **sentença**.) In this case, this means, we must apply $\rightarrow\mathbb{F}$.

1.	$\mathbb{F}(\varphi \wedge \psi) \rightarrow \varphi \checkmark$	Assumption
2.	$\mathbb{T}\varphi \wedge \psi$	$\rightarrow\mathbb{F} 1$
3.	$\mathbb{F}\varphi$	$\rightarrow\mathbb{F} 1$

To keep track of which **fórmulas assinadas** we have applied their corresponding rules to, we write a checkmark next to the sentence. However, *only* write a checkmark if the rule has been applied to all open branches. Once a **fórmula assinada** has had the corresponding rule applied in every open branch, we will not have to return to it and apply the rule again. In this case, there is only one branch, so the rule only has to be applied once. (Note that checkmarks are only a convenience for constructing tableaux and are not officially part of the syntax of tableaux.)

There is one new **fórmula assinada** to which we can apply a rule: the $\mathbb{T}\varphi \wedge \psi$ on line 2. Applying the $\wedge\mathbb{T}$ rule results in:

1.	$\mathbb{F}(\varphi \wedge \psi) \rightarrow \varphi \checkmark$	Assumption
2.	$\mathbb{T}\varphi \wedge \psi \checkmark$	$\rightarrow\mathbb{F} 1$
3.	$\mathbb{F}\varphi$	$\rightarrow\mathbb{F} 1$
4.	$\mathbb{T}\varphi$	$\wedge\mathbb{T} 2$
5.	$\mathbb{T}\psi$	$\wedge\mathbb{T} 2$
	$\otimes$	

Since the branch now contains both $\mathbb{T}\varphi$ (on line 4) and $\mathbb{F}\varphi$ (on line 3), the branch is closed. Since it is the only branch, the **tableau** is closed. We have found a closed **tableau** for $(\varphi \wedge \psi) \rightarrow \varphi$.

Exemplo 11.5. Now let's find a closed **tableau** for $(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)$.

We begin with the corresponding assumption:

1. $\mathbb{F}(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)$ Assumption

The one **fórmula assinada** in this **tableau** has **operador principal** $\rightarrow$ and sign $\mathbb{F}$, so we apply the $\rightarrow\mathbb{F}$ rule to it to obtain:

1. $\mathbb{F}(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)$ ✓ Assumption
2. $\mathbb{T}\neg\varphi \vee \psi$ $\rightarrow\mathbb{F} 1$
3. $\mathbb{F}(\varphi \rightarrow \psi)$ $\rightarrow\mathbb{F} 1$

We now have a choice as to whether to apply $\vee\mathbb{T}$ to line 2 or $\rightarrow\mathbb{F}$ to line 3. It actually doesn't matter which order we pick, as long as each **fórmula assinada** has its corresponding rule applied in every branch. So let's pick the first one. The $\vee\mathbb{T}$ rule allows the **tableau** to branch, and the two conclusions of the rule will be the new **fórmulas assinadas** added to the two new branches. This results in:

1. $\mathbb{F}(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)$ ✓ Assumption
2. $\mathbb{T}\neg\varphi \vee \psi$ ✓ $\rightarrow\mathbb{F} 1$
3. $\mathbb{F}(\varphi \rightarrow \psi)$ $\rightarrow\mathbb{F} 1$
4. $\mathbb{T}\neg\varphi$ $\mathbb{T}\psi$ $\vee\mathbb{T} 2$

We have not applied the $\rightarrow\mathbb{F}$ rule to line 3 yet: let's do that now. To save time, we apply it to both branches. Recall that we write a checkmark next to a **fórmula assinada** only if we have applied the corresponding rule in every open branch. So it's a good idea to apply a rule at the end of every branch that contains the **fórmula assinada** the rule applies to. That way we won't have to return to that **fórmula assinada** lower down in the various branches.

1. $\mathbb{F}(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)$ ✓ Assumption
 2. $\mathbb{T}\neg\varphi \vee \psi$ ✓ $\rightarrow\mathbb{F} 1$
 3. $\mathbb{F}(\varphi \rightarrow \psi)$ ✓ $\rightarrow\mathbb{F} 1$
 4. $\mathbb{T}\neg\varphi$ $\mathbb{T}\psi$ $\vee\mathbb{T} 2$
 5. $\mathbb{T}\varphi$ $\mathbb{T}\varphi$ $\rightarrow\mathbb{F} 3$
 6. $\mathbb{F}\psi$ $\mathbb{F}\psi$ $\rightarrow\mathbb{F} 3$
- ⊗

The right branch is now closed. On the left branch, we can still apply the $\neg\mathbb{T}$ rule to line 4. This results in $\mathbb{F}\varphi$ and closes the left branch:

11.4. EXAMPLES OF TABLEAUX

1.	$\mathbb{F}(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi) \checkmark$	Assumption
2.	$\mathbb{T}\neg\varphi \vee \psi \checkmark$	$\rightarrow\mathbb{F} 1$
3.	$\mathbb{F}(\varphi \rightarrow \psi) \checkmark$	$\rightarrow\mathbb{F} 1$
$\swarrow \quad \searrow$		
4.	$\mathbb{T}\neg\varphi \quad \mathbb{T}\psi$	$\vee\mathbb{T} 2$
5.	$\mathbb{T}\varphi \quad \mathbb{T}\varphi$	$\rightarrow\mathbb{F} 3$
6.	$\mathbb{F}\psi \quad \mathbb{F}\psi$	$\rightarrow\mathbb{F} 3$
7.	$\mathbb{F}\varphi \quad \otimes$	$\neg\mathbb{T} 4$
	$\otimes$	

Exemplo 11.6. We can give **tableaux** for any number of **fórmulas assinadas** as assumptions. Often it is also necessary to apply more than one rule that allows branching; and in general a **tableau** can have any number of branches. For instance, consider a **tableau** for $\{\mathbb{T}\varphi \vee (\psi \wedge \chi), \mathbb{F}(\varphi \vee \psi) \wedge (\varphi \vee \chi)\}$. We start by applying the $\vee\mathbb{T}$ to the first assumption:

1.	$\mathbb{T}\varphi \vee (\psi \wedge \chi) \checkmark$	Assumption
2.	$\mathbb{F}(\varphi \vee \psi) \wedge (\varphi \vee \chi)$	Assumption
$\swarrow \quad \searrow$		
3.	$\mathbb{T}\varphi \quad \mathbb{T}\psi \wedge \chi$	$\vee\mathbb{T} 1$

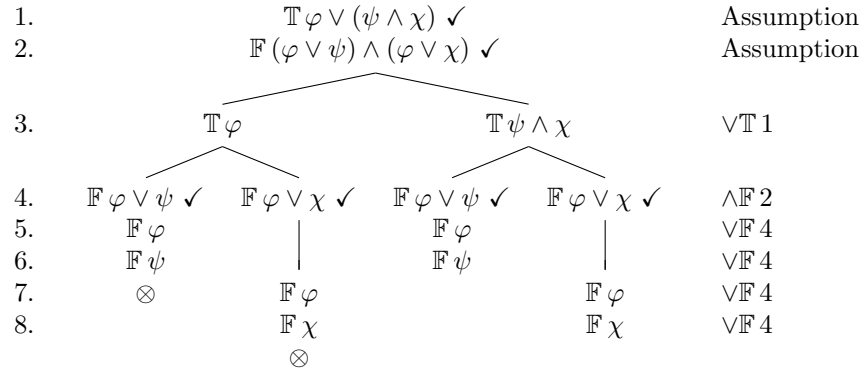
Now we can apply the $\wedge\mathbb{F}$ rule to line 2. We do this on both branches simultaneously, and can therefore check off line 2:

1.	$\mathbb{T}\varphi \vee (\psi \wedge \chi) \checkmark$	Assumption
2.	$\mathbb{F}(\varphi \vee \psi) \wedge (\varphi \vee \chi) \checkmark$	Assumption
$\swarrow \quad \searrow$		
3.	$\mathbb{T}\varphi \quad \mathbb{T}\psi \wedge \chi$	$\vee\mathbb{T} 1$
$\swarrow \quad \searrow \quad \swarrow \quad \searrow$		
4.	$\mathbb{F}\varphi \vee \psi \quad \mathbb{F}\varphi \vee \chi \quad \mathbb{F}\varphi \vee \psi \quad \mathbb{F}\varphi \vee \chi$	$\wedge\mathbb{F} 2$

Now we can apply $\vee\mathbb{F}$ to all the branches containing $\varphi \vee \psi$:

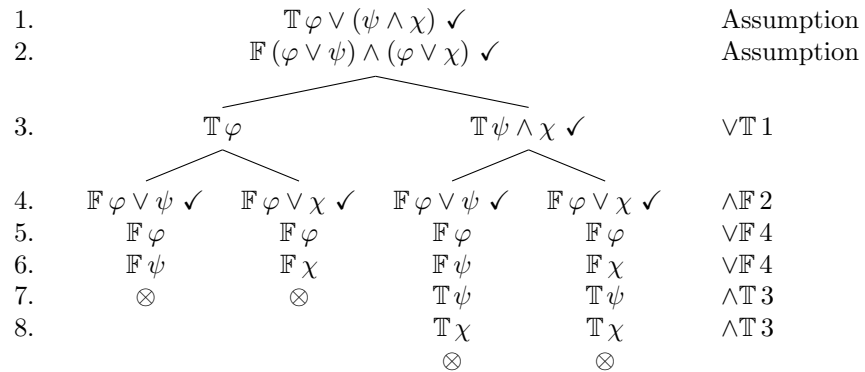
1.	$\mathbb{T}\varphi \vee (\psi \wedge \chi) \checkmark$	Assumption
2.	$\mathbb{F}(\varphi \vee \psi) \wedge (\varphi \vee \chi) \checkmark$	Assumption
$\swarrow \quad \searrow$		
3.	$\mathbb{T}\varphi \quad \mathbb{T}\psi \wedge \chi$	$\vee\mathbb{T} 1$
$\swarrow \quad \searrow \quad \swarrow \quad \searrow$		
4.	$\mathbb{F}\varphi \vee \psi \checkmark \quad \mathbb{F}\varphi \vee \chi \quad \mathbb{F}\varphi \vee \psi \checkmark \quad \mathbb{F}\varphi \vee \chi$	$\wedge\mathbb{F} 2$
5.	$\mathbb{F}\varphi \quad \mathbb{F}\varphi$	$\vee\mathbb{F} 4$
6.	$\mathbb{F}\psi \quad \mathbb{F}\psi$	$\vee\mathbb{F} 4$
	$\otimes$	

The leftmost branch is now closed. Let's now apply $\vee\mathbb{F}$ to $\varphi \vee \chi$:



Note that we moved the result of applying $\vee \mathbb{F}$ a second time below for clarity. In this instance it would not have been needed, since the justifications would have been the same.

Two branches remain open, and $\mathbb{T} \psi \wedge \chi$ on line 3 remains unchecked. We apply $\wedge \mathbb{T}$ to it to obtain a closed **tableau**:



For comparison, here's a closed **tableau** for the same set of assumptions in which the rules are applied in a different order:

11.4. EXAMPLES OF TABLEAUX

1.	$\mathbb{T} \varphi \vee (\psi \wedge \chi) \checkmark$	Assumption
2.	$\mathbb{F} (\varphi \vee \psi) \wedge (\varphi \vee \chi) \checkmark$	Assumption
3.	$\mathbb{F} \varphi \vee \psi \checkmark$	$\wedge \mathbb{F} 2$
4.	$\mathbb{F} \varphi$	$\vee \mathbb{F} 3$
5.	$\mathbb{F} \psi$	$\vee \mathbb{F} 3$
6.	$\mathbb{T} \varphi$	$\vee \mathbb{T} 1$
7.	$\mathbb{T} \psi \wedge \chi \checkmark$	$\wedge \mathbb{T} 6$
8.	$\mathbb{T} \psi$	$\wedge \mathbb{T} 6$
	$\mathbb{T} \chi$	
	$\otimes$	

Problema 11.1. Give closed **tableaux** of the following:

1. $\mathbb{T} \varphi \wedge (\psi \wedge \chi), \mathbb{F} (\varphi \wedge \psi) \wedge \chi$.
2. $\mathbb{T} \varphi \vee (\psi \vee \chi), \mathbb{F} (\varphi \vee \psi) \vee \chi$.
3. $\mathbb{T} \varphi \rightarrow (\psi \rightarrow \chi), \mathbb{F} \psi \rightarrow (\varphi \rightarrow \chi)$.
4. $\mathbb{T} \varphi, \mathbb{F} \neg \neg \varphi$.

Problema 11.2. Give closed **tableaux** of the following:

1. $\mathbb{T} (\varphi \vee \psi) \rightarrow \chi, \mathbb{F} \varphi \rightarrow \chi$.
2. $\mathbb{T} (\varphi \rightarrow \chi) \wedge (\psi \rightarrow \chi), \mathbb{F} (\varphi \vee \psi) \rightarrow \chi$.
3. $\mathbb{F} \neg (\varphi \wedge \neg \varphi)$.
4. $\mathbb{T} \psi \rightarrow \varphi, \mathbb{F} \neg \varphi \rightarrow \neg \psi$.
5. $\mathbb{F} (\varphi \rightarrow \neg \varphi) \rightarrow \neg \varphi$.
6. $\mathbb{F} \neg (\varphi \rightarrow \psi) \rightarrow \neg \psi$.
7. $\mathbb{T} \varphi \rightarrow \chi, \mathbb{F} \neg (\varphi \wedge \neg \chi)$.
8. $\mathbb{T} \varphi \wedge \neg \chi, \mathbb{F} \neg (\varphi \rightarrow \chi)$.
9. $\mathbb{T} \varphi \vee \psi, \neg \psi, \mathbb{F} \varphi$.
10. $\mathbb{T} \neg \varphi \vee \neg \psi, \mathbb{F} \neg (\varphi \wedge \psi)$.
11. $\mathbb{F} (\neg \varphi \wedge \neg \psi) \rightarrow \neg (\varphi \vee \psi)$.
12. $\mathbb{F} \neg (\varphi \vee \psi) \rightarrow (\neg \varphi \wedge \neg \psi)$.

Problema 11.3. Give closed **tableaux** of the following:

1. $\mathbb{T} \neg (\varphi \rightarrow \psi), \mathbb{F} \varphi$.

2. $\mathbb{T} \neg(\varphi \wedge \psi), \mathbb{F} \neg\varphi \vee \neg\psi$.
3. $\mathbb{T} \varphi \rightarrow \psi, \mathbb{F} \neg\varphi \vee \psi$.
4. $\mathbb{F} \neg\neg\varphi \rightarrow \varphi$.
5. $\mathbb{T} \varphi \rightarrow \psi, \mathbb{T} \neg\varphi \rightarrow \psi, \mathbb{F} \psi$.
6. $\mathbb{T} (\varphi \wedge \psi) \rightarrow \chi, \mathbb{F} (\varphi \rightarrow \chi) \vee (\psi \rightarrow \chi)$.
7. $\mathbb{T} (\varphi \rightarrow \psi) \rightarrow \varphi, \mathbb{F} \varphi$.
8. $\mathbb{F} (\varphi \rightarrow \psi) \vee (\psi \rightarrow \chi)$.

-NoValue-

[content/propositional-logic/.../first-order-logic/tableaux/proof-theoretic-notions.t](#)

11.5 Proof-Theoretic Notions

pl:tab:ptn:
sec

This section collects the definitions of the provability relation and consistency for tableaux.

explanation Just as we've defined a number of important semantic notions (validity, entailment, satisfiability), we now define corresponding *proof-theoretic notions*. These are not defined by appeal to satisfaction of **sentenças** in **estruturas**, but by appeal to the existence of certain closed **tableaux**. It was an important discovery that these notions coincide. That they do is the content of the *soundness* and *completeness theorems*.

Definição 11.7 (Theorems). A **sentença** φ is a *theorem* if there is a closed **tableau** for $\mathbb{F} \varphi$. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

Definição 11.8 (Derivabilidade). A **sentença** φ is *derivável* from a set of **sentenças** Γ , $\Gamma \vdash \varphi$ iff there is a finite set $\{\psi_1, \dots, \psi_n\} \subseteq \Gamma$ and a closed **tableau** for the set

$$\{\mathbb{F} \varphi, \mathbb{T} \psi_1, \dots, \mathbb{T} \psi_n\}.$$

If φ is not *derivável* from Γ we write $\Gamma \nvdash \varphi$.

Definição 11.9 (Consistency). A set of **sentenças** Γ is *inconsistent* iff there is a finite set $\{\psi_1, \dots, \psi_n\} \subseteq \Gamma$ and a closed **tableau** for the set

$$\{\mathbb{T} \psi_1, \dots, \mathbb{T} \psi_n\}.$$

If Γ is not inconsistent, we say it is *consistent*.

11.5. PROOF-THEORETIC NOTIONS

*pl:tab;ptn:
prop:reflexivity*

Proposição 11.10 (Reflexivity). *If $\varphi \in \Gamma$, then $\Gamma \vdash \varphi$.*

Prova. If $\varphi \in \Gamma$, $\{\varphi\}$ is a finite subset of Γ and the **tableau**

1. $\mathbb{F}\varphi$ Assumption
 2. $\mathbb{T}\varphi$ Assumption
- $\otimes$

is closed. $\square$

*pl:tab;ptn:
prop:monotonicity*

Proposição 11.11 (Monotonicity). *If $\Gamma \subseteq \Delta$ and $\Gamma \vdash \varphi$, then $\Delta \vdash \varphi$.*

Prova. Any finite subset of Γ is also a finite subset of Δ . $\square$

*pl:tab;ptn:
prop:transitivity*

Proposição 11.12 (Transitivity). *If $\Gamma \vdash \varphi$ and $\{\varphi\} \cup \Delta \vdash \psi$, then $\Gamma \cup \Delta \vdash \psi$.*

Prova. If $\{\varphi\} \cup \Delta \vdash \psi$, then there is a finite subset $\Delta_0 = \{\chi_1, \dots, \chi_n\} \subseteq \Delta$ such that

$$\{\mathbb{F}\psi, \mathbb{T}\varphi, \mathbb{T}\chi_1, \dots, \mathbb{T}\chi_n\}$$

has a closed **tableau**. If $\Gamma \vdash \varphi$ then there are $\theta_1, \dots, \theta_m$ such that

$$\{\mathbb{F}\varphi, \mathbb{T}\theta_1, \dots, \mathbb{T}\theta_m\}$$

has a closed **tableau**.

Now consider the **tableau** with assumptions

$$\mathbb{F}\psi, \mathbb{T}\chi_1, \dots, \mathbb{T}\chi_n, \mathbb{T}\theta_1, \dots, \mathbb{T}\theta_m.$$

Apply the Cut rule on φ . This generates two branches, one has $\mathbb{T}\varphi$ in it, the other $\mathbb{F}\varphi$. Thus, on the one branch, all of

$$\{\mathbb{F}\psi, \mathbb{T}\varphi, \mathbb{T}\chi_1, \dots, \mathbb{T}\chi_n\}$$

are available. Since there is a closed **tableau** for these assumptions, we can attach it to that branch; every branch through $\mathbb{T}\varphi$ closes. On the other branch, all of

$$\{\mathbb{F}\varphi, \mathbb{T}\theta_1, \dots, \mathbb{T}\theta_m\}$$

are available, so we can also complete the other side to obtain a closed **tableau**. This shows $\Gamma \cup \Delta \vdash \psi$. $\square$

Note that this means that in particular if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. It follows also that if $\varphi_1, \dots, \varphi_n \vdash \psi$ and $\Gamma \vdash \varphi_i$ for each i , then $\Gamma \vdash \psi$.

*pl:tab;ptn:
prop:incons*

Proposição 11.13. *Γ is inconsistent iff $\Gamma \vdash \varphi$ for every **sentença** φ .*

Prova. Exercise. □

Problema 11.4. Prove [Proposição 11.13](#)

Proposição 11.14 (Compactness).

pl:tab:ptn:
prop:proves-compact

1. If $\Gamma \vdash \varphi$ then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vdash \varphi$.
2. If every finite subset of Γ is consistent, then Γ is consistent.

Prova. 1. If $\Gamma \vdash \varphi$, then there is a finite subset $\Gamma_0 = \{\psi_1, \dots, \psi_n\}$ and a closed [tableau](#) for

$$\{\mathbb{F} \varphi, \mathbb{T} \psi_1, \dots, \mathbb{T} \psi_n\}$$

This [tableau](#) also shows $\Gamma_0 \vdash \varphi$.

2. If Γ is inconsistent, then for some finite subset $\Gamma_0 = \{\psi_1, \dots, \psi_n\}$ there is a closed [tableau](#) for

$$\{\mathbb{T} \psi_1, \dots, \mathbb{T} \psi_n\}$$

This closed [tableau](#) shows that Γ_0 is inconsistent. □

-NoValue-

[content/propositional-logic/./first-order-logic/tableaux/provability-consistency.t](#)

11.6 Derivabilidade and Consistency

We will now establish a number of properties of the [derivabilidade](#) relation. They are independently interesting, but each will play a role in the proof of the completeness theorem.

pl:tab:prv:
sec

Proposição 11.15. *If $\Gamma \vdash \varphi$ and $\Gamma \cup \{\varphi\}$ is inconsistent, then Γ is inconsistent.*

pl:tab:prv:
prop:provability-contr

Prova. There are finite $\Gamma_0 = \{\psi_1, \dots, \psi_n\}$ and $\Gamma_1 = \{\chi_1, \dots, \chi_m\} \subseteq \Gamma$ such that

$$\begin{aligned} &\{\mathbb{F} \varphi, \mathbb{T} \psi_1, \dots, \mathbb{T} \psi_n\} \\ &\{\mathbb{T} \varphi, \mathbb{T} \chi_1, \dots, \mathbb{T} \chi_m\} \end{aligned}$$

have closed [tableaux](#). Using the Cut rule on φ we can combine these into a single closed [tableau](#) that shows $\Gamma_0 \cup \Gamma_1$ is inconsistent. Since $\Gamma_0 \subseteq \Gamma$ and $\Gamma_1 \subseteq \Gamma$, $\Gamma_0 \cup \Gamma_1 \subseteq \Gamma$, hence Γ is inconsistent. □

Proposição 11.16. *$\Gamma \vdash \varphi$ iff $\Gamma \cup \{\neg \varphi\}$ is inconsistent.*

pl:tab:prv:
prop:prov-incons

11.6. DERIVABILIDADE AND CONSISTENCY

Prova. First suppose $\Gamma \vdash \varphi$, O.s., there is a closed **tableau** for

$$\{\mathbb{F} \varphi, \mathbb{T} \psi_1, \dots, \mathbb{T} \psi_n\}$$

Using the $\neg\mathbb{T}$ rule, this can be turned into a closed **tableau** for

$$\{\mathbb{T} \neg\varphi, \mathbb{T} \psi_1, \dots, \mathbb{T} \psi_n\}.$$

On the other hand, if there is a closed **tableau** for the latter, we can turn it into a closed **tableau** of the former by removing every formula that results from $\neg\mathbb{T}$ applied to the first assumption $\mathbb{T} \neg\varphi$ as well as that assumption, and adding the assumption $\mathbb{F} \varphi$. For if a branch was closed before because it contained the conclusion of $\neg\mathbb{T}$ applied to $\mathbb{T} \neg\varphi$, O.s., $\mathbb{F} \varphi$, the corresponding branch in the new **tableau** is also closed. If a branch in the old tableau was closed because it contained the assumption $\mathbb{T} \neg\varphi$ as well as $\mathbb{F} \neg\varphi$ we can turn it into a closed branch by applying $\neg\mathbb{F}$ to $\mathbb{F} \neg\varphi$ to obtain $\mathbb{T} \varphi$. This closes the branch since we added $\mathbb{F} \varphi$ as an assumption. $\square$

Problema 11.5. Prove that $\Gamma \vdash \neg\varphi$ iff $\Gamma \cup \{\varphi\}$ is inconsistent.

pl:tab:prv: prop:explicit-inc **Proposição 11.17.** *If $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$, then Γ is inconsistent.*

Prova. Suppose $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$. Then there are $\psi_1, \dots, \psi_n \in \Gamma$ such that

$$\{\mathbb{F} \varphi, \mathbb{T} \psi_1, \dots, \mathbb{T} \psi_n\}$$

has a closed tableau. Replace the assumption $\mathbb{F} \varphi$ by $\mathbb{T} \neg\varphi$, and insert the conclusion of $\neg\mathbb{T}$ applied to $\mathbb{F} \varphi$ after the assumptions. Any **sentença** in the **tableau** justified by appeal to line 1 in the old **tableau** is now justified by appeal to line $n+1$. So if the old **tableau** was closed, the new one is. It shows that Γ is inconsistent, since all assumptions are in Γ . $\square$

pl:tab:prv: prop:provability-exhaustive **Proposição 11.18.** *If $\Gamma \cup \{\varphi\}$ and $\Gamma \cup \{\neg\varphi\}$ are both inconsistent, then Γ is inconsistent.*

Prova. If there are $\psi_1, \dots, \psi_n \in \Gamma$ and $\chi_1, \dots, \chi_m \in \Gamma$ such that

$$\begin{aligned} &\{\mathbb{T} \varphi, \mathbb{T} \psi_1, \dots, \mathbb{T} \psi_n\} \text{ and} \\ &\{\mathbb{T} \neg\varphi, \mathbb{T} \chi_1, \dots, \mathbb{T} \chi_m\} \end{aligned}$$

both have closed **tableaux**, we can construct a single, combined **tableau** that shows that Γ is inconsistent by using as assumptions $\mathbb{T} \psi_1, \dots, \mathbb{T} \psi_n$ together with $\mathbb{T} \chi_1, \dots, \mathbb{T} \chi_m$, followed by an application of the Cut rule. This yields two branches, one starting with $\mathbb{T} \varphi$, the other with $\mathbb{F} \varphi$.

On the left left side, add the part of the first **tableau** below its assumptions. Here, every rule application is still correct, since each of the assumptions of the first **tableau**, including $\mathbb{T} \varphi$, is available. Thus, every branch below $\mathbb{T} \varphi$ closes.

On the right side, add the part of the second **tableau** below its assumption, with the results of any applications of $\neg\mathbb{T}$ to $\mathbb{T}\neg\varphi$ removed. The conclusion of $\neg\mathbb{T}$ to $\mathbb{T}\neg\varphi$ is $\mathbb{F}\varphi$, which is nevertheless available, as it is the conclusion of the Cut rule on the right side of the combined **tableau**.

If a branch in the second tableau was closed because it contained the assumption $\mathbb{T}\neg\varphi$ (which no longer appears as an assumption in the combined **tableau**) as well as $\mathbb{F}\neg\varphi$, we can applying $\neg\mathbb{F}$ to $\mathbb{F}\neg\varphi$ to obtain $\mathbb{T}\varphi$. Now the corresponding branch in the combined **tableau** also closes, because it contains the right-hand conclusion of the Cut rule, $\mathbb{F}\varphi$. If a branch in the second **tableau** closed for any other reason, the corresponding branch in the combined **tableau** also closes, since any **fórmulas assinadas** other than $\mathbb{T}\neg\varphi$ occurring on the branch in the old, second **tableau** also occur on the corresponding branch in the combined **tableau**. $\square$

-NoValue-

content/propositional-logic/./first-order-logic/tableaux/provability-propositional

11.7 Derivabilidade and the Propositional Connectives

explanation We establish that the **derivabilidade** relation $\vdash$ of tableaux is strong enough to establish some basic facts involving the propositional connectives, such as that $\varphi \wedge \psi \vdash \varphi$ and $\varphi, \varphi \rightarrow \psi \vdash \psi$ (modus ponens). These facts are needed for the proof of the completeness theorem.

pl:tab:ppr:
sec

Proposição 11.19.

1. Both $\varphi \wedge \psi \vdash \varphi$ and $\varphi \wedge \psi \vdash \psi$.
2. $\varphi, \psi \vdash \varphi \wedge \psi$.

pl:tab:ppr:
prop:provability-land
pl:tab:ppr:
prop:provability-land-left
pl:tab:ppr:
prop:provability-land-right

Prova. 1. Both $\{\mathbb{F}\varphi, \mathbb{T}\varphi \wedge \psi\}$ and $\{\mathbb{F}\psi, \mathbb{T}\varphi \wedge \psi\}$ have closed **tableaux**

1.	$\mathbb{F}\varphi$	Assumption
2.	$\mathbb{T}\varphi \wedge \psi$	Assumption
3.	$\mathbb{T}\varphi$	$\wedge\mathbb{T} 2$
4.	$\mathbb{T}\psi$	$\wedge\mathbb{T} 2$
	$\otimes$	

1.	$\mathbb{F}\psi$	Assumption
2.	$\mathbb{T}\varphi \wedge \psi$	Assumption
3.	$\mathbb{T}\varphi$	$\wedge\mathbb{T} 2$
4.	$\mathbb{T}\psi$	$\wedge\mathbb{T} 2$
	$\otimes$	

11.7. DERIVABILIDADE AND THE PROPOSITIONAL CONNECTIVES

2. Here is a closed **tableau** for $\{\mathbb{T}\varphi, \mathbb{T}\psi, \mathbb{F}\varphi \wedge \psi\}$:

1.	$\mathbb{F}\varphi \wedge \psi$	Assumption
2.	$\mathbb{T}\varphi$	Assumption
3.	$\mathbb{T}\psi$	Assumption
4.	$\mathbb{F}\varphi$ $\mathbb{F}\psi$	$\wedge\mathbb{F}1$
	$\otimes$ $\otimes$	

pl:tab:ppr:
prop:provability-lor

Proposição 11.20.

1. $\{\varphi \vee \psi, \neg\varphi, \neg\psi\}$ is inconsistent.
2. Both $\varphi \vdash \varphi \vee \psi$ and $\psi \vdash \varphi \vee \psi$.

Prova. 1. We give a closed **tableau** of $\{\mathbb{T}\varphi \vee \psi, \mathbb{T}\neg\varphi, \mathbb{T}\neg\psi\}$:

1.	$\mathbb{T}\varphi \vee \psi$	Assumption
2.	$\mathbb{T}\neg\varphi$	Assumption
3.	$\mathbb{T}\neg\psi$	Assumption
4.	$\mathbb{F}\varphi$	$\neg\mathbb{T}2$
5.	$\mathbb{F}\psi$	$\neg\mathbb{T}3$
6.	$\mathbb{T}\varphi$ $\mathbb{T}\psi$	$\vee\mathbb{T}1$
	$\otimes$ $\otimes$	

2. Both $\{\mathbb{F}\varphi \vee \psi, \mathbb{T}\varphi\}$ and $\{\mathbb{F}\varphi \vee \psi, \mathbb{T}\psi\}$ have closed **tableaux**:

1.	$\mathbb{F}\varphi \vee \psi$	Assumption
2.	$\mathbb{T}\varphi$	Assumption
3.	$\mathbb{F}\varphi$	$\vee\mathbb{F}1$
4.	$\mathbb{F}\psi$	$\vee\mathbb{F}1$
	$\otimes$	

1.	$\mathbb{F}\varphi \vee \psi$	Assumption
2.	$\mathbb{T}\psi$	Assumption
3.	$\mathbb{F}\varphi$	$\vee\mathbb{F}1$
4.	$\mathbb{F}\psi$	$\vee\mathbb{F}1$
	$\otimes$	

pl:tab:ppr:
prop:provability-lif

Proposição 11.21.

1. $\varphi, \varphi \rightarrow \psi \vdash \psi$.

2. Both $\neg\varphi \vdash \varphi \rightarrow \psi$ and $\psi \vdash \varphi \rightarrow \psi$.

pl:tab:ppr:
prop:provability-lif-left
pl:tab:ppr:
prop:provability-lif-right

Prova. 1. $\{\mathbb{F}\psi, \mathbb{T}\varphi \rightarrow \psi, \mathbb{T}\varphi\}$ has a closed **tableau**:

1.	$\mathbb{F}\psi$	Assumption
2.	$\mathbb{T}\varphi \rightarrow \psi$	Assumption
3.	$\mathbb{T}\varphi$	Assumption
4.	$\mathbb{F}\varphi$ $\mathbb{T}\psi$	$\rightarrow\mathbb{T}2$
	$\otimes$ $\otimes$	

2. Both $\{\mathbb{F}\varphi \rightarrow \psi, \mathbb{T}\neg\varphi\}$ and $\{\mathbb{F}\varphi \rightarrow \psi, \mathbb{T}\psi\}$ have closed **tableaux**:

1.	$\mathbb{F}\varphi \rightarrow \psi$	Assumption
2.	$\mathbb{T}\neg\varphi$	Assumption
3.	$\mathbb{T}\varphi$	$\rightarrow\mathbb{F}1$
4.	$\mathbb{F}\psi$	$\rightarrow\mathbb{F}1$
5.	$\mathbb{F}\varphi$	$\neg\mathbb{T}2$
	$\otimes$	

1.	$\mathbb{F}\varphi \rightarrow \psi$	Assumption
2.	$\mathbb{T}\psi$	Assumption
3.	$\mathbb{T}\varphi$	$\rightarrow\mathbb{F}1$
4.	$\mathbb{F}\psi$	$\rightarrow\mathbb{F}1$
	$\otimes$	

-NoValue-

`content/propositional-logic/./first-order-logic/tableaux/soundness.tex`

11.8 Soundness

explanation A **derivação** system, such as tableaux, is *sound* if it cannot **deriva** things that do not actually hold. Soundness is thus a kind of guaranteed safety property for **derivação** systems. Depending on which proof theoretic property is in question, we would like to know for instance, that *pl:tab:sou:sec*

1. every **derivável** φ is a tautology;
2. if a **sentença** is **derivável** from some others, it is also a consequence of them;

11.8. SOUNDNESS

3. if a set of *sentenças* is inconsistent, it is unsatisfiable.

These are important properties of a *derivação* system. If any of them do not hold, the *derivação* system is deficient—it would *deriva* too much. Consequently, establishing the soundness of a *derivação* system is of the utmost importance.

Because all these proof-theoretic properties are defined via closed *tableaux* of some kind or other, proving (1)–(3) above requires proving something about the semantic properties of closed *tableaux*. We will first define what it means for a *fórmula assinada* to be satisfied in a structure, and then show that if a *tableau* is closed, no structure satisfies all its assumptions. (1)–(3) then follow as corollaries from this result.

Definição 11.22. A *valoração* $\mathbf{v}$ *satisfies* a *fórmula assinada* $\mathbb{T}\varphi$ iff $\mathbf{v} \models \varphi$, and it satisfies $\mathbb{F}\varphi$ iff $\mathbf{v} \not\models \varphi$. $\mathbf{v}$ satisfies a set of *fórmulas assinadas* Γ iff it satisfies every $S\varphi \in \Gamma$. Γ is *satisfiable* if there is a *valoração* that satisfies it, and *unsatisfiable* otherwise.

Teorema 11.23 (Soundness). *If Γ has a closed *tableau*, Γ is unsatisfiable.*

Prova. Let's call a branch of a *tableau* *satisfiable* iff the set of *fórmulas assinadas* on it is satisfiable, and let's call a *tableau* *satisfiable* if it contains at least one satisfiable branch.

We show the following: Extending a satisfiable *tableau* by one of the rules of inference always results in a satisfiable *tableau*. This will prove the theorem: any closed *tableau* results by applying rules of inference to the *tableau* consisting only of assumptions from Γ . So if Γ were satisfiable, any *tableau* for it would be satisfiable. A closed *tableau*, however, is clearly not satisfiable: every branch contains both $\mathbb{T}\varphi$ and $\mathbb{F}\varphi$, and no structure can both satisfy and not satisfy φ .

Suppose we have a satisfiable *tableau*, O.s., a *tableau* with at least one satisfiable branch. Applying a rule of inference either adds *fórmulas assinadas* to a branch, or splits a branch in two. If the *tableau* has a satisfiable branch which is not extended by the rule application in question, it remains a satisfiable branch in the extended *tableau*, so the extended *tableau* is satisfiable. So we only have to consider the case where a rule is applied to a satisfiable branch.

Let Γ be the set of *fórmulas assinadas* on that branch, and let $S\varphi \in \Gamma$ be the *fórmula assinada* to which the rule is applied. If the rule does not result in a split branch, we have to show that the extended branch, O.s., Γ together with the conclusions of the rule, is still satisfiable. If the rule results in a split branch, we have to show that at least one of the two resulting branches is satisfiable.

First, we consider the possible inferences that do not result in a split branch.

1. The branch is expanded by applying $\neg\mathbb{T}$ to $\mathbb{T}\neg\psi \in \Gamma$. Then the extended branch contains the *fórmulas assinadas* $\Gamma \cup \{\mathbb{F}\psi\}$. Suppose $\mathbf{v} \models \Gamma$. In particular, $\mathbf{v} \models \neg\psi$. Thus, $\mathbf{v} \not\models \psi$, O.s., $\mathbf{v}$ satisfies $\mathbb{F}\psi$.

2. The branch is expanded by applying $\neg\mathbb{F}$ to $\mathbb{F}\neg\psi \in \Gamma$: Exercise.
3. The branch is expanded by applying $\wedge\mathbb{T}$ to $\mathbb{T}\psi \wedge \chi \in \Gamma$, which results in two new **fórmulas assinadas** on the branch: $\mathbb{T}\psi$ and $\mathbb{T}\chi$. Suppose $\mathfrak{v} \models \Gamma$, in particular $\mathfrak{v} \models \psi \wedge \chi$. Then $\mathfrak{v} \models \psi$ and $\mathfrak{v} \models \chi$. This means that $\mathfrak{v}$ satisfies both $\mathbb{T}\psi$ and $\mathbb{T}\chi$.
4. The branch is expanded by applying $\vee\mathbb{F}$ to $\mathbb{F}\psi \vee \chi \in \Gamma$: Exercise.
5. The branch is expanded by applying $\rightarrow\mathbb{F}$ to $\mathbb{F}\psi \rightarrow \chi \in \Gamma$: This results in two new **fórmulas assinadas** on the branch: $\mathbb{T}\psi$ and $\mathbb{F}\chi$. Suppose $\mathfrak{v} \models \Gamma$, in particular $\mathfrak{v} \models \psi \rightarrow \chi$. Then $\mathfrak{v} \models \psi$ and $\mathfrak{v} \models \chi$. This means that $\mathfrak{v}$ satisfies both $\mathbb{T}\psi$ and $\mathbb{F}\chi$.

Now let's consider the possible inferences that result in a split branch.

1. The branch is expanded by applying $\wedge\mathbb{F}$ to $\mathbb{F}\psi \wedge \chi \in \Gamma$, which results in two branches, a left one continuing through $\mathbb{F}\psi$ and a right one through $\mathbb{F}\chi$. Suppose $\mathfrak{v} \models \Gamma$, in particular $\mathfrak{v} \models \psi \wedge \chi$. Then $\mathfrak{v} \models \psi$ or $\mathfrak{v} \models \chi$. In the former case, $\mathfrak{v}$ satisfies $\mathbb{F}\psi$, O.s., $\mathfrak{v}$ satisfies the formulas on the left branch. In the latter, $\mathfrak{v}$ satisfies $\mathbb{F}\chi$, O.s., $\mathfrak{v}$ satisfies the formulas on the right branch.
2. The branch is expanded by applying $\vee\mathbb{T}$ to $\mathbb{T}\psi \vee \chi \in \Gamma$: Exercise.
3. The branch is expanded by applying $\rightarrow\mathbb{T}$ to $\mathbb{T}\psi \rightarrow \chi \in \Gamma$: Exercise.
4. The branch is expanded by Cut: This results in two branches, one containing $\mathbb{T}\psi$, the other containing $\mathbb{F}\psi$. Since $\mathfrak{v} \models \Gamma$ and either $\mathfrak{v} \models \psi$ or $\mathfrak{v} \models \neg\psi$, $\mathfrak{v}$ satisfies either the left or the right branch. $\square$

Problema 11.6. Complete the proof of **Teorema 11.23**.

Corolário 11.24. *If $\vdash \varphi$ then φ is a tautology.*

*pl:tab:sou:
cor:weak-soundness*

Corolário 11.25. *If $\Gamma \vdash \varphi$ then $\Gamma \models \varphi$.*

*pl:tab:sou:
cor:entailment-soundness*

Prova. If $\Gamma \vdash \varphi$ then for some $\psi_1, \dots, \psi_n \in \Gamma$, $\{\mathbb{F}\varphi, \mathbb{T}\psi_1, \dots, \mathbb{T}\psi_n\}$ has a closed **tableau**. By **Teorema 11.23**, every **valoração** $\mathfrak{v}$ either makes some ψ_i false or makes φ true. Hence, if $\mathfrak{v} \models \Gamma$ then also $\mathfrak{v} \models \varphi$. $\square$

Corolário 11.26. *If Γ is satisfiable, then it is consistent.*

*pl:tab:sou:
cor:consistency-soundness*

Prova. We prove the contrapositive. Suppose that Γ is not consistent. Then there are $\psi_1, \dots, \psi_n \in \Gamma$ and a closed **tableau** for $\{\mathbb{T}\psi_1, \dots, \mathbb{T}\psi_n\}$. By **Teorema 11.23**, there is no $\mathfrak{v}$ such that $\mathfrak{v} \models \psi_i$ for all $i = 1, \dots, n$. But then Γ is not satisfiable. $\square$

Capítulo 12

Axiomatic Derivações

No effort has been made yet to ensure that the material in this chapter respects various tags indicating which connectives and quantifiers are primitive or defined: all are assumed to be primitive, except $\leftrightarrow$ which is assumed to be defined. If the FOL tag is true, we produce a version with quantifiers, otherwise without.

-NoValue-

`content/propositional-logic/.../first-order-logic/axiomatic-deduction/rules-and-proofs.tex`

12.1 Rules and Derivações

pl:axd:rul:
sec

Axiomatic **derivações** are perhaps the simplest **derivação** system for logic. A **derivação** is just a sequence of **fórmulas**. To count as a **derivação**, every **fórmula** in the sequence must either be an instance of an axiom, or must follow from one or more **fórmulas** that precede it in the sequence by a rule of inference. A **derivação** **deriva** its last **fórmula**.

explanation

Definição 12.1 (Derivabilidade). If Γ is a set of **fórmulas** of $\mathcal{L}$ then a **derivação** from Γ is a finite sequence $\varphi_1, \dots, \varphi_n$ of **fórmulas** where for each $i \leq n$ one of the following holds:

1. $\varphi_i \in \Gamma$; or
2. φ_i is an axiom; or
3. φ_i follows from some φ_j (and φ_k) with $j < i$ (and $k < i$) by a rule of inference.

What counts as a correct **derivação** depends on which inference rules we allow (and of course what we take to be axioms). And an inference rule is

an if-then statement that tells us that, under certain conditions, a step A_i in a **derivação** is a correct inference step.

Definição 12.2 (Rule of inference). A *rule of inference* gives a sufficient condition for what counts as a correct inference step in a **derivação** from Γ .

For instance, since any one-element sequence φ with $\varphi \in \Gamma$ trivially counts as a **derivação**, the following might be a very simple rule of inference:

If $\varphi \in \Gamma$, then φ is always a correct inference step in any **derivação** from Γ .

Similarly, if φ is one of the axioms, then φ by itself is a **derivação**, and so this is also a rule of inference:

If φ is an axiom, then φ is a correct inference step.

It gets more interesting if the rule of inference appeals to **fórmulas** that appear before the step considered. The following rule is called *modus ponens*:

If $\psi \rightarrow \varphi$ and ψ occur higher up in the **derivação**, then φ is a correct inference step.

If this is the only rule of inference, then our definition of **derivação** above amounts to this: $\varphi_1, \dots, \varphi_n$ is a **derivação** iff for each $i \leq n$ one of the following holds:

1. $\varphi_i \in \Gamma$; or
2. φ_i is an axiom; or
3. for some $j < i$, φ_j is $\psi \rightarrow \varphi_i$, and for some $k < i$, φ_k is ψ .

The last clause says that φ_i follows from φ_j ($\psi \rightarrow \varphi_i$) and φ_k (ψ) by modus ponens. If we can go from 1 to n , and each time we find a **fórmula** φ_i that is either in Γ , an axiom, or which a rule of inference tells us that it is a correct inference step, then the entire sequence counts as a correct **derivação**.

Definição 12.3 (Derivabilidade). A **fórmula** φ is *derivável* from Γ , written $\Gamma \vdash \varphi$, if there is a **derivação** from Γ ending in φ .

Definição 12.4 (Theorems). A **fórmula** φ is a *theorem* if there is a **derivação** of φ from the empty set. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

-NoValue-

content/propositional-logic/.../first-order-logic/axiomatic-deduction/axioms-rules-p

12.2. AXIOMS AND RULES FOR THE PROPOSITIONAL CONNECTIVES

12.2 Axioms and Rules for the Propositional Connectives

pl:axd:prp:
sec

Definição 12.5 (Axioms). The set of Ax_0 of *axioms* for the propositional connectives comprises all *fórmulas* of the following forms:

pl:axd:prp:	$(\varphi \wedge \psi) \rightarrow \varphi$	(12.1)
ax:land1 pl:axd:prp:	$(\varphi \wedge \psi) \rightarrow \psi$	(12.2)
ax:land2 pl:axd:prp:	$\varphi \rightarrow (\psi \rightarrow (\varphi \wedge \psi))$	(12.3)
ax:land3 pl:axd:prp:	$\varphi \rightarrow (\varphi \vee \psi)$	(12.4)
ax:lor1 pl:axd:prp:	$\varphi \rightarrow (\psi \vee \varphi)$	(12.5)
ax:lor2 pl:axd:prp:	$(\varphi \rightarrow \chi) \rightarrow ((\psi \rightarrow \chi) \rightarrow ((\varphi \vee \psi) \rightarrow \chi))$	(12.6)
ax:lor3 pl:axd:prp:	$\varphi \rightarrow (\psi \rightarrow \varphi)$	(12.7)
ax:lif1 pl:axd:prp:	$(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$	(12.8)
ax:lif2 pl:axd:prp:	$(\varphi \rightarrow \psi) \rightarrow ((\varphi \rightarrow \neg\psi) \rightarrow \neg\varphi)$	(12.9)
ax:lnot1 pl:axd:prp:	$\neg\varphi \rightarrow (\varphi \rightarrow \psi)$	(12.10)
ax:lnot2 pl:axd:prp:	$\top$	(12.11)
ax:ltrue pl:axd:prp:	$\perp \rightarrow \varphi$	(12.12)
ax:lfalse1 pl:axd:prp:	$(\varphi \rightarrow \perp) \rightarrow \neg\varphi$	(12.13)
ax:lfalse2 pl:axd:prp:	$\neg\neg\varphi \rightarrow \varphi$	(12.14)
ax:dne		

Definição 12.6 (Modus ponens). If ψ and $\psi \rightarrow \varphi$ already occur in a *derivação*, then φ is a correct inference step.

We'll abbreviate the rule modus ponens as “MP.”

-NoValue-

content/propositional-logic/./first-order-logic/axiomatic-deduction/proving-things.tex

12.3 Examples of Derivações

pl:axd:pro:
sec

Exemplo 12.7. Suppose we want to prove $(\neg\theta \vee \alpha) \rightarrow (\theta \rightarrow \alpha)$. Clearly, this is not an instance of any of our axioms, so we have to use the MP rule to *deriva* it. Our only rule is MP, which given φ and $\varphi \rightarrow \psi$ allows us to justify ψ . One strategy would be to use [Eq. \(12.6\)](#) with φ being $\neg\theta$, ψ being α , and χ being $\theta \rightarrow \alpha$. O.s., the instance

$$(\neg\theta \rightarrow (\theta \rightarrow \alpha)) \rightarrow ((\alpha \rightarrow (\theta \rightarrow \alpha)) \rightarrow ((\neg\theta \vee \alpha) \rightarrow (\theta \rightarrow \alpha))).$$

Why? Two applications of MP yield the last part, which is what we want. And we easily see that $\neg\theta \rightarrow (\theta \rightarrow \alpha)$ is an instance of [Eq. \(12.10\)](#), and $\alpha \rightarrow (\theta \rightarrow \alpha)$ is an instance of [Eq. \(12.7\)](#). So our *derivação* is:

1. $\neg\theta \rightarrow (\theta \rightarrow \alpha)$ Eq. (12.10)
2. $(\neg\theta \rightarrow (\theta \rightarrow \alpha)) \rightarrow$
 $((\alpha \rightarrow (\theta \rightarrow \alpha)) \rightarrow ((\neg\theta \vee \alpha) \rightarrow (\theta \rightarrow \alpha)))$ Eq. (12.6)
3. $(\alpha \rightarrow (\theta \rightarrow \alpha)) \rightarrow ((\neg\theta \vee \alpha) \rightarrow (\theta \rightarrow \alpha))$ 1, 2, MP
4. $\alpha \rightarrow (\theta \rightarrow \alpha)$ Eq. (12.7)
5. $(\neg\theta \vee \alpha) \rightarrow (\theta \rightarrow \alpha)$ 3, 4, MP

Exemplo 12.8. Let's try to find a **derivação** of $\theta \rightarrow \theta$. It is not an instance of an axiom, so we have to use MP to **deriva** it. Eq. (12.7) is an axiom of the form $\varphi \rightarrow \psi$ to which we could apply MP. To be useful, of course, the ψ which MP would justify as a correct step in this case would have to be $\theta \rightarrow \theta$, since this is what we want to **deriva**. That means φ would also have to be θ , O.s., we might look at this instance of Eq. (12.7):

$$\theta \rightarrow (\theta \rightarrow \theta)$$

In order to apply MP, we would also need to justify the corresponding second premise, namely φ . But in our case, that would be θ , and we won't be able to **deriva** θ by itself. So we need a different strategy.

The other axiom involving just $\rightarrow$ is Eq. (12.8), O.s.,

$$(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$$

We could get to the last nested conditional by applying MP twice. Again, that would mean that we want an instance of Eq. (12.8) where $\varphi \rightarrow \chi$ is $\theta \rightarrow \theta$, the **fórmula** we are aiming for. Then of course, φ and χ are both θ . How should we pick ψ so that both $\varphi \rightarrow (\psi \rightarrow \chi)$ and $\varphi \rightarrow \psi$, O.s., in our case $\theta \rightarrow (\psi \rightarrow \theta)$ and $\theta \rightarrow \psi$, are also **deriváveis**? Well, the first of these is already an instance of Eq. (12.7), whatever we decide ψ to be. And $\theta \rightarrow \psi$ would be another instance of Eq. (12.7) if ψ were $(\theta \rightarrow \theta)$. So, our **derivação** is:

1. $\theta \rightarrow ((\theta \rightarrow \theta) \rightarrow \theta)$ Eq. (12.7)
2. $(\theta \rightarrow ((\theta \rightarrow \theta) \rightarrow \theta)) \rightarrow$
 $((\theta \rightarrow (\theta \rightarrow \theta)) \rightarrow (\theta \rightarrow \theta))$ Eq. (12.8)
3. $(\theta \rightarrow (\theta \rightarrow \theta)) \rightarrow (\theta \rightarrow \theta)$ 1, 2, MP
4. $\theta \rightarrow (\theta \rightarrow \theta)$ Eq. (12.7)
5. $\theta \rightarrow \theta$ 3, 4, MP

Exemplo 12.9. Sometimes we want to show that there is a **derivação** of some **fórmula** from some other **fórmulas** Γ . For instance, let's show that we can **deriva** $\varphi \rightarrow \chi$ from $\Gamma = \{\varphi \rightarrow \psi, \psi \rightarrow \chi\}$.

1. $\varphi \rightarrow \psi$ HYP
2. $\psi \rightarrow \chi$ HYP
3. $(\psi \rightarrow \chi) \rightarrow (\varphi \rightarrow (\psi \rightarrow \chi))$ Eq. (12.7)
4. $\varphi \rightarrow (\psi \rightarrow \chi)$ 2, 3, MP
5. $(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow$
 $((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$ Eq. (12.8)
6. $((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$ 4, 5, MP
7. $\varphi \rightarrow \chi$ 1, 6, MP

12.4. PROOF-THEORETIC NOTIONS

The lines labelled “HYP” (for “hypothesis”) indicate that the fórmula on that line is um membro of Γ .

pl:axd:pro: prop:chain **Proposição 12.10.** *If $\Gamma \vdash \varphi \rightarrow \psi$ and $\Gamma \vdash \psi \rightarrow \chi$, then $\Gamma \vdash \varphi \rightarrow \chi$*

Prova. Suppose $\Gamma \vdash \varphi \rightarrow \psi$ and $\Gamma \vdash \psi \rightarrow \chi$. Then there is a derivação of $\varphi \rightarrow \psi$ from Γ ; and a derivação of $\psi \rightarrow \chi$ from Γ as well. Combine these into a single derivação by concatenating them. Now add lines 3–7 of the derivação in the preceding example. This is a derivação of $\varphi \rightarrow \chi$ —which is the last line of the new derivação—from Γ . Note that the justifications of lines 4 and 7 remain valid if the reference to line number 1 is replaced by reference to the last line of the derivação of $\varphi \rightarrow \psi$, and reference to line number 2 by reference to the last line of the derivação of $\psi \rightarrow \chi$. $\square$

Problema 12.1. Show that the following hold by exhibiting derivações from the axioms:

1. $(\varphi \wedge \psi) \rightarrow (\psi \wedge \varphi)$
2. $((\varphi \wedge \psi) \rightarrow \chi) \rightarrow (\varphi \rightarrow (\psi \rightarrow \chi))$
3. $\neg(\varphi \vee \psi) \rightarrow \neg\varphi$

-NoValue-

<content/propositional-logic/./first-order-logic/axiomatic-deduction/proof-theoretic-notions.t>

12.4 Proof-Theoretic Notions

pl:axd:ptn: sec Just as we’ve defined a number of important semantic notions (tautology, entailment, satisfiability), we now define corresponding *proof-theoretic notions*. *explanation* These are not defined by appeal to satisfaction of sentenças in estruturas, but by appeal to the derivabilidade or não derivabilidade of certain formulas. It was an important discovery that these notions coincide. That they do is the content of the *soundness* and *completeness theorems*.

Definição 12.11 (Derivabilidade). A fórmula φ is *derivável* from Γ , written $\Gamma \vdash \varphi$, if there is a derivação from Γ ending in φ .

Definição 12.12 (Theorems). A fórmula φ is a *theorem* if there is a derivação of φ from the empty set. We write $\vdash \varphi$ if φ is a theorem and $\not\vdash \varphi$ if it is not.

Definição 12.13 (Consistency). A set Γ of fórmulas is *consistent* if and only if $\Gamma \not\vdash \perp$; it is *inconsistent* otherwise.

pl:axd:ptn: prop:reflexivity **Proposição 12.14 (Reflexivity).** *If $\varphi \in \Gamma$, then $\Gamma \vdash \varphi$.*

Prova. The fórmula φ by itself is a derivação of φ from Γ . □

Proposição 12.15 (Monotonicity). *If $\Gamma \subseteq \Delta$ and $\Gamma \vdash \varphi$, then $\Delta \vdash \varphi$.* pl:axd:ptn:
prop:monotonicity

Prova. Any derivação of φ from Γ is also a derivação of φ from Δ . □

Proposição 12.16 (Transitivity). *If $\Gamma \vdash \varphi$ and $\{\varphi\} \cup \Delta \vdash \psi$, then $\Gamma \cup \Delta \vdash \psi$.* pl:axd:ptn:
prop:transitivity

Prova. Suppose $\{\varphi\} \cup \Delta \vdash \psi$. Then there is a derivação $\psi_1, \dots, \psi_l = \psi$ from $\{\varphi\} \cup \Delta$. Some of the steps in that derivação will be correct because of a rule which refers to a prior line $\psi_i = \varphi$. By hypothesis, there is a derivação of φ from Γ , O.s., a derivação $\varphi_1, \dots, \varphi_k = \varphi$ where every φ_i is an axiom, um membro of Γ , or correct by a rule of inference. Now consider the sequence

$$\varphi_1, \dots, \varphi_k = \varphi, \psi_1, \dots, \psi_l = \psi.$$

This is a correct derivação of ψ from $\Gamma \cup \Delta$ since every $B_i = \varphi$ is now justified by the same rule which justifies $\varphi_k = \varphi$. □

Note that this means that in particular if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. It follows also that if $\varphi_1, \dots, \varphi_n \vdash \psi$ and $\Gamma \vdash \varphi_i$ for each i , then $\Gamma \vdash \psi$.

Proposição 12.17. *Γ is inconsistent iff $\Gamma \vdash \varphi$ for every φ .* pl:axd:ptn:
prop:incons

Prova. Exercise. □

Problema 12.2. Prove **Proposição 12.17**.

Proposição 12.18 (Compactness). pl:axd:ptn:
prop:proves-compact

1. *If $\Gamma \vdash \varphi$ then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vdash \varphi$.*
2. *If every finite subset of Γ is consistent, then Γ is consistent.*

Prova. 1. If $\Gamma \vdash \varphi$, then there is a finite sequence of fórmulas $\varphi_1, \dots, \varphi_n$ so that $\varphi \equiv \varphi_n$ and each φ_i is either a logical axiom, um membro of Γ or follows from previous fórmulas by modus ponens. Take Γ_0 to be those φ_i which are in Γ . Then the derivação is likewise a derivação from Γ_0 , and so $\Gamma_0 \vdash \varphi$.

2. This is the contrapositive of (1) for the special case $\varphi \equiv \perp$. □

-NoValue-

12.5. THE DEDUCTION THEOREM

12.5 The Deduction Theorem

As we've seen, giving *derivações* in an axiomatic system is cumbersome, and *derivações* may be hard to find. Rather than actually write out long lists of *fórmulas*, it is generally easier to argue that such *derivações* exist, by making use of a few simple results. We've already established three such results: **Proposição 12.14** says we can always assert that $\Gamma \vdash \varphi$ when we know that $\varphi \in \Gamma$. **Proposição 12.15** says that if $\Gamma \vdash \varphi$ then also $\Gamma \cup \{\psi\} \vdash \varphi$. And **Proposição 12.16** implies that if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. Here's another simple result, a “meta”-version of modus ponens:

Proposição 12.19. *If $\Gamma \vdash \varphi$ and $\Gamma \vdash \varphi \rightarrow \psi$, then $\Gamma \vdash \psi$.*

Prova. We have that $\{\varphi, \varphi \rightarrow \psi\} \vdash \psi$:

1. φ Hyp.
2. $\varphi \rightarrow \psi$ Hyp.
3. ψ 1, 2, MP

By **Proposição 12.16**, $\Gamma \vdash \psi$. □

The most important result we'll use in this context is the deduction theorem:

Teorema 12.20 (Deduction Theorem). *$\Gamma \cup \{\varphi\} \vdash \psi$ if and only if $\Gamma \vdash \varphi \rightarrow \psi$.*

Prova. The “if” direction is immediate. If $\Gamma \vdash \varphi \rightarrow \psi$ then also $\Gamma \cup \{\varphi\} \vdash \varphi \rightarrow \psi$ by **Proposição 12.15**. Also, $\Gamma \cup \{\varphi\} \vdash \varphi$ by **Proposição 12.14**. So, by **Proposição 12.19**, $\Gamma \cup \{\varphi\} \vdash \psi$.

For the “only if” direction, we proceed by induction on the length of the *derivação* of ψ from $\Gamma \cup \{\varphi\}$.

For the induction basis, we prove the claim for every *derivação* of length 1. A *derivação* of ψ from $\Gamma \cup \{\varphi\}$ of length 1 consists of ψ by itself; and if it is correct ψ is either $\in \Gamma \cup \{\varphi\}$ or is an axiom. If $\psi \in \Gamma$ or is an axiom, then $\Gamma \vdash \psi$. We also have that $\Gamma \vdash \psi \rightarrow (\varphi \rightarrow \psi)$ by **Eq. (12.7)**, and **Proposição 12.19** gives $\Gamma \vdash \varphi \rightarrow \psi$. If $\psi \in \{\varphi\}$ then $\Gamma \vdash \varphi \rightarrow \psi$ because the last *sentença* $\varphi \rightarrow \psi$ is the same as $\varphi \rightarrow \varphi$, and we have *derivad* that in **Exemplo 12.8**.

For the inductive step, suppose a *derivação* of ψ from $\Gamma \cup \{\varphi\}$ ends with a step ψ which is justified by modus ponens. (If it is not justified by modus ponens, $\psi \in \Gamma$, $\psi \equiv \varphi$, or ψ is an axiom, and the same reasoning as in the induction basis applies.) Then some previous steps in the *derivação* are $\chi \rightarrow \psi$ and χ , for some *fórmula* χ . O.s., $\Gamma \cup \{\varphi\} \vdash \chi \rightarrow \psi$ and $\Gamma \cup \{\varphi\} \vdash \chi$, and the respective *derivações* are shorter, so the inductive hypothesis applies to them. We thus have both:

$$\begin{aligned} \Gamma \vdash \varphi \rightarrow (\chi \rightarrow \psi); \\ \Gamma \vdash \varphi \rightarrow \chi. \end{aligned}$$

But also

$$\Gamma \vdash (\varphi \rightarrow (\chi \rightarrow \psi)) \rightarrow ((\varphi \rightarrow \chi) \rightarrow (\varphi \rightarrow \psi)),$$

by [Eq. \(12.8\)](#), and two applications of [Proposição 12.19](#) give $\Gamma \vdash \varphi \rightarrow \psi$, as required. $\square$

Notice how [Eq. \(12.7\)](#) and [Eq. \(12.8\)](#) were chosen precisely so that the Deduction Theorem would hold.

The following are some useful facts about [derivabilidade](#), which we leave as exercises.

Proposição 12.21.

1. $\vdash (\varphi \rightarrow \psi) \rightarrow ((\psi \rightarrow \chi) \rightarrow (\varphi \rightarrow \chi))$;
2. If $\Gamma \cup \{\neg\varphi\} \vdash \neg\psi$ then $\Gamma \cup \{\psi\} \vdash \varphi$ (Contraposition);
3. $\{\varphi, \neg\varphi\} \vdash \psi$ (Ex Falso Quodlibet, Explosion);
4. $\{\neg\neg\varphi\} \vdash \varphi$ (Double Negation Elimination);
5. If $\Gamma \vdash \neg\neg\varphi$ then $\Gamma \vdash \varphi$;

pl:axd:ded:
prop:derivfacts
pl:axd:ded:
derivfacts:a
pl:axd:ded:
derivfacts:b
pl:axd:ded:
derivfacts:c
pl:axd:ded:
derivfacts:d
pl:axd:ded:
derivfacts:e

Problema 12.3. Prove [Proposição 12.21](#)

-NoValue-

[content/propositional-logic/.../first-order-logic/axiomatic-deduction/provability-co](#)

12.6 Derivabilidade and Consistency

We will now establish a number of properties of the [derivabilidade](#) relation. They are independently interesting, but each will play a role in the proof of the completeness theorem.

pl:axd:prv:
sec

Proposição 12.22. If $\Gamma \vdash \varphi$ and $\Gamma \cup \{\varphi\}$ is inconsistent, then Γ is inconsistent.

pl:axd:prv:
prop:provability-contr

Prova. If $\Gamma \cup \{\varphi\}$ is inconsistent, then $\Gamma \cup \{\varphi\} \vdash \perp$. By [Proposição 12.14](#), $\Gamma \vdash \psi$ for every $\psi \in \Gamma$. Since also $\Gamma \vdash \varphi$ by hypothesis, $\Gamma \vdash \psi$ for every $\psi \in \Gamma \cup \{\varphi\}$. By [Proposição 12.16](#), $\Gamma \vdash \perp$, O.s., Γ is inconsistent. $\square$

Proposição 12.23. $\Gamma \vdash \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ is inconsistent.

pl:axd:prv:
prop:prov-incons

Prova. First suppose $\Gamma \vdash \varphi$. Then $\Gamma \cup \{\neg\varphi\} \vdash \varphi$ by [Proposição 12.15](#). $\Gamma \cup \{\neg\varphi\} \vdash \neg\varphi$ by [Proposição 12.14](#). We also have $\vdash \neg\varphi \rightarrow (\varphi \rightarrow \perp)$ by [Eq. \(12.10\)](#). So by two applications of [Proposição 12.19](#), we have $\Gamma \cup \{\neg\varphi\} \vdash \perp$.

Now assume $\Gamma \cup \{\neg\varphi\}$ is inconsistent, O.s., $\Gamma \cup \{\neg\varphi\} \vdash \perp$. By the deduction theorem, $\Gamma \vdash \neg\varphi \rightarrow \perp$. $\Gamma \vdash (\neg\varphi \rightarrow \perp) \rightarrow \neg\neg\varphi$ by [Eq. \(12.13\)](#), so $\Gamma \vdash \neg\neg\varphi$ by [Proposição 12.19](#). Since $\Gamma \vdash \neg\neg\varphi \rightarrow \varphi$ ([Eq. \(12.14\)](#)), we have $\Gamma \vdash \varphi$ by [Proposição 12.19](#) again. $\square$

12.7. DERIVABILIDADE AND THE PROPOSITIONAL CONNECTIVES

Problema 12.4. Prove that $\Gamma \vdash \neg\varphi$ iff $\Gamma \cup \{\varphi\}$ is inconsistent.

pl:axd:prv: prop:explicit-inc **Proposição 12.24.** If $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$, then Γ is inconsistent.

Prova. $\Gamma \vdash \neg\varphi \rightarrow (\varphi \rightarrow \perp)$ by Eq. (12.10). $\Gamma \vdash \perp$ by two applications of **Proposição 12.19**. $\square$

pl:axd:prv: prop:provability-exhaustive **Proposição 12.25.** If $\Gamma \cup \{\varphi\}$ and $\Gamma \cup \{\neg\varphi\}$ are both inconsistent, then Γ is inconsistent.

Prova. Exercise. $\square$

Problema 12.5. Prove **Proposição 12.25**

-NoValue-

[content/propositional-logic/./first-order-logic/axiomatic-deduction/provability-propositional](#)

12.7 Derivabilidade and the Propositional Connectives

pl:axd:ppr: sec We establish that the **derivabilidade** relation $\vdash$ of axiomatic deduction is strong *explanation* enough to establish some basic facts involving the propositional connectives, such as that $\varphi \wedge \psi \vdash \varphi$ and $\varphi, \varphi \rightarrow \psi \vdash \psi$ (modus ponens). These facts are needed for the proof of the completeness theorem.

pl:axd:ppr: prop:provability-land **Proposição 12.26.**

- pl:axd:ppr: prop:provability-land-left* 1. Both $\varphi \wedge \psi \vdash \varphi$ and $\varphi \wedge \psi \vdash \psi$
- pl:axd:ppr: prop:provability-land-right* 2. $\varphi, \psi \vdash \varphi \wedge \psi$.

Prova. 1. From Eq. (12.1) and Eq. (12.1) by modus ponens.

2. From Eq. (12.3) by two applications of modus ponens. $\square$

pl:axd:ppr: prop:provability-lor **Proposição 12.27.**

1. $\varphi \vee \psi, \neg\varphi, \neg\psi$ is inconsistent.
2. Both $\varphi \vdash \varphi \vee \psi$ and $\psi \vdash \varphi \vee \psi$.

Prova. 1. From Eq. (12.9) we get $\vdash \neg\varphi \rightarrow (\varphi \rightarrow \perp)$ and $\vdash \neg\psi \rightarrow (\psi \rightarrow \perp)$. So by the deduction theorem, we have $\{\neg\varphi\} \vdash \varphi \rightarrow \perp$ and $\{\neg\psi\} \vdash \psi \rightarrow \perp$. From Eq. (12.6) we get $\{\neg\varphi, \neg\psi\} \vdash (\varphi \vee \psi) \rightarrow \perp$. By the deduction theorem, $\{\varphi \vee \psi, \neg\varphi, \neg\psi\} \vdash \perp$.

2. From Eq. (12.4) and Eq. (12.5) by modus ponens. $\square$

Proposição 12.28.

1. $\varphi, \varphi \rightarrow \psi \vdash \psi$.
2. Both $\neg\varphi \vdash \varphi \rightarrow \psi$ and $\psi \vdash \varphi \rightarrow \psi$.

*pl:axd:ppr:
prop:provability-lif
pl:axd:ppr:
prop:provability-lif-left
pl:axd:ppr:
prop:provability-lif-right*

Prova. 1. We can **deriva**:

1. φ HYP
2. $\varphi \rightarrow \psi$ HYP
3. ψ 1, 2, MP

2. By **Eq. (12.10)** and **Eq. (12.7)** and the deduction theorem, respectively.
 $\square$

-NoValue-

`content/propositional-logic/./first-order-logic/axiomatic-deduction/soundness.tex`

12.8 Soundness

explanation A **derivação** system, such as axiomatic deduction, is *sound* if it cannot **deriva** things that do not actually hold. Soundness is thus a kind of guaranteed safety property for **derivação** systems. Depending on which proof theoretic property is in question, we would like to know for instance, that

*pl:axd:sou:
sec*

1. every **derivável** φ is valid;
2. if φ is **derivável** from some others Γ , it is also a consequence of them;
3. if a set of **fórmulas** Γ is inconsistent, it is unsatisfiable.

These are important properties of a **derivação** system. If any of them do not hold, the **derivação** system is deficient—it would **deriva** too much. Consequently, establishing the soundness of a **derivação** system is of the utmost importance.

Proposição 12.29. *If φ is an axiom, then $\mathbf{v} \models \varphi$ for each **valoração** $\mathbf{v}$.*

Prova. Do truth tables for each axiom to verify that they are tautologies. $\square$

Teorema 12.30 (Soundness). *If $\Gamma \vdash \varphi$ then $\Gamma \models \varphi$.*

*pl:axd:sou:
thm:soundness*

Prova. By induction on the length of the **derivação** of φ from Γ . If there are no steps justified by inferences, then all **fórmulas** in the **derivação** are either instances of axioms or are in Γ . By the previous proposition, all the axioms are tautologies, and hence if φ is an axiom then $\Gamma \models \varphi$. If $\varphi \in \Gamma$, then trivially $\Gamma \models \varphi$.

If the last step of the *derivação* of φ is justified by modus ponens, then there are *fórmulas* ψ and $\psi \rightarrow \varphi$ in the *derivação*, and the induction hypothesis applies to the part of the *derivação* ending in those *fórmulas* (since they contain at least one fewer step justified by an inference). So, by induction hypothesis, $\Gamma \models \psi$ and $\Gamma \models \psi \rightarrow \varphi$. Then $\Gamma \models \varphi$ by *Teorema 7.21*.

pl:axd:sou:
cor:weak-soundness

Corolário 12.31. *If $\vdash \varphi$, then φ is a tautology.*

pl:axd:sou:
cor:consistency-soundness

Corolário 12.32. *If Γ is satisfiable, then it is consistent.*

Prova. We prove the contrapositive. Suppose that Γ is not consistent. Then $\Gamma \vdash \perp$. O.s., there is a *derivação* of $\perp$ from Γ . By *Teorema 12.30*, any *valoração* $\mathbf{v}$ that satisfies Γ must satisfy $\perp$. Since $\mathbf{v} \neq \perp$ for every *valoração* $\mathbf{v}$, no $\mathbf{v}$ can satisfy Γ , O.s., Γ is not satisfiable. $\square$

Capítulo 13

O Teorema da Completude

`content/propositional-logic/./first-order-logic/completeness/introduction.tex`

13.1 Introdução

pl:com:int:
sec

O teorema da completude é um dos resultados mais fundamentais sobre lógica. Ele vem em duas formulações, cuja equivalência iremos provar. Em sua primeira formulação, ele diz algo fundamental sobre a relação entre consequência semântica e nosso sistema de *derivação*: se a *sentença* φ segue de algumas *sentenças* Γ , então também existe a *derivação* que estabelece $\Gamma \vdash \varphi$. Assim, o sistema de *derivação* é tão forte quanto pode possivelmente ser sem provar coisas que de fato não se seguem.

Em sua segunda formulação, ele pode ser enunciado como um resultado de existência de modelo: todo conjunto consistente de *sentenças* é satisfatível. A consistência é uma noção da teoria da prova: ela diz que nosso sistema de *derivação* é incapaz de produzir certas *derivações*. Mas quem garante que, só porque não há *derivações* de um certo tipo a partir de Γ , está garantido que

existe **avaliação** $\mathfrak{v}$ com $\mathfrak{v} \models \Gamma$? Antes de o teorema da completude ser provado pela primeira vez — na verdade, antes de termos os sistemas de **derivação** que temos hoje — o grande matemático alemão David Hilbert defendia a visão de que a consistência das teorias matemáticas garante a existência dos objetos de que elas tratam. Ele colocou isso da seguinte forma em uma carta a Gottlob Frege:

Se os axiomas dados arbitrariamente não se contradizem uns aos outros com todas as suas consequências, então eles são verdadeiros e as coisas definidas pelos axiomas existem. Este é para mim o critério de verdade e existência.

Frege discordou veementemente. A segunda formulação do teorema da completude mostra que Hilbert estava certo pelo menos no sentido de que, se os axiomas são consistentes, então existe *alguma* **avaliação** que os torna todos verdadeiros.

Estas não são as únicas razões pelas quais o teorema da completude — ou melhor, sua prova — é importante. Ele tem uma série de consequências importantes, algumas das quais discutiremos separadamente. Por exemplo, como toda **derivação** que mostra $\Gamma \vdash \varphi$ é finita e, portanto, só pode usar uma quantidade finita das **sentenças** em Γ , segue do teorema da completude que, se φ é uma consequência de Γ , então já é uma consequência de um subconjunto finito de Γ . Isso é chamado de *compacidade*. Equivalentemente, se todo subconjunto finito de Γ é consistente, então o próprio Γ deve ser consistente.

Embora o teorema da compacidade decorra do teorema da completude pelo desvio através de **derivações**, também é possível usar a *prova* do teorema da completude para estabelecê-lo diretamente. Pois o que a prova faz é tomar um conjunto de **sentenças** com uma certa propriedade — a consistência — e construir a **estrutura** a partir desse conjunto que possui certas propriedades (neste caso, que ela satisfaz o conjunto). Quase exatamente a mesma construção pode ser usada para estabelecer a compacidade diretamente, partindo de conjuntos “finitamente satisfatíveis” de **sentenças** em vez de conjuntos consistentes.

`content/propositional-logic/./first-order-logic/completeness/outline.tex`

13.2 Esboço da Prova

A prova do teorema da completude é um pouco complexa e, em uma primeira leitura, é fácil se perder. Então, vamos esboçar a prova. O primeiro passo é uma mudança de perspectiva, que nos permite enxergar um caminho para uma prova. Quando a completude é pensada como “sempre que $\Gamma \models \varphi$ então $\Gamma \vdash \varphi$ ”, pode ser difícil até mesmo ter uma ideia: pois, para mostrar que $\Gamma \vdash \varphi$, temos que encontrar a **derivação**, e não parece que a hipótese de que $\Gamma \models \varphi$ nos ajude nisso de alguma forma. Para alguns sistemas de prova, é possível construir diretamente a **derivação**, mas adotaremos uma abordagem um pouco diferente. A mudança de perspectiva necessária é esta: a completude também pode ser

pl:com:out:
sec

13.2. ESBOÇO DA PROVA

formulada como: “se Γ é consistente, então é satisfatível”. Talvez possamos usar a informação em Γ juntamente com a hipótese de que ela é consistente para construir a **valoração** que satisfaça toda **fórmula** em Γ . Afinal, sabemos que tipo de **valoração** estamos procurando: uma que seja como Γ a descreve!

Se Γ contém apenas **variáveis proposicionais**, é fácil construir um modelo para ele. Tudo o que temos a fazer é apresentar a **valoração** $\mathbf{v}$ tal que $\mathbf{v} \models p$ para todo $p \in \Gamma$. Bem, faça $\mathbf{v}(p) = \mathbb{T}$ se e somente se $p \in \Gamma$.

Agora suponha que Γ contenha alguma **fórmula** $\neg\psi$, com ψ atômica. Poderíamos temer que a construção de $\mathbf{v}$ interfira na possibilidade de tornar $\neg\psi$ verdadeira. Mas é aqui que entra a consistência de Γ : se $\neg\psi \in \Gamma$, então $\psi \notin \Gamma$, caso contrário Γ seria inconsistente. E se $\psi \notin \Gamma$, então, de acordo com nossa construção de $\mathbf{v}$, $\mathbf{v} \not\models \psi$, logo $\mathbf{v} \models \neg\psi$. Até aqui, tudo bem.

E se Γ contiver fórmulas complexas, não atômicas? Digamos que ele contenha $\varphi \wedge \psi$. Para tornar isso verdadeiro, devemos proceder como se tanto φ quanto ψ estivessem em Γ . E se $\varphi \vee \psi \in \Gamma$, então teremos que tornar pelo menos um deles verdadeiro, isto é, proceder como se um deles estivesse em Γ .

Isso sugere a seguinte ideia: adicionamos **fórmulas** adicionais a Γ de modo a (a) manter o conjunto resultante consistente e (b) garantir que, para toda **sentença** φ atômica possível, ou φ está no conjunto resultante, ou $\neg\varphi$ está, e (c) tais que, sempre que $\varphi \wedge \psi$ está no conjunto, então tanto φ quanto ψ também estão; se $\varphi \vee \psi$ está no conjunto, pelo menos um dentre φ ou ψ também está, etc. Continuamos fazendo isso (potencialmente para sempre). Chame o conjunto de todas as **fórmulas** assim adicionadas de Γ^* . Então a nossa construção acima nos forneceria a **valoração** $\mathbf{v}$ para a qual poderíamos provar, por indução, que ela satisfaz todas as sentenças em Γ^* e, portanto, também todas as sentenças em Γ , já que $\Gamma \subseteq \Gamma^*$. Acontece que garantir (a) e (b) é suficiente. Um conjunto de sentenças para o qual (b) vale é chamado de *completo*. Assim, nossa tarefa será estender o conjunto consistente Γ a um conjunto consistente e completo Γ^* .

Então, aqui está o que faremos. Primeiro investigamos as propriedades dos conjuntos consistentes **completos**; em particular, provamos que um conjunto consistente **completo** contém $\varphi \wedge \psi$ se e somente se contém tanto φ quanto ψ , contém $\varphi \vee \psi$ se e somente se contém pelo menos um deles, etc. (**Proposição 13.2**). Em seguida, tomaremos o conjunto consistente Γ e mostraremos que ele pode ser estendido a um conjunto consistente e **completo** Γ^* (**Lema 13.3**). Este conjunto Γ^* é o que usaremos para definir nossa **valoração** $\mathbf{v}(\Gamma^*)$. A valoração é determinada pelas **variáveis proposicionais** em Γ^* (**Definição 13.4**). Usaremos as propriedades dos conjuntos consistentes completos para mostrar que, de fato, $\mathbf{v}(\Gamma^*) \models \varphi$ se e somente se $\varphi \in \Gamma^*$ (**Lema 13.5**) e, portanto, em particular, $\mathbf{v}(\Gamma^*) \models \Gamma$.

13.3 Conjuntos Consistentes Completos de Sentenças

Definição 13.1 (Conjunto completo). Um conjunto Γ de sentenças é *completo* se e somente se, para qualquer sentença φ , ou $\varphi \in \Gamma$ ou $\neg\varphi \in \Gamma$.

pl:com:ccs:
sec
pl:com:ccs:
def:complete-set

explanation Conjuntos completos de sentenças não deixam questões sem resposta. Para qualquer sentença φ , Γ “diz” se φ é verdadeira ou falsa. A importância dos conjuntos completos estende-se para além da prova do teorema da completude. Uma teoria que é completa e axiomatizável, por exemplo, é sempre decidível.

explanation Conjuntos consistentes completos são importantes na prova de completude já que podemos garantir que todo conjunto consistente de sentenças Γ está contido em um conjunto consistente completo Γ^* . Um conjunto consistente completo contém, para cada sentença φ , ou φ ou sua negação $\neg\varphi$, mas não ambas. Isso é verdade em particular para variáveis proposicionais, de modo que, a partir de um conjunto consistente completo, podemos construir a *valoração* onde a valor de verdade atribuído às variáveis proposicionais é definida de acordo com quais variáveis proposicionais estão em Γ^* . Pode-se então mostrar que esta *valoração* torna verdadeiras todas as sentenças em Γ^* (e, portanto, também todas aquelas em Γ). A prova deste último fato requer que $\neg\varphi \in \Gamma^*$ se e somente se $\varphi \notin \Gamma^*$, $(\varphi \vee \psi) \in \Gamma^*$ se e somente se $\varphi \in \Gamma^*$ ou $\psi \in \Gamma^*$, etc.

No que se segue, usaremos com frequência, de modo tácito, as propriedades de reflexividade, monotonicidade e transitividade de $\vdash$ (ver Seções 9.6, 10.5, 11.5 e 12.4).

Proposição 13.2. Suponha que Γ seja completo e consistente. Então:

pl:com:ccs:
prop:ccs
pl:com:ccs:
prop:ccs-prov-in
pl:com:ccs:
prop:ccs-and
pl:com:ccs:
prop:ccs-or
pl:com:ccs:
prop:ccs-if

1. Se $\Gamma \vdash \varphi$, então $\varphi \in \Gamma$.
2. $\varphi \wedge \psi \in \Gamma$ se e somente se $\varphi \in \Gamma$ e $\psi \in \Gamma$.
3. $\varphi \vee \psi \in \Gamma$ se e somente se $\varphi \in \Gamma$ ou $\psi \in \Gamma$.
4. $\varphi \rightarrow \psi \in \Gamma$ se e somente se $\varphi \notin \Gamma$ ou $\psi \in \Gamma$.

Prova. Suponhamos, para tudo o que se segue, que Γ seja completo e consistente.

1. Se $\Gamma \vdash \varphi$, então $\varphi \in \Gamma$.

Suponha que $\Gamma \vdash \varphi$. Suponha, ao contrário, que $\varphi \notin \Gamma$. Como Γ é completo, $\neg\varphi \in \Gamma$. Por Proposições 9.19, 10.17, 11.17 e 12.24, Γ é inconsistente. Isso contradiz a suposição de que Γ é consistente. Logo, não pode ser o caso que $\varphi \notin \Gamma$, de modo que $\varphi \in \Gamma$.

2. $\varphi \wedge \psi \in \Gamma$ se e somente se $\varphi \in \Gamma$ e $\psi \in \Gamma$:

Para a direção direta, suponha $\varphi \wedge \psi \in \Gamma$. Então por Proposições 9.21, 10.19, 11.19 e 12.26, item (1), $\Gamma \vdash \varphi$ e $\Gamma \vdash \psi$. Por (1), $\varphi \in \Gamma$ e $\psi \in \Gamma$, como requerido.

13.4. LEMA DE LINDENBAUM

Para a direção reversa, sejam $\varphi \in \Gamma$ e $\psi \in \Gamma$. Por [Proposições 9.21, 10.19, 11.19 e 12.26](#), item (2), $\Gamma \vdash \varphi \wedge \psi$. Por (1), $\varphi \wedge \psi \in \Gamma$.

3. Primeiro mostramos que, se $\varphi \vee \psi \in \Gamma$, então $\varphi \in \Gamma$ ou $\psi \in \Gamma$. Suponha $\varphi \vee \psi \in \Gamma$ mas $\varphi \notin \Gamma$ e $\psi \notin \Gamma$. Como Γ é [completo](#), $\neg\varphi \in \Gamma$ e $\neg\psi \in \Gamma$. Por [Proposições 9.22, 10.20, 11.20 e 12.27](#), item (1), Γ é inconsistente, uma contradição. Logo, $\varphi \in \Gamma$ ou $\psi \in \Gamma$.

Para a direção reversa, suponha que $\varphi \in \Gamma$ ou $\psi \in \Gamma$. Por [Proposições 9.22, 10.20, 11.20 e 12.27](#), item (2), $\Gamma \vdash \varphi \vee \psi$. Por (1), $\varphi \vee \psi \in \Gamma$, como requerido.

4. Para a direção direta, suponha $\varphi \rightarrow \psi \in \Gamma$ e suponha, ao contrário, que $\varphi \in \Gamma$ e $\psi \notin \Gamma$. Sob essas suposições, $\varphi \rightarrow \psi \in \Gamma$ e $\varphi \in \Gamma$. Por [Proposições 9.23, 10.21, 11.21 e 12.28](#), item (1), $\Gamma \vdash \psi$. Mas então, por (1), $\psi \in \Gamma$, contradizendo a suposição de que $\psi \notin \Gamma$.

Para a direção reversa, considere primeiro o caso em que $\varphi \notin \Gamma$. Como Γ é [completo](#), $\neg\varphi \in \Gamma$. Por [Proposições 9.23, 10.21, 11.21 e 12.28](#), item (2), $\Gamma \vdash \varphi \rightarrow \psi$. Novamente por (1), obtemos que $\varphi \rightarrow \psi \in \Gamma$, como requerido.

Agora considere o caso em que $\psi \in \Gamma$. Por [Proposições 9.23, 10.21, 11.21 e 12.28](#), item (2) novamente, $\Gamma \vdash \varphi \rightarrow \psi$. Por (1), $\varphi \rightarrow \psi \in \Gamma$. $\square$

Problema 13.1. Complete a prova de [Proposição 13.2](#).

[content/propositional-logic/./first-order-logic/completeness/lindenbaums-lemma.tex](#)

13.4 Lema de Lindenbaum

pl:com:lin: sec Provamos agora um lema que mostra que qualquer conjunto consistente de explanation [sentenças](#) está contido em algum conjunto de sentenças que não é apenas consistente, mas também [completo](#). A prova funciona adicionando uma [sentença](#) de cada vez, garantindo a cada passo que o conjunto permanece consistente. Fazemos isso de modo que, para todo φ , ou φ ou $\neg\varphi$ seja adicionada em algum estágio. A união de todos os estágios dessa construção então contém ou φ ou sua negação $\neg\varphi$ e é, portanto, completa. Ela também é consistente, já que nos asseguramos, a cada estágio, de não introduzir uma inconsistência.

pl:com:lin: lem:lindenbaum **Lema 13.3 (Lema de Lindenbaum).** *Todo conjunto consistente Γ em uma linguagem $\mathcal{L}$ pode ser estendido a um conjunto consistente [completo](#) Γ^* .*

Prova. Seja Γ consistente. Sejam $\varphi_0, \varphi_1, \dots$ uma enumeração de todas as [sentenças](#) de $\mathcal{L}$. Defina $\Gamma_0 = \Gamma$, e

$$\Gamma_{n+1} = \begin{cases} \Gamma_n \cup \{\varphi_n\} & \text{se } \Gamma_n \cup \{\varphi_n\} \text{ é consistente;} \\ \Gamma_n \cup \{\neg\varphi_n\} & \text{caso contrário.} \end{cases}$$

Seja $\Gamma^* = \bigcup_{n \geq 0} \Gamma_n$.

Cada Γ_n é consistente: Γ_0 é consistente por definição. Se $\Gamma_{n+1} = \Gamma_n \cup \{\varphi_n\}$, isso é porque este último é consistente. Se não é, $\Gamma_{n+1} = \Gamma_n \cup \{\neg\varphi_n\}$. Temos que verificar que $\Gamma_n \cup \{\neg\varphi_n\}$ é consistente. Suponha que não seja. Então ambos $\Gamma_n \cup \{\varphi_n\}$ e $\Gamma_n \cup \{\neg\varphi_n\}$ são inconsistentes. Isso significa que Γ_n seria inconsistente por [Proposições 9.20, 10.18, 11.18 e 12.25](#), contrariando a hipótese de indução.

Para todo n e todo $i < n$, $\Gamma_i \subseteq \Gamma_n$. Isso segue por uma simples indução sobre n . Para $n = 0$, não há $i < 0$, de modo que a afirmação vale automaticamente. Para o passo indutivo, suponha que ela seja verdadeira para n . Mostramos que, se $i < n + 1$, então $\Gamma_i \subseteq \Gamma_{n+1}$. Temos $\Gamma_{n+1} = \Gamma_n \cup \{\varphi_n\}$ ou $\Gamma_n \cup \{\neg\varphi_n\}$ por construção. Logo $\Gamma_n \subseteq \Gamma_{n+1}$. Se $i < n + 1$, então $\Gamma_i \subseteq \Gamma_n$ por hipótese indutiva (se $i < n$) ou pelo fato trivial de que $\Gamma_n \subseteq \Gamma_n$ (se $i = n$). Obtemos que $\Gamma_i \subseteq \Gamma_{n+1}$ por transitividade de $\subseteq$.

Disso segue que Γ^* é consistente. Eis o porquê: Seja $\Gamma' \subseteq \Gamma^*$ finito. Cada $\psi \in \Gamma'$ também está em Γ_i para algum i . Seja n o maior destes. Como $\Gamma_i \subseteq \Gamma_n$ se $i \leq n$, todo $\psi \in \Gamma'$ também está em Γ_n , isto é, $\Gamma' \subseteq \Gamma_n$, e Γ_n é consistente. Assim, todo subconjunto finito $\Gamma' \subseteq \Gamma^*$ é consistente. Por [Proposições 9.16, 10.14, 11.14 e 12.18](#), Γ^* é consistente.

Toda [sentença](#) de $\text{Frm}(\mathcal{L})$ aparece na lista usada para definir Γ^* . Se $\varphi_n \notin \Gamma^*$, então isso é porque $\Gamma_n \cup \{\varphi_n\}$ era inconsistente. Mas então $\neg\varphi_n \in \Gamma^*$, de modo que Γ^* é [completo](#). $\square$

[content/propositional-logic/.../first-order-logic/completeness/construction-of-model](#)

13.5 Construção de um Modelo

[explanation](#)

Estamos agora prontos para definir [a valoração](#) que torna verdadeiras todas as $\varphi \in \Gamma$. Para fazer isso, primeiro aplicamos o Lema de Lindenbaum: obtemos um $\Gamma^* \supseteq \Gamma$ consistente completo. Deixamos as [variáveis proposicionais](#) em Γ^* determinarem $\mathbf{v}(\Gamma^*)$.

[pl:com:mod:sec](#)

Definição 13.4. Suponha que Γ^* seja um conjunto consistente completo de [fórmulas](#). Então deixamos

[pl:com:mod:defn:termmodel](#)

$$\mathbf{v}(\Gamma^*)(p) = \begin{cases} \mathbb{T} & \text{se } p \in \Gamma^* \\ \mathbb{F} & \text{se } p \notin \Gamma^* \end{cases}$$

Lema 13.5 (Lema da Verdade). $\mathbf{v}(\Gamma^*) \models \varphi$ se e somente se $\varphi \in \Gamma^*$.

[pl:com:mod:lem:truth](#)

Prova. Provamos ambas as direções simultaneamente e por indução sobre φ .

1. $\varphi \equiv \perp$: $\mathbf{v}(\Gamma^*) \not\models \perp$ por definição de satisfação. Por outro lado, $\perp \notin \Gamma^*$ já que Γ^* é consistente.

13.6. O TEOREMA DA COMPLETUDE

2. $\varphi \equiv \top$: $\mathbf{v}(\Gamma^*) \models \top$ por definição de satisfação. Por outro lado, $\top \in \Gamma^*$ já que Γ^* é consistente e **completo**, e $\Gamma^* \vdash \top$.
3. $\varphi \equiv p$: $\mathbf{v}(\Gamma^*) \models p$ se e somente se $\mathbf{v}(\Gamma^*)(p) = \mathbb{T}$ (pela definição de satisfação) se e somente se $p \in \Gamma^*$ (pela construção de $\mathbf{v}(\Gamma^*)$).
4. $\varphi \equiv \neg\psi$: $\mathbf{v}(\Gamma^*) \models \varphi$ se e somente se $\mathbf{v}(\Gamma^*) \not\models \psi$ (por definição de satisfação). Por hipótese de indução, $\mathbf{v}(\Gamma^*) \not\models \psi$ se e somente se $\psi \notin \Gamma^*$. Como Γ^* é consistente e **completo**, $\psi \notin \Gamma^*$ se e somente se $\neg\psi \in \Gamma^*$.
5. $\varphi \equiv \psi \wedge \chi$: $\mathbf{v}(\Gamma^*) \models \varphi$ se e somente se temos ambos $\mathbf{v}(\Gamma^*) \models \psi$ e $\mathbf{v}(\Gamma^*) \models \chi$ (por definição de satisfação) se e somente se $\psi \in \Gamma^*$ e $\chi \in \Gamma^*$ (pela hipótese de indução). Por **Proposição 13.2(2)**, este é o caso se e somente se $(\psi \wedge \chi) \in \Gamma^*$.
6. $\varphi \equiv \psi \vee \chi$: $\mathbf{v}(\Gamma^*) \models \varphi$ se e somente se $\mathbf{v}(\Gamma^*) \models \psi$ ou $\mathbf{v}(\Gamma^*) \models \chi$ (por definição de satisfação) se e somente se $\psi \in \Gamma^*$ ou $\chi \in \Gamma^*$ (por hipótese de indução). Este é o caso se e somente se $(\psi \vee \chi) \in \Gamma^*$ (por **Proposição 13.2(3)**).
7. $\varphi \equiv \psi \rightarrow \chi$: $\mathbf{v}(\Gamma^*) \models \varphi$ se e somente se $\mathbf{v}(\Gamma^*) \not\models \psi$ ou $\mathbf{v}(\Gamma^*) \models \chi$ (por definição de satisfação) se e somente se $\psi \notin \Gamma^*$ ou $\chi \in \Gamma^*$ (por hipótese de indução). Este é o caso se e somente se $(\psi \rightarrow \chi) \in \Gamma^*$ (por **Proposição 13.2(4)**).

content/propositional-logic/./first-order-logic/completeness/completeness-thm.tex

13.6 O Teorema da Completude

pl:com:cth:
sec

Vamos combinar nossos resultados: chegamos ao teorema da completude.

explanation

pl:com:cth:
thm:completeness

Teorema 13.6 (Teorema da Completude). *Seja Γ um conjunto de **sentenças**. Se Γ é consistente, então é satisfatível.*

Prova. Suponha que Γ seja consistente. Por **Lema 13.3**, existe um $\Gamma^* \supseteq \Gamma$ que é consistente e **completo**. Por **Lema 13.5**, $\mathbf{v}(\Gamma^*) \models \varphi$ se e somente se $\varphi \in \Gamma^*$. Disso segue, em particular, que, para todo $\varphi \in \Gamma$, $\mathbf{v}(\Gamma^*) \models \varphi$, de modo que Γ é satisfatível. $\square$

pl:com:cth:
cor:completeness

Corolário 13.7 (Teorema da Completude, Segunda Versão). *Para todo Γ e **sentenças** φ : se $\Gamma \models \varphi$ então $\Gamma \vdash \varphi$.*

Prova. Note que os Γ em **Corolário 13.7** e **Teorema 13.6** são universalmente quantificados. Para nos certificarmos de que não nos confundimos, reformulemos **Teorema 13.6** usando uma variável diferente: para qualquer conjunto de **sentenças** Δ , se Δ é consistente, então é satisfatível. Por contraposição,

se Δ não é satisfatível, então Δ é inconsistente. Usaremos isso para provar o corolário.

Suponha que $\Gamma \models \varphi$. Então $\Gamma \cup \{\neg\varphi\}$ é insatisfatível por [Proposição 7.20](#). Tomando $\Gamma \cup \{\neg\varphi\}$ como nosso Δ , a versão anterior de [Teorema 13.6](#) nos dá que $\Gamma \cup \{\neg\varphi\}$ é inconsistente. Por [Proposições 9.18, 10.16, 11.16 e 12.23](#), $\Gamma \vdash \varphi$. $\square$

Problema 13.2. Use [Corolário 13.7](#) para provar [Teorema 13.6](#), mostrando assim que as duas formulações do teorema da completude são equivalentes.

Problema 13.3. Para que um sistema de [derivação](#) seja completo, suas regras devem ser fortes o suficiente para provar que todo conjunto insatisfatível é inconsistente. Quais das regras de [derivação](#) foram necessárias para provar a completude? Alguma dessas regras não é usada em nenhum lugar da prova? Para responder a essas perguntas, faça uma lista ou diagrama que mostre quais das regras de [derivação](#) foram usadas em quais resultados que levam à prova de [Teorema 13.6](#). Certifique-se de anotar quaisquer usos tácitos de regras nessas provas.

[content/propositional-logic/./first-order-logic/completeness/compactness.tex](#)

13.7 O Teorema da Compacidade

Uma consequência importante do teorema da completude é o teorema da compacidade. O teorema da compacidade afirma que, se cada subconjunto *finito* de um conjunto de [sentenças](#) é satisfatível, o conjunto inteiro é satisfatível — mesmo que o próprio conjunto seja infinito. Isso está longe de ser óbvio. Não há nada que pareça descartar, ao menos à primeira vista, a possibilidade de existirem conjuntos infinitos de [sentenças](#) que sejam contraditórios, mas em que a contradição só surge, por assim dizer, do número infinito. O teorema da compacidade diz que tal cenário pode ser descartado: não existem conjuntos infinitos insatisfatíveis de [sentenças](#) tais que cada subconjunto finito deles seja satisfatível. Como o teorema da completude, ele tem uma versão relacionada à consequência lógica: se um conjunto infinito de [sentenças](#) implica algo, então já um subconjunto finito o implica.

[pl:com:com:](#)
[sec](#)

Definição 13.8. Um conjunto Γ de [fórmulas](#) é *finitamente satisfatível* se e somente se todo $\Gamma_0 \subseteq \Gamma$ finito é satisfatível.

Teorema 13.9 (Teorema da Compacidade). *O seguinte vale para quaisquer sentenças Γ e φ :*

[pl:com:com:](#)
[thm:compactness](#)

1. $\Gamma \models \varphi$ se e somente se existe um $\Gamma_0 \subseteq \Gamma$ finito tal que $\Gamma_0 \models \varphi$.
2. Γ é satisfatível se e somente se é finitamente satisfatível.

13.8. UMA PROVA DIRETA DO TEOREMA DA COMPACIDADE

Prova. Provamos (2). Se Γ é satisfatível, então existe a **avaliação** $\mathbf{v}$ tal que $\mathbf{v} \models \varphi$ para todo $\varphi \in \Gamma$. É claro que essa $\mathbf{v}$ também satisfaz todo subconjunto finito de Γ , de modo que Γ é finitamente satisfatível.

Agora suponha que Γ seja finitamente satisfatível. Então todo subconjunto finito $\Gamma_0 \subseteq \Gamma$ é satisfatível. Pela correção (**Corolários 9.28, 10.24, 11.26 e 12.32**), todo subconjunto finito é consistente. Então o próprio Γ deve ser consistente por **Proposições 9.16, 10.14, 11.14 e 12.18**. Pela completude (**Teorema 13.6**), como Γ é consistente, ele é satisfatível. $\square$

Problema 13.4. Prove (1) de **Teorema 13.9**.

content/propositional-logic/./first-order-logic/completeness/compactness-direct.tex

13.8 Uma Prova Direta do Teorema da Compacidade

pl:com:cpd:
sec

Podemos provar o Teorema da Compacidade diretamente, sem apelar ao Teorema da Completude, usando as mesmas ideias da prova do teorema da completude. Na prova do Teorema da Completude, começamos com um conjunto Γ consistente de **sentenças**, expandimos-o para um conjunto Γ^* de **sentenças** consistente e **completo**, e então mostramos que no **avaliação** $\mathbf{v}(\Gamma^*)$ construído a partir de Γ^* , todas as **sentenças** de Γ são verdadeiras, de modo que Γ é satisfatível.

Podemos usar o mesmo método para mostrar que um conjunto finitamente satisfatível de sentenças é satisfatível. Só temos que provar as versões correspondentes dos resultados que levam ao lema da verdade, substituindo “consistente” por “finitamente satisfatível”.

pl:com:cpd:
prop:fsat-ccs

Proposição 13.10. *Suponha que Γ é **completo** e finitamente satisfatível. Então:*

1. $(\varphi \wedge \psi) \in \Gamma$ se, e somente se, tanto $\varphi \in \Gamma$ quanto $\psi \in \Gamma$.
2. $(\varphi \vee \psi) \in \Gamma$ se, e somente se, $\varphi \in \Gamma$ ou $\psi \in \Gamma$.
3. $(\varphi \rightarrow \psi) \in \Gamma$ se, e somente se, $\varphi \notin \Gamma$ ou $\psi \in \Gamma$.

Problema 13.5. Prove **Proposição 13.10**. Evite o uso de $\vdash$.

pl:com:cpd:
lem:fsat-lindenbaum

Lema 13.11. *Todo conjunto finitamente satisfatível Γ pode ser estendido a um conjunto **completo** e finitamente satisfatível Γ^* .*

Problema 13.6. Prove **Lema 13.11**. (Dica: o passo crucial é mostrar que se Γ_n é finitamente satisfatível, então ou $\Gamma_n \cup \{\varphi_n\}$ ou $\Gamma_n \cup \{\neg\varphi_n\}$ é finitamente satisfatível.)

pl:com:cpd:
thm:compactness-direct

Teorema 13.12 (Compacidade). *Γ é satisfatível se, e somente se, é finitamente satisfatível.*

CAPÍTULO 13. O TEOREMA DA COMPLETUDE

Prova. Se Γ é satisfatível, então há a **avaliação** $\mathfrak{v}$ tal que $\mathfrak{v} \models \varphi$ para todo $\varphi \in \Gamma$. É claro que esta $\mathfrak{v}$ também satisfaz todo subconjunto finito de Γ , de modo que Γ é finitamente satisfatível.

Agora suponha que Γ é finitamente satisfatível. Por **Lema 13.11**, Γ pode ser estendido a um conjunto **completo** e finitamente satisfatível Γ^* . Construa o **avaliação** $\mathfrak{v}(\Gamma^*)$ como em **Definição 13.4**. A prova do Lema da Verdade (**Lema 13.5**) funciona se substituírmos referências a **Proposição 13.2**. $\square$

Problema 13.7. Escreva a prova completa do Lema da Verdade (**Lema 13.5**) na versão necessária para a prova de **Teorema 13.12**.

Parte III

Lógica de Primeira Ordem

Esta parte cobre a metateoria da lógica de primeira ordem através completude. Atualmente não depende de um tratamento separado de lógica proposicional; tudo está provado. Os arquivos de origem serão exclua o material sobre quantificadores (e substitua “*estrutura*” com “*valoração*”, $\mathfrak{M}$ com $\mathfrak{v}$, etc.) se a tag “FOL” é falsa. Na verdade, a maior parte do material da peça na lógica proposicional é simplesmente o material de primeira ordem com o Etiqueta “FOL” desativada.

Se a parte sobre lógica proposicional for incluída, isso resulta em uma muita repetição. Está planeado, no entanto, tornar possível deixe esta parte levar em conta o material sobre lógica proposicional (e excluir o material já abordado, bem como encurtar as provas com referências aos respectivos lugares no proposicional parte).

Capítulo 14

Introdução à Lógica de Primeira Ordem

`content/first-order-logic/introduction/first-order-logic.tex`

14.1 Lógica de Primeira Ordem

fol:int:fol:
sec Você provavelmente já conhece a lógica de primeira ordem da sua primeira introdução à lógica formal.¹ Pode conhecê-la como “lógica quantificacional” ou “lógica de predicados”. A lógica de primeira ordem, antes de tudo, é uma

¹De fato, mais ou menos assumimos que sim! Se não conhece, pode revisar um livro-texto mais elementar, como *forall x* (Magnus et al., 2021).

linguagem formal. Isso significa que ela tem um certo vocabulário, e suas expressões são cadeias desse vocabulário. Mas nem toda cadeia é permitida. Há diferentes tipos de expressões permitidas: termos, fórmulas e sentenças. Estamos principalmente interessados em sentenças da lógica de primeira ordem: elas nos fornecem um análogo formal de sentenças do inglês, e sobre elas podemos fazer as perguntas que um lógico tipicamente se interessa. Por exemplo:

- ψ segue logicamente de φ ?
- φ é logicamente verdadeira, logicamente falsa ou contingente?
- φ e ψ são equivalentes?

Essas perguntas são primariamente perguntas sobre o “significado” das sentenças da lógica de primeira ordem. Por exemplo, um filósofo analisaria a pergunta de se ψ segue logicamente de φ como perguntando: há um caso em que φ é verdadeira mas ψ é falsa (ψ não segue de φ), ou todo caso que torna φ verdadeira também torna ψ verdadeira (ψ segue de φ)? Mas ainda não nos foi dito o que é um “caso”—esse é o trabalho da *semântica*. A semântica da lógica de primeira ordem fornece um modelo matematicamente preciso da ideia intuitiva do filósofo de “caso”, e também—e isso é importante—do que significa uma sentença φ ser verdadeira em um caso. Chamamos o modelo matematicamente preciso que desenvolveremos de *a estrutura*. A relação que torna “verdadeira em” precisa é chamada de relação de *satisfação*. Assim, o que definiremos é “ φ é satisfeita em $\mathfrak{M}$ ” (em símbolos: $\mathfrak{M} \models \varphi$) para sentenças φ e estruturas $\mathfrak{M}$. Uma vez feito isso, também podemos dar definições precisas dos outros termos semânticos como “segue de” ou “é logicamente verdadeira”. Essas definições tornarão possível decidir, novamente com precisão matemática, se, por exemplo, $\forall x (\varphi(x) \rightarrow \psi(x)), \exists x \varphi(x) \models \exists x \psi(x)$. A resposta será, é claro, “sim”. Se você já foi treinado para simbolizar sentenças do inglês na lógica de primeira ordem, reconhecerá isto como, por exemplo, a simbolização de “Todas as formigas são insetos, há formigas, portanto há insetos.” Isso é obviamente um argumento válido, e assim nosso modelo matemático de “segue de” para nossa linguagem formal deve dar a mesma resposta.

Outro tópico que você provavelmente lembra da sua primeira introdução à lógica formal é que há *derivações*. Se você fez um primeiro curso de lógica formal, seu professor o fez praticar a encontrar tais *derivações*, talvez até uma *derivação* que mostra que a consequência acima vale. Há muitas maneiras diferentes de dar *derivações*: você pode ter feito algo chamado “dedução natural” ou “árvores de verdade”, mas há muitas outras. O propósito dos sistemas de *derivação* é fornecer ferramentas com as quais as perguntas dos lógicos acima podem ser respondidas: por exemplo, uma *derivação* de dedução natural em que $\forall x (\varphi(x) \rightarrow \psi(x))$ e $\exists x \varphi(x)$ são premissas e $\exists x \psi(x)$ é a conclusão (última linha) *verifica* que $\exists x \psi(x)$ segue logicamente de $\forall x (\varphi(x) \rightarrow \psi(x))$ e $\exists x \varphi(x)$.

Mas por quê? À primeira vista, os sistemas de *derivação* não têm nada a ver com semântica: dar uma *derivação* formal envolve apenas dispor símbolos de certas maneiras regidas por regras; eles não mencionam “casos” ou “verdadeira

em” de forma alguma. A conexão entre sistemas de **derivação** e semântica tem de ser estabelecida por uma investigação metalógica. O que é necessário é uma prova matemática, por exemplo, de que uma **derivação** formal de $\exists x \psi(x)$ a partir das premissas $\forall x (\varphi(x) \rightarrow \psi(x))$ e $\exists x \varphi(x)$ é possível se, e somente se, $\exists x \psi(x)$ é consequência de $\forall x (\varphi(x) \rightarrow \psi(x))$ e $\exists x \varphi(x)$ em conjunto. Antes que isso possa ser feito, entretanto, muito trabalho minucioso tem de ser realizado para acertar as definições de sintaxe e semântica.

content/first-order-logic/introduction/syntax.tex

14.2 Sintaxe

fol:int:syn:
sec

Primeiro devemos tornar preciso quais cadeias de símbolos contam como **sentenças** da lógica de primeira ordem. Faremos isso mais adiante; por enquanto, procederemos por exemplos. Os blocos de construção básicos—o vocabulário—da lógica de primeira ordem dividem-se em duas partes. A primeira parte são os símbolos que usamos para dizer coisas específicas ou para destacar coisas específicas. Destacamos coisas usando **símbolos de constante**, e dizemos coisas sobre o que destacamos usando **símbolos de predicado**. Por exemplo, poderíamos usar a como **a símbolo de constante** para destacar uma única coisa, e então dizer algo sobre ela usando **a sentença** $P(a)$. Se você tem significados para “ a ” e “ P ” em mente, pode ler $P(a)$ como uma sentença do inglês (e provavelmente já fez isso quando aprendeu lógica formal pela primeira vez). Uma vez que se tem tais **sentenças** simples da lógica de primeira ordem, pode-se construir outras mais complexas usando a segunda parte do vocabulário: os símbolos lógicos (conectivos e quantificadores). Assim, por exemplo, podemos formar expressões como $(P(a) \wedge Q(b))$ ou $\exists x P(x)$.

Para fornecer as definições precisas de semântica e as regras dos nossos sistemas de **derivação** necessárias para o estudo meta-lógico rigoroso, temos primeiro que dar uma definição precisa do que conta como **a sentença** da lógica de primeira ordem. A ideia básica é fácil o suficiente de entender: há algumas **sentenças** simples que podemos formar apenas a partir de **símbolos de predicado** e **símbolos de constante**, como $P(a)$. E então a partir destas formamos outras mais complexas usando os conectivos e quantificadores. Mas quais são exatamente as regras pelas quais nos é permitido formar **sentenças** mais complexas? Elas devem ser especificadas; caso contrário, não definimos “**sentença** da lógica de primeira ordem” com precisão suficiente. Há algumas questões. A primeira é obter as cadeias certas para contarem como **sentenças**. A segunda é fazê-lo de tal forma que possamos dar provas matemáticas sobre *todas* as **sentenças**. Finalmente, teremos também que dar definições precisas de algumas operações rudimentares com **sentenças**, como “substituir todo x em φ por b .” O problema é que os quantificadores e **variáveis** que temos na lógica de primeira ordem tornam não inteiramente óbvio como isso deve ser feito. Por exemplo, $\exists x P(a)$ deve contar como **a sentença**? E quanto a $\exists x \exists x P(x)$? Qual deveria ser o resultado de “substituir x por b em $(P(x) \wedge \exists x P(x))$ ”?

14.3 Fórmulas

Eis a abordagem que usaremos para especificar rigorosamente as *sentenças* da lógica de primeira ordem e para lidar com as questões decorrentes do uso de *variáveis*. Primeiro definimos um conjunto *diferente* de expressões: *fórmulas*. Uma vez feito isso, podemos considerar o papel que as *variáveis* desempenham nelas—e, com base em algumas outras ideias, nomeadamente as de *variáveis* “livres” e “ligadas”, podemos definir o que é uma *sentença* (nomeadamente, uma *fórmula* sem *variáveis* livres). Fazemos isso não apenas porque torna a definição de “*sentença*” mais gerenciável, mas também porque será crucial para a maneira como definimos a noção semântica de satisfação.

Vamos definir “*fórmula*” para uma linguagem simples de primeira ordem, contendo apenas um único *símbolo de predicado* P e um único *símbolo de constante* a , e apenas os símbolos lógicos $\neg$, $\wedge$ e $\exists$. Nossas definições completas serão muito mais gerais: permitiremos infinitos *símbolos de predicado* e *símbolos de constante*. De fato, também consideraremos *símbolos de função* que podem ser combinados com *símbolos de constante* e *variáveis* para formar “termos”. Por enquanto, a e as *variáveis* serão nossos únicos termos. Precisamos de infinitas *variáveis*. Oficialmente usaremos os símbolos $v_0, v_1, \dots$, como *variáveis*.

Definição 14.1. O conjunto de *fórmulas* Frm é definido da seguinte forma:

1. $P(a)$ e $P(v_i)$ são *fórmulas* ($i \in \mathbb{N}$). fol:int:fml:
fmls-atom
2. Se φ é uma *fórmula*, então $\neg\varphi$ é uma *fórmula*. fol:int:fml:
fmls-not
3. Se φ e ψ são *fórmulas*, então $(\varphi \wedge \psi)$ é uma *fórmula*.
4. Se φ é uma *fórmula* e x é uma *variável*, então $\exists x \varphi$ é uma *fórmula*. fol:int:fml:
fmls-ex
5. Nada mais é uma *fórmula*. fol:int:fml:
fmls-limit

(1) nos diz que $P(a)$ e $P(v_i)$ são *fórmulas*, para qualquer $i \in \mathbb{N}$. Estas são as chamadas *fórmulas atômicas*. Elas nos dão algo com que começar. As outras cláusulas nos dão maneiras de formar novas *fórmulas* a partir daquelas que já formamos. Assim, por exemplo, por (2), obtemos que $\neg P(v_2)$ é uma *fórmula*, já que $P(v_2)$ já é uma *fórmula* por (1). Então, por (4), obtemos que $\exists v_2 \neg P(v_2)$ é outra *fórmula*, e assim por diante. (5) nos diz que *somente* cadeias que podemos formar desta maneira contam como *fórmulas*. Em particular, $\exists v_0 P(a)$ e $\exists v_0 \exists v_0 P(a)$ *contam* como *fórmulas*, e $(\neg P(a))$ não conta, por causa dos parênteses externos supérfluos.

Esta maneira de definir *fórmulas* é chamada de *definição indutiva*, e nos permite provar coisas sobre *fórmulas* usando uma versão de prova por indução chamada *indução estrutural*. Estas são discutidas de maneira geral em [Seção 75.4](#)

14.4. SATISFAÇÃO

e [Seção 75.5](#), que você deve revisar antes de mergulhar nas provas mais adiante. Basicamente, a ideia é que, se você quer dar uma prova de que algo é verdadeiro para todas as [fórmulas](#), mostra primeiro que é verdadeiro para as [fórmulas](#) atômicas e então que se for verdadeiro para qualquer [fórmula](#) φ (e ψ), *também* é verdadeiro para $\neg\varphi$, $(\varphi \wedge \psi)$ e $\exists x\varphi$. Por exemplo, isso prova que é verdadeiro para $\exists v_2 \neg P(v_2)$: da primeira parte você sabe que é verdadeiro para a [fórmula](#) atômica $P(v_2)$. Então obtém que é verdadeiro para $\neg P(v_2)$ pela segunda parte, e novamente que é verdadeiro para $\exists v_2 \neg P(v_2)$ em si. Como todas as [fórmulas](#) são geradas indutivamente a partir de [fórmulas](#) atômicas, isso funciona para qualquer uma delas.

content/first-order-logic/introduction/satisfaction.tex

14.4 Satisfação

fol:int:sat:
sec

Já podemos avançar para a semântica da lógica de primeira ordem assim que sabemos o que são [fórmulas](#): aqui, a definição básica é a de [a estrutura](#). Para nossa linguagem simples, uma [estrutura](#) $\mathfrak{M}$ tem apenas três componentes: um conjunto não vazio $|\mathfrak{M}|$ chamado de [domínio](#), o que a designa em $\mathfrak{M}$, e do que P é verdadeiro em $\mathfrak{M}$. O objeto designado por a é denotado por $a^{\mathfrak{M}}$ e o conjunto de coisas das quais P é verdadeiro por $P^{\mathfrak{M}}$. A [estrutura](#) $\mathfrak{M}$ consiste apenas nestas três coisas: $|\mathfrak{M}|$, $a^{\mathfrak{M}} \in |\mathfrak{M}|$ e $P^{\mathfrak{M}} \subseteq |\mathfrak{M}|$. O caso geral será mais complicado, pois haverá muitos [símbolos de predicado](#) e [símbolos de constante](#), os [símbolos de constante](#) podem ter mais de um lugar, e haverá também [símbolos de função](#).

Isso basta para dar uma definição de satisfação para [fórmulas](#) que não contêm [variáveis](#). A ideia é dar uma definição indutiva que espelha a maneira como definimos [fórmulas](#). Especificamos quando uma fórmula atômica é satisfeita em $\mathfrak{M}$, e então quando, por exemplo, $\neg\varphi$ é satisfeita em $\mathfrak{M}$ com base em se φ é ou não satisfeita em $\mathfrak{M}$. Por exemplo, poderíamos definir:

1. $P(a)$ é satisfeita em $\mathfrak{M}$ se, e somente se, $a^{\mathfrak{M}} \in P^{\mathfrak{M}}$.
2. $\neg\varphi$ é satisfeita em $\mathfrak{M}$ se, e somente se, φ não é satisfeita em $\mathfrak{M}$.
3. $(\varphi \wedge \psi)$ é satisfeita em $\mathfrak{M}$ se, e somente se, φ é satisfeita em $\mathfrak{M}$ e ψ também é satisfeita em $\mathfrak{M}$.

Digamos que $|\mathfrak{M}| = \{0, 1, 2\}$, $a^{\mathfrak{M}} = 1$ e $P^{\mathfrak{M}} = \{1, 2\}$. Esta definição nos diria que $P(a)$ é satisfeita em $\mathfrak{M}$ (pois $a^{\mathfrak{M}} = 1 \in \{1, 2\} = P^{\mathfrak{M}}$). Nos diria ainda que $\neg P(a)$ não é satisfeita em $\mathfrak{M}$, e que, por sua vez, $\neg\neg P(a)$ é, e que $(\neg P(a) \wedge P(a))$ não é satisfeita, e assim por diante.

A dificuldade surge quando queremos dar uma definição para os quantificadores: gostaríamos de dizer algo como: “ $\exists v_0 P(v_0)$ é satisfeita se, e somente se, $P(v_0)$ é satisfeita.” Mas a [estrutura](#) $\mathfrak{M}$ não nos diz o que fazer com [variáveis](#). O que realmente queremos dizer é que $P(v_0)$ é satisfeita *para algum valor de* v_0 . Para tornar isso preciso, precisamos de uma maneira de atribuir

membros de $|\mathfrak{M}|$ não apenas a a , mas também a v_0 . Para isso, introduzimos *atribuições de variáveis*. Uma atribuição de variáveis é simplesmente uma função s que mapeia **variáveis** para **membros** de $|\mathfrak{M}|$ (em nosso exemplo, para um entre 1, 2 ou 3). Como não sabemos de antemão quais **variáveis** podem aparecer em uma **fórmula**, não podemos limitar a quais **variáveis** s atribui valores. A solução simples é exigir que s atribua valores a *todas* as **variáveis** $v_0, v_1, \dots$. Usaremos apenas aquelas de que precisarmos.

Em vez de definir a satisfação de **fórmulas** apenas em relação a uma **estrutura**, definiremos em relação a uma **estrutura** $\mathfrak{M}$ e uma atribuição de variáveis s , e escreveremos $\mathfrak{M}, s \models \varphi$ por brevidade. Nossa definição agora incluirá uma cláusula adicional para lidar com **fórmulas** atômicas contendo **variáveis**:

1. $\mathfrak{M}, s \models P(a)$ se, e somente se, $a^{\mathfrak{M}} \in P^{\mathfrak{M}}$.
2. $\mathfrak{M}, s \models P(v_i)$ se, e somente se, $s(v_i) \in P^{\mathfrak{M}}$.
3. $\mathfrak{M}, s \models \neg\varphi$ se, e somente se, não $\mathfrak{M}, s \models \varphi$.
4. $\mathfrak{M}, s \models (\varphi \wedge \psi)$ se, e somente se, $\mathfrak{M}, s \models \varphi$ e $\mathfrak{M}, s \models \psi$.

Ok, isso resolve um problema: agora podemos dizer quando $\mathfrak{M}$ satisfaz $P(v_0)$ para o valor $s(v_0)$. Para obter a definição correta para $\exists v_0 P(v_0)$ temos que fazer mais uma coisa: queremos que $\mathfrak{M}, s \models \exists v_0 P(v_0)$ se, e somente se, $\mathfrak{M}, s' \models P(v_0)$ para *alguma* maneira s' de atribuir um valor a v_0 . Mas o valor atribuído a v_0 não precisa necessariamente ser o valor que $s(v_0)$ escolhe. Introduziremos uma notação para isso: se $m \in |\mathfrak{M}|$, então deixamos $s[m/v_0]$ ser a atribuição que é igual a s (para todas as **variáveis** diferentes de v_0), exceto que a v_0 atribui m . Agora nossa definição pode ser:

5. $\mathfrak{M}, s \models \exists v_i \varphi$ se, e somente se, $\mathfrak{M}, s[m/v_i] \models \varphi$ para algum $m \in |\mathfrak{M}|$.

Funciona? Digamos que $s(v_i) = 0$ para todo $i \in \mathbb{N}$. $\mathfrak{M}, s \models \exists v_0 P(v_0)$ se, e somente se, há um $m \in |\mathfrak{M}|$ tal que $\mathfrak{M}, s[m/v_0] \models P(v_0)$. E há: podemos escolher $m = 1$ ou $m = 2$. Note que isso é verdadeiro mesmo que o valor $s(v_0)$ atribuído a v_0 por s —neste caso, 0—não faça o trabalho. Temos $\mathfrak{M}, s[1/v_0] \models P(v_0)$, mas não $\mathfrak{M}, s \models P(v_0)$.

Se isso parece confuso e trabalhoso: é. Mas a complexidade adicional é necessária para dar uma definição precisa e indutiva de satisfação para todas as **fórmulas**, e precisamos de algo assim para definir precisamente as noções semânticas. Há outras maneiras de fazer isso, mas todas são igualmente (in)elegantes.

content/first-order-logic/introduction/sentences.tex

14.5 Sentenças

fol:int:snt:
sec

14.6. NOÇÕES SEMÂNTICAS

Ok, agora temos uma (esboço de uma) definição de satisfação (“verdadeira em”) para **estruturas** e **fórmulas**. Mas isso requer um ingrediente adicional—uma atribuição de **variável**—e o que queríamos era uma definição de **sentenças**. Como nos livramos das atribuições, e o que são **sentenças**?

Você provavelmente se lembra de uma discussão, em sua primeira introdução à lógica formal, sobre a relação entre **variáveis** e quantificadores. Um quantificador é sempre seguido por **a variável** e, então, na parte da **sentença** à qual esse quantificador se aplica (seu “escopo”), entendemos que a **variável** está “ligada” a esse quantificador. Em **fórmulas**, não se exigia que toda **variável** tivesse um quantificador correspondente, e as **variáveis** sem quantificadores correspondentes são “livres” ou “não ligadas”. Tomaremos as **sentenças** como sendo todas aquelas **fórmulas** que não têm **variáveis** livres.

Novamente, a ideia intuitiva de quando uma ocorrência de **a variável** em **a fórmula** φ está ligada, de qual quantificador a liga e de quando ela é livre não é difícil de captar. Você pode ter aprendido um método para testar isso, talvez envolvendo a contagem de parênteses. Temos que insistir em uma definição precisa—e, como definimos **fórmulas** por indução, podemos dar uma definição das ocorrências livres e ligadas de **a variável** x em **a fórmula** φ também por indução. Por exemplo, ela poderia ter este aspecto para nossa linguagem simplificada:

1. Se φ é atômica, todas as ocorrências de x nela são livres (isto é, a ocorrência de x em $P(x)$ é livre).
2. Se φ é da forma $\neg\psi$, então uma ocorrência de x em $\neg\psi$ é livre se, e somente se, a ocorrência correspondente de x for livre em ψ (isto é, as ocorrências livres de variáveis em ψ são exatamente as ocorrências correspondentes em $\neg\psi$).
3. Se φ é da forma $(\psi \wedge \chi)$, então uma ocorrência de x em $(\psi \wedge \chi)$ é livre se, e somente se, a ocorrência correspondente de x for livre em ψ ou em χ .
4. Se φ é da forma $\exists x \psi$, então nenhuma ocorrência de x em φ é livre; se é da forma $\exists y \psi$, em que y é uma **variável** diferente de x , então uma ocorrência de x em $\exists y \psi$ é livre se, e somente se, a ocorrência correspondente de x for livre em ψ .

Uma vez que tenhamos uma definição precisa de ocorrências livres e ligadas de variáveis, podemos simplesmente dizer: **a sentença** é qualquer **fórmula** sem ocorrências livres de **variáveis**.

`content/first-order-logic/introduction/semantic-notions.tex`

14.6 Noções Semânticas

fol:int:sem:
sec Mencionamos acima que, quando consideramos se $\mathfrak{M}, s \models \varphi$ vale, (por conveniência) deixamos s atribuir valores a todas as **variáveis**, mas apenas os

valores que atribui às **variáveis** em φ são usados. De fato, são apenas os valores das **variáveis livres** em φ que importam. É claro que, por sermos cuidadosos, vamos provar esse fato. Como as **sentenças** não têm variáveis livres, s não importa de modo algum quando se trata de saber se são satisfeitas em **a estrutura**. Assim, quando φ é **a sentença**, podemos definir $\mathfrak{M} \models \varphi$ como significando “ $\mathfrak{M}, s \models \varphi$ para todo s ,” o que, aliás, é verdadeiro se, e somente se, $\mathfrak{M}, s \models \varphi$ para pelo menos um s . Precisamos introduzir atribuições de variáveis para obter uma definição funcional de satisfação para **fórmulas**, mas para **sentenças**, a satisfação é independente das atribuições de variáveis.

Uma vez que temos uma definição de “ $\mathfrak{M} \models \varphi$,” sabemos o que “caso” e “verdadeira em” significam no que diz respeito às **sentenças** da lógica de primeira ordem. Com base na definição de $\mathfrak{M} \models \varphi$ para **sentenças**, podemos então definir as noções semânticas básicas de validade, consequência e satisfatibilidade. Uma sentença é válida, $\models \varphi$, se toda **estrutura** a satisfaz. Ela é consequência de um conjunto de **sentenças**, $\Gamma \models \varphi$, se toda **estrutura** que satisfaz todas as **sentenças** em Γ também satisfaz φ . E um conjunto de **sentenças** é satisfatível se alguma **estrutura** satisfaz todas as **sentenças** nele ao mesmo tempo.

Como as **fórmulas** são definidas indutivamente, e a satisfação é, por sua vez, definida por indução na estrutura das **fórmulas**, podemos usar indução para provar propriedades da nossa semântica e relacionar as noções semânticas definidas. Vamos reunir e provar algumas dessas propriedades, em parte porque são individualmente interessantes, mas principalmente porque muitas delas serão úteis quando formos investigar a relação entre semântica e sistemas de **derivação**. Para isso, também teremos que definir (precisamente, isto é, por indução) algumas noções sintáticas e operações que ainda não mencionamos.

content/first-order-logic/introduction/substitution.tex

14.7 Substituição

Discutiremos um exemplo para ilustrar como as coisas se articulam e como o desenvolvimento da sintaxe e da semântica estabelece as bases para nossas investigações mais avançadas posteriormente. Nossos sistemas de **derivação** deveriam permitir-nos derivar $P(a)$ de $\forall v_0 P(v_0)$. Talvez até queiramos afirmar isso como uma regra de inferência. No entanto, para fazê-lo, devemos ser capazes de enunciá-lo nos termos mais gerais: não apenas para P , a e v_0 , mas para qualquer **fórmula** φ , e termo t , e **variável** x . (Lembre-se de que **símbolos de constante** são termos, mas vamos considerar também termos mais complicados construídos a partir de **símbolos de constante** e **símbolos de função**.) Então, queremos ser capazes de dizer algo como, “sempre que você tiver derivado $\forall x \varphi(x)$ você está justificado ao inferir $\varphi(t)$ —o resultado da remoção de $\forall x$ e substituindo x por t .” Mas o que exatamente significa “substituir x por t ”? Qual é a relação entre $\varphi(x)$ e $\varphi(t)$? Isso sempre funciona?

Para tornar isso preciso, definimos a operação de **substituição**. A substituição é realmente complicada, porque não podemos simplesmente substituir

fol:int:sub:
sec

14.8. MODELOS E TEORIAS

todos os x em φ por t , e nem todos os t podem ser substituídos por qualquer x . Lidaremos com isso novamente usando definições indutivas. Mas uma vez feito isso, especificando uma regra de inferência como “inferir $\varphi(t)$ de $\forall x \varphi(x)$ ” torna-se uma definição precisa. Além disso, seremos capazes de mostrar que esta é uma boa regra de inferência no sentido de que $\varphi(t)$ é consequência de $\forall x \varphi(x)$. Mas para provar isso, temos de provar novamente algo que à primeira vista pode levá-lo a perguntar “por que estamos fazendo isso?” Que $\varphi(t)$ seja consequência de $\forall x \varphi(x)$ depende do fato de que se $\mathfrak{M} \models \varphi(t)$ vale ou não depende apenas do valor do termo t , ou seja, se deixarmos m ser qualquer **membro** de $|\mathfrak{M}|$ que for selecionado por t , então $\mathfrak{M}, s \models \varphi(t)$ se, e somente se, $\mathfrak{M}, s[m/x] \models \varphi(x)$. Isso vale mesmo quando t contém **variáveis**, mas teremos que ter cuidado com a forma exata como declaramos o resultado.

`content/first-order-logic/introduction/models-theories.tex`

14.8 Modelos e Teorias

fol:int:mod:
sec

Uma vez que definimos a sintaxe e a semântica da lógica de primeira ordem, podemos trabalhar investigando as propriedades de **estruturas** e as noções semânticas. Também podemos definir sistemas de **derivação** e investigá-los. Para um conjunto de **sentenças**, podemos perguntar: quais **estruturas** tornam todas as **sentenças** desse conjunto verdadeiras? Dado um conjunto de **sentenças** Γ , uma **estrutura** $\mathfrak{M}$ que as satisfaz é chamada de *modelo de Γ* . Podemos começar de Γ e tentar encontrar seus modelos—como eles são? Quão grandes ou pequenos precisam ser? Mas também podemos começar com uma única **estrutura** ou coleção de **estruturas** e perguntar: quais **sentenças** são verdadeiras nelas? Há **sentenças** que *caracterizam* essas **estruturas** no sentido de que elas, e somente elas, são verdadeiras nelas? Esses tipos de questões são o domínio da *teoria dos modelos*. Elas também fundamentam o *método axiomático*: descrever uma coleção de **estruturas** por um conjunto de **sentenças**, os axiomas de uma teoria. Isso é possível pela observação de que exatamente aquelas **sentenças** que são consequência na lógica de primeira ordem dos axiomas são verdadeiras em todos os modelos dos axiomas.

Como exemplo muito simples, considere pré-ordens. Uma pré-ordem é uma relação R sobre algum conjunto A que é reflexiva e transitiva. Um conjunto A com uma relação binária $R \subseteq A \times A$ sobre ele é exatamente o que precisaríamos para dar uma **estrutura** para uma linguagem de primeira ordem com um único símbolo de predicado de dois lugares P : definiríamos $|\mathfrak{M}| = A$ e $P^{\mathfrak{M}} = R$. Como R é uma pré-ordem, é reflexiva e transitiva, e podemos encontrar um conjunto Γ de **sentenças** da lógica de primeira ordem que diz isso:

$$\begin{aligned} & \forall v_0 P(v_0, v_0) \\ & \forall v_0 \forall v_1 \forall v_2 ((P(v_0, v_1) \wedge P(v_1, v_2)) \rightarrow P(v_0, v_2)) \end{aligned}$$

Estas **sentenças** são apenas as simbolizações de “para todo x , Rxx ” (R é reflexiva) e “sempre que Rxy e Ryz , então também Rxz ” (R é transitiva). Vemos

que uma **estrutura** $\mathfrak{M}$ é um modelo dessas duas **sentenças** Γ se, e somente se, R (isto é, $P^{\mathfrak{M}}$), é uma pré-ordem sobre A (isto é, $|\mathfrak{M}|$). Em outras palavras, os modelos de Γ são exatamente as pré-ordens. Qualquer propriedade de todas as pré-ordens que possa ser expressa na linguagem de primeira ordem apenas com P como **símbolo de predicado** (como reflexividade e transitividade acima), é consequência das duas **sentenças** em Γ e vice-versa. Assim, qualquer coisa que possamos provar sobre modelos de Γ provamos sobre todas as pré-ordens.

Para qualquer teoria particular e classe de modelos (como Γ e todas as pré-ordens), haverá questões interessantes sobre o que pode ser expresso na linguagem de primeira ordem correspondente, e o que não pode ser expresso. Há algumas propriedades de **estruturas** que são interessantes para todas as linguagens e classes de modelos, a saber, aquelas que dizem respeito ao tamanho do **domínio**. Pode-se sempre expressar, por exemplo, que o **domínio** contém exatamente n **membros**, para qualquer $n \in \mathbb{Z}^+$. Pode-se também expressar, usando um conjunto de infinitas **sentenças**, que o **domínio** é infinito. Mas não se pode expressar que o domínio é finito, ou que o domínio é **não enumerável**. Estes resultados sobre as limitações das linguagens de primeira ordem são consequências dos teoremas da compacidade e de Löwenheim–Skolem.

`content/first-order-logic/introduction/soundness-completeness.tex`

14.9 Correção e Completude

Também introduziremos sistemas de **derivação** para a lógica de primeira ordem. Há muitos sistemas de **derivação** que os lógicos desenvolveram, mas todos definem a mesma relação de **derivabilidade** entre **sentenças**. Dizemos que Γ **deriva** φ , $\Gamma \vdash \varphi$, se há uma **derivação** de um certo tipo precisamente definido. As **derivações** são sempre disposições finitas de símbolos—talvez uma lista de **sentenças**, ou alguma estrutura mais complicada. O propósito dos sistemas de **derivação** é fornecer uma ferramenta para determinar se uma **sentença** é consequência de algum conjunto Γ . Para servir a esse propósito, deve ser verdade que $\Gamma \models \varphi$ se, e somente se, $\Gamma \vdash \varphi$.

fol:int:scp:
sec

Se $\Gamma \vdash \varphi$ mas não $\Gamma \models \varphi$, nosso sistema de **derivação** seria forte demais, provaria demais. A propriedade de que se $\Gamma \vdash \varphi$ então $\Gamma \models \varphi$ é chamada de **correção**, e é um requisito mínimo para qualquer bom sistema de **derivação**. Por outro lado, se $\Gamma \models \varphi$ mas não $\Gamma \vdash \varphi$, então nosso sistema de **derivação** é fraco demais, não prova o suficiente. A propriedade de que se $\Gamma \models \varphi$ então $\Gamma \vdash \varphi$ é chamada de **completude**. A correção é usualmente relativamente fácil de provar (por indução na estrutura das **derivações**, que são definidas indutivamente). A completude é mais difícil de provar.

Correção e completude têm várias consequências importantes. Se um conjunto de **sentenças** Γ **deriva** uma contradição (como $\varphi \wedge \neg\varphi$), ele é chamado de **inconsistente**. Γ s inconsistentes não podem ter modelos; são insatisfatíveis. Da completude segue a contrapositiva: qualquer Γ que não seja inconsistente—ou, como diremos, **consistente**—tem um modelo. De fato, isso é equivalente à com-

pletude, e é a forma de completude que de fato provaremos. É um resultado profundo e talvez surpreendente: apenas o fato de que você não pode provar $\varphi \wedge \neg\varphi$ a partir de Γ garante que há uma **estrutura** que é como Γ a descreve. Assim, a completude dá uma resposta à pergunta: quais conjuntos de **sentenças** têm modelos? Resposta: todos e somente os conjuntos consistentes têm.

Os teoremas de correção e completude têm duas consequências importantes: a compacidade e o teorema de Löwenheim–Skolem. Estes são resultados importantes na teoria de modelos e podem ser usados para estabelecer muitos resultados interessantes. Já mencionamos dois: a lógica de primeira ordem não pode expressar que o **domínio** de uma **estrutura** é finito ou que é **não enumerável**.

Historicamente, tudo isso—como definir a sintaxe e a semântica da lógica de primeira ordem, como definir bons sistemas de **derivação**, como provar que são corretos e completos, ficar claro sobre o que pode e não pode ser expresso em linguagens de primeira ordem—levou muito tempo para ser descoberto e acertado. Agora sabemos como fazer isso, mas percorrer todos os detalhes ainda pode ser confuso e tedioso. Mas também é importante, porque os métodos desenvolvidos aqui para a linguagem formal da lógica de primeira ordem são aplicados em todo lugar em lógica, ciência da computação e linguística. Assim, trabalhar os detalhes compensa a longo prazo.

Capítulo 15

Sintaxe da Lógica de Primeira Ordem

[content/first-order-logic/syntax-and-semantics/intro-syntax.tex](#)

15.1 Introdução

fol:syn:itx:
sec Para desenvolver a teoria e a metateoria da lógica de primeira ordem, devemos primeiro definir a sintaxe e a semântica de suas expressões. As expressões da lógica de primeira ordem são termos e **fórmulas**. Termos são formados a partir de **variáveis**, **símbolos de constante** e **símbolos de função**. **Fórmulas**, por sua vez, são formadas a partir de **símbolos de predicado** juntamente com

termos (estes formam as menores *fórmulas*, as “atômicas”), e então, a partir de *fórmulas* atômicas, podemos formar outras mais complexas usando conectivos lógicos e quantificadores. Há muitas maneiras diferentes de estabelecer as regras de formação; damos apenas uma possível. Outros sistemas escolherão símbolos diferentes, selecionarão conjuntos diferentes de conectivos como primitivos, usarão parênteses de forma diferente (ou nem mesmo os usarão, como no caso da chamada notação polonesa). O que todas as abordagens têm em comum, no entanto, é que as regras de formação definem o conjunto de termos e *fórmulas* *indutivamente*. Se feito corretamente, cada expressão pode resultar essencialmente de uma única maneira de acordo com as regras de formação. A definição indutiva que resulta em expressões que são *de leitura única* significa que podemos dar significados a essas expressões usando o mesmo método—a definição indutiva.

`content/first-order-logic/syntax-and-semantics/first-order-languages.tex`

15.2 Linguagens de Primeira Ordem

As expressões da lógica de primeira ordem são construídas a partir de um vocabulário básico contendo *variáveis*, *símbolos de constante*, *símbolos de predicado* e às vezes *símbolos de função*. A partir deles, juntamente com conectivos lógicos, quantificadores e símbolos de pontuação como parênteses e vírgulas, formam-se *termos* e *fórmulas*.

fol:syn:fol:
sec

explanation

Informalmente, *símbolos de predicado* são nomes para propriedades e relações, *símbolos de constante* são nomes para objetos individuais, e *símbolos de função* são nomes para mapeamentos. Estes, exceto a *predicado de identidade* $=$, são os *símbolos não lógicos* e juntos constituem uma linguagem. Qualquer linguagem de primeira ordem $\mathcal{L}$ é determinada por seus símbolos não lógicos. No caso mais geral, $\mathcal{L}$ contém infinitos símbolos de cada tipo.

No caso geral, fazemos uso dos seguintes símbolos na lógica de primeira ordem:

1. Símbolos lógicos
 - a) Conectivos lógicos: $\neg$ (negação), $\wedge$ (conjunção), $\vee$ (disjunção), $\rightarrow$ (*condicional*), $\leftrightarrow$ (*bicondicional*), $\forall$ (quantificador universal), $\exists$ (quantificador existencial).
 - b) A constante proposicional para *falsidade* $\perp$.
 - c) A constante proposicional para *verdade* $\top$.
 - d) A *predicado de identidade* de dois lugares $=$.
 - e) Um conjunto *infinito enumerável* de *variáveis*: $v_0, v_1, v_2, \dots$
2. Símbolos não lógicos, que constituem a *linguagem padrão* da lógica de primeira ordem

15.2. LINGUAGENS DE PRIMEIRA ORDEM

- a) Um conjunto infinito enumerável de símbolos de predicado de n lugares para cada $n > 0$: $A_0^n, A_1^n, A_2^n, \dots$
 - b) Um conjunto infinito enumerável de símbolos de constante: $c_0, c_1, c_2, \dots$
 - c) Um conjunto infinito enumerável de símbolos de função de n lugares para cada $n > 0$: $f_0^n, f_1^n, f_2^n, \dots$
3. Sinais de pontuação: $(,)$ e a vírgula.

A maioria de nossas definições e resultados será formulada para a linguagem padrão completa da lógica de primeira ordem. No entanto, dependendo da aplicação, também podemos restringir a linguagem a apenas alguns símbolos de predicado, símbolos de constante e símbolos de função.

Exemplo 15.1. A linguagem $\mathcal{L}_A$ da aritmética contém um único símbolo de predicado de dois lugares $<$, um único símbolo de constante 0 , um símbolo de função de um lugar $!$, e dois símbolos de função de dois lugares $+$ e $\times$.

Exemplo 15.2. A linguagem da teoria dos conjuntos $\mathcal{L}_Z$ contém apenas o único símbolo de predicado de dois lugares $\in$.

Exemplo 15.3. A linguagem das ordens $\mathcal{L}_{\leq}$ contém apenas o símbolo de predicado de dois lugares $\leq$.

Novamente, estas são convenções: oficialmente, estes são apenas apelidos, por exemplo, $<$, $\in$ e $\leq$ são apelidos para A_0^2 , 0 para c_0 , $!$ para f_0^1 , $+$ para f_0^2 , $\times$ para f_1^2 .

Você pode estar familiarizado com terminologia e símbolos diferentes dos que usamos acima. Textos de lógica (e professores) comumente usam $\sim$, $\neg$, ou $!$ para “negação”, $\wedge$, $\cdot$, ou $\&$ para “conjunção”. Símbolos comumente usados para o “condicional” ou “implicação” são $\rightarrow$, $\Rightarrow$, e $\supset$. Símbolos para “bicondicional”, “bi-implicação”, ou “equivalência (material)” são $\leftrightarrow$, $\Leftrightarrow$, e $\equiv$. O símbolo $\perp$ é chamado de várias formas: “falsidade”, “falsum”, “absurdo”, ou “bottom”. O símbolo $\top$ é chamado de várias formas: “verdade”, “verum”, ou “top”.

É convencional usar letras minúsculas (por exemplo, a , b , c) do início do alfabeto latino para símbolos de constante (às vezes chamados de nomes), e letras minúsculas do final (por exemplo, x , y , z) para variáveis. Quantificadores combinam-se com variáveis, por exemplo, x ; variações notacionais incluem $\forall x$, $(\forall x)$, (x) , Πx , $\bigwedge_x$ para o quantificador universal e $\exists x$, $(\exists x)$, (Ex) , Σx , $\bigvee_x$ para o quantificador existencial.

Poderíamos tratar todos os operadores proposicionais e ambos os quantificadores como símbolos primitivos da linguagem. Poderíamos, em vez disso, escolher um estoque menor de símbolos primitivos e tratar os outros operadores lógicos como definidos. Conjuntos “funcionalmente completos para a verdade” de operadores booleanos incluem $\{\neg, \vee\}$, $\{\neg, \wedge\}$, e $\{\neg, \rightarrow\}$ —estes podem ser

combinados com qualquer um dos quantificadores para uma linguagem de primeira ordem expressivamente completa.

Você pode estar familiarizado com dois outros **operadores lógicos**: a barra de Sheffer $|$ (em homenagem a Henry Sheffer), e a seta de Peirce $\downarrow$, também conhecida como adaga de Quine. Quando recebem suas leituras usuais de “nand” e “nor” (respectivamente), esses operadores são funcionalmente completos para a verdade por si sós.

`content/first-order-logic/syntax-and-semantics/terms-formulas.tex`

15.3 Termos e Fórmulas

Uma vez dada uma linguagem de primeira ordem $\mathcal{L}$, podemos definir expressões construídas a partir do vocabulário básico de $\mathcal{L}$. Estas incluem, em particular, *termos* e *fórmulas*. fol:syn:frm:
sec

Definição 15.4 (Termos). O conjunto de *termos* $\text{Trm}(\mathcal{L})$ de $\mathcal{L}$ é definido indutivamente por: fol:syn:frm:
defn:terms

1. Toda **variável** é um termo.
2. Todo **símbolo de constante** de $\mathcal{L}$ é um termo.
3. Se f é um **símbolo de função** de n lugares e $t_1, \dots, t_n$ são termos, então $f(t_1, \dots, t_n)$ é um termo.
4. Nada mais é um termo.

Um termo que não contém **variáveis** é um *termo fechado*.

explanation Os **símbolos de constante** aparecem em nossa especificação da linguagem e dos termos como uma categoria separada de símbolos, mas eles poderiam, em vez disso, ter sido incluídos como **símbolos de função** de zero lugares. Poderíamos então dispensar a segunda cláusula da definição de termos. Bastaria entender $f(t_1, \dots, t_n)$ como apenas f sozinho se $n = 0$.

Definição 15.5 (Fórmulas). O conjunto de *fórmulas* $\text{Frm}(\mathcal{L})$ da linguagem $\mathcal{L}$ é definido indutivamente como segue: fol:syn:frm:
defn:formulas

1. $\perp$ é uma **fórmula** atômica.
2. $\top$ é uma **fórmula** atômica.
3. Se R é um **símbolo de predicado** de n lugares de $\mathcal{L}$ e $t_1, \dots, t_n$ são termos de $\mathcal{L}$, então $R(t_1, \dots, t_n)$ é uma **fórmula** atômica.
4. Se t_1 e t_2 são termos de $\mathcal{L}$, então $=(t_1, t_2)$ é uma **fórmula** atômica.
5. Se φ é **a fórmula**, então $\neg\varphi$ é **a fórmula**.

15.3. TERMOS E FÓRMULAS

6. Se φ e ψ são fórmulas, então $(\varphi \wedge \psi)$ é a fórmula.
7. Se φ e ψ são fórmulas, então $(\varphi \vee \psi)$ é a fórmula.
8. Se φ e ψ são fórmulas, então $(\varphi \rightarrow \psi)$ é a fórmula.
9. Se φ e ψ são fórmulas, então $(\varphi \leftrightarrow \psi)$ é a fórmula.
10. Se φ é a fórmula e x é a variável, então $\forall x \varphi$ é a fórmula.
11. Se φ é a fórmula e x é a variável, então $\exists x \varphi$ é a fórmula.
12. Nada mais é a fórmula.

As definições do conjunto de termos e do conjunto de fórmulas são *definições indutivas*. Essencialmente, construímos o conjunto de fórmulas em infinitos estágios. No estágio inicial, declaramos que todas as fórmulas atômicas são fórmulas; isso corresponde aos primeiros casos da definição, isto é, os casos para $\top$, $\perp$, $R(t_1, \dots, t_n)$ e $=(t_1, t_2)$. “Fórmula atômica” significa, portanto, qualquer fórmula desta forma.

Os outros casos da definição dão regras para construir novas fórmulas a partir de fórmulas já construídas. No segundo estágio, podemos usá-los para construir fórmulas a partir de fórmulas atômicas. No terceiro estágio, construímos novas fórmulas a partir das fórmulas atômicas e daquelas obtidas no segundo estágio, e assim por diante. Uma fórmula é qualquer coisa que seja eventualmente construída em tal estágio, e nada mais.

Por convenção, escrevemos $=$ entre seus argumentos e omitimos os parênteses: $t_1 = t_2$ é uma abreviação para $=(t_1, t_2)$. Além disso, $\neg=(t_1, t_2)$ é abreviado como $t_1 \neq t_2$. Ao escrever uma fórmula $(\psi * \chi)$ construída a partir de ψ , χ usando um conectivo de dois lugares $*$, frequentemente omitiremos o par de parênteses mais externo e escreveremos simplesmente $\psi * \chi$.

Alguns textos de lógica exigem que a variável x deva ocorrer em φ para que $\exists x \varphi$ e $\forall x \varphi$ contem como fórmulas. Nada de ruim acontece se você não exigir isso, e isso torna as coisas mais fáceis.

Se trabalhamos em uma linguagem para uma aplicação específica, frequentemente escreveremos *símbolos de predicado* e *símbolos de função* de dois lugares entre os respectivos termos, por exemplo, $t_1 < t_2$ e $(t_1 + t_2)$ na linguagem da aritmética e $t_1 \in t_2$ na linguagem da teoria dos conjuntos. A função sucessora na linguagem da aritmética é até escrita convencionalmente *depois* de seu argumento: t' . Oficialmente, no entanto, estas são apenas abreviações convencionais para $A_0^2(t_1, t_2)$, $f_0^2(t_1, t_2)$, $A_0^2(t_1, t_2)$ e $f_0^1(t)$, respectivamente.

Definição 15.6 (Identidade sintática). O símbolo $\equiv$ expressa a identidade sintática entre cadeias de símbolos, isto é, $\varphi \equiv \psi$ sse φ e ψ são cadeias de símbolos do mesmo comprimento e que contêm o mesmo símbolo em cada posição.

CAPÍTULO 15. SINTAXE DA LÓGICA DE PRIMEIRA ORDEM

O símbolo $\equiv$ pode ser ladeado por cadeias obtidas por concatenação, por exemplo, $\varphi \equiv (\psi \vee \chi)$ significa: a cadeia de símbolos φ é a mesma cadeia que aquela obtida concatenando um parêntese de abertura, a cadeia ψ , o símbolo $\vee$, a cadeia χ , e um parêntese de fechamento, nesta ordem. Se este é o caso, então sabemos que o primeiro símbolo de φ é um parêntese de abertura, φ contém ψ como uma subcadeia (começando no segundo símbolo), essa subcadeia é seguida por $\vee$, etc.

Como termos e fórmulas são construídos a partir de elementos básicos via definições indutivas, podemos usar os seguintes princípios de indução para provar coisas sobre eles.

Lema 15.7 (Princípio de indução sobre termos). *Seja $\mathcal{L}$ uma linguagem de primeira ordem. Se uma propriedade P é tal que* fol:syn:frm:
lem:trmind

1. *ela vale para toda variável v ,*
2. *ela vale para todo símbolo de constante a de $\mathcal{L}$, e*
3. *ela vale para $f(t_1, \dots, t_n)$ sempre que vale para $t_1, \dots, t_n$ e f é um símbolo de função de n lugares de $\mathcal{L}$*

(supondo que $t_1, \dots, t_n$ são termos de $\mathcal{L}$), então P vale para todo termo em $\text{Trm}(\mathcal{L})$.

Problema 15.1. Prove Lema 15.7.

Lema 15.8 (Princípio de indução sobre fórmulas). *Seja $\mathcal{L}$ uma linguagem de primeira ordem. Se uma propriedade P vale para todas as fórmulas atômicas e é tal que* fol:syn:frm:
thm:frmind

1. *ela vale para $\neg\varphi$ sempre que vale para φ ;*
2. *ela vale para $(\varphi \wedge \psi)$ sempre que vale para φ e ψ ;*
3. *ela vale para $(\varphi \vee \psi)$ sempre que vale para φ e ψ ;*
4. *ela vale para $(\varphi \rightarrow \psi)$ sempre que vale para φ e ψ ;*
5. *ela vale para $(\varphi \leftrightarrow \psi)$ sempre que vale para φ e ψ ;*
6. *ela vale para $\exists x \varphi$ sempre que vale para φ ;*
7. *ela vale para $\forall x \varphi$ sempre que vale para φ ;*

(supondo que φ e ψ são fórmulas de $\mathcal{L}$), então P vale para todas as fórmulas em $\text{Frm}(\mathcal{L})$.

15.4. LEITURA ÚNICA

15.4 Leitura Única

fol:syn:unq:
sec

A maneira como definimos **fórmulas** garante que toda **fórmula** tem uma *leitura única*, isto é, há essencialmente apenas uma maneira de construí-la de acordo com nossas regras de formação para **fórmulas** e apenas uma maneira de “interpretá-la”. Se não fosse assim, teríamos **fórmulas** ambíguas, isto é, **fórmulas** que têm mais de uma leitura ou interpretação—e isso é claramente algo que queremos evitar. Mas, mais importante, sem essa propriedade, a maioria das definições e provas que vamos dar não funcionaria.

explanation

Talvez a melhor maneira de deixar isso claro seja ver o que aconteceria se tivéssemos dado regras ruins para formar **fórmulas** que não garantissem a leitura única. Por exemplo, poderíamos ter esquecido os parênteses nas regras de formação para conectivos, por exemplo, poderíamos ter permitido isto:

Se φ e ψ são **fórmulas**, então $\varphi \rightarrow \psi$ também é.

Partindo de uma fórmula atômica θ , isso nos permitiria formar $\theta \rightarrow \theta$. A partir disso, juntamente com θ , obteríamos $\theta \rightarrow \theta \rightarrow \theta$. Mas há duas maneiras de fazer isso:

1. Tomamos θ como φ e $\theta \rightarrow \theta$ como ψ .
2. Tomamos φ como $\theta \rightarrow \theta$ e ψ é θ .

Correspondentemente, há duas maneiras de “ler” a **fórmula** $\theta \rightarrow \theta \rightarrow \theta$. Ela é da forma $\psi \rightarrow \chi$ onde ψ é θ e χ é $\theta \rightarrow \theta$, mas *ela é também* da forma $\psi \rightarrow \chi$ com ψ sendo $\theta \rightarrow \theta$ e χ sendo θ .

Se isso acontece, nossas definições nem sempre funcionarão. Por exemplo, quando definimos o **operador principal** de uma fórmula, dizemos: em uma fórmula da forma $\psi \rightarrow \chi$, o **operador principal** é a ocorrência indicada de $\rightarrow$. Mas se podemos casar a fórmula $\theta \rightarrow \theta \rightarrow \theta$ com $\psi \rightarrow \chi$ das duas maneiras diferentes mencionadas acima, então em um caso obtemos a primeira ocorrência de $\rightarrow$ como o **operador principal**, e no segundo caso a segunda ocorrência. Mas pretendemos que o **operador principal** seja uma *função* da **fórmula**, isto é, toda **fórmula** deve ter exatamente uma ocorrência de **operador principal**.

Lema 15.9. *O número de parênteses esquerdos e direitos em a fórmula φ é igual.*

Prova. Provamos isso por indução sobre a maneira como φ é construída. Isso requer duas coisas: (a) Temos primeiro que provar que todas as fórmulas atômicas têm a propriedade em questão (a base da indução). (b) Então temos que provar que, quando construímos novas fórmulas a partir de fórmulas dadas, as novas fórmulas têm a propriedade, desde que as antigas a tenham.

Seja $l(\varphi)$ o número de parênteses esquerdos, e $r(\varphi)$ o número de parênteses direitos em φ , e $l(t)$ e $r(t)$ similarmente o número de parênteses esquerdos e direitos em um termo t .

Problema 15.2. Prove que, para qualquer termo t , $l(t) = r(t)$.

CAPÍTULO 15. SINTAXE DA LÓGICA DE PRIMEIRA ORDEM

1. $\varphi \equiv \perp$: φ tem 0 parênteses esquerdos e 0 parênteses direitos.
2. $\varphi \equiv \top$: φ tem 0 parênteses esquerdos e 0 parênteses direitos.
3. $\varphi \equiv R(t_1, \dots, t_n)$: $l(\varphi) = 1 + l(t_1) + \dots + l(t_n) = 1 + r(t_1) + \dots + r(t_n) = r(\varphi)$. Aqui usamos o fato, deixado como exercício, de que $l(t) = r(t)$ para qualquer termo t .
4. $\varphi \equiv t_1 = t_2$: $l(\varphi) = l(t_1) + l(t_2) = r(t_1) + r(t_2) = r(\varphi)$.
5. $\varphi \equiv \neg\psi$: Por hipótese de indução, $l(\psi) = r(\psi)$. Assim $l(\varphi) = l(\psi) = r(\psi) = r(\varphi)$.
6. $\varphi \equiv (\psi * \chi)$: Por hipótese de indução, $l(\psi) = r(\psi)$ e $l(\chi) = r(\chi)$. Assim $l(\varphi) = 1 + l(\psi) + l(\chi) = 1 + r(\psi) + r(\chi) = r(\varphi)$.
7. $\varphi \equiv \forall x \psi$: Por hipótese de indução, $l(\psi) = r(\psi)$. Assim, $l(\varphi) = l(\psi) = r(\psi) = r(\varphi)$.
8. $\varphi \equiv \exists x \psi$: Analogamente. □

Definição 15.10 (Prefixo próprio). Uma cadeia de símbolos ψ é um *prefixo próprio* de uma cadeia de símbolos φ se a concatenação de ψ e uma cadeia de símbolos não vazia resulta em φ .

Lema 15.11. Se φ é *a fórmula*, e ψ é um *prefixo próprio* de φ , então ψ não é *a fórmula*. fol:syn:unq:
lem:no-prefix

Prova. Exercício. □

Problema 15.3. Prove Lema 15.11.

Proposição 15.12. Se φ é uma *fórmula* atômica, então ela satisfaz uma, e apenas uma, das seguintes condições. fol:syn:unq:
prop:unique-atomic

1. $\varphi \equiv \perp$.
2. $\varphi \equiv \top$.
3. $\varphi \equiv R(t_1, \dots, t_n)$ onde R é um *símbolo de predicado* de n lugares, $t_1, \dots, t_n$ são termos, e cada um de $R, t_1, \dots, t_n$ é determinado de forma única.
4. $\varphi \equiv t_1 = t_2$ onde t_1 e t_2 são termos determinados de forma única.

Prova. Exercício. □

Problema 15.4. Prove Proposição 15.12 (Dica: Formule e prove uma versão de Lema 15.11 para termos.)

15.5. OPERADOR PRINCIPAL DE UMA FÓRMULA

Proposição 15.13 (Leitura Única). *Toda fórmula satisfaz uma, e apenas uma, das seguintes condições.*

1. φ é atômica.
2. φ é da forma $\neg\psi$.
3. φ é da forma $(\psi \wedge \chi)$.
4. φ é da forma $(\psi \vee \chi)$.
5. φ é da forma $(\psi \rightarrow \chi)$.
6. φ é da forma $(\psi \leftrightarrow \chi)$.
7. φ é da forma $\forall x \psi$.
8. φ é da forma $\exists x \psi$.

Além disso, em cada caso ψ , ou ψ e χ , são determinados de forma única. Isso significa que, por exemplo, não há pares diferentes ψ, χ e ψ', χ' tais que φ seja ao mesmo tempo da forma $(\psi \rightarrow \chi)$ e $(\psi' \rightarrow \chi')$.

Prova. As regras de formação exigem que, se a fórmula não é atômica, ela deve começar com um parêntese de abertura ($($, $\neg$, ou um quantificador. Por outro lado, toda fórmula que começa com um dos seguintes símbolos deve ser atômica: a símbolo de predicado, a símbolo de função, a símbolo de constante, $\perp$, $\top$.

Então, na verdade só temos que mostrar que se φ é da forma $(\psi * \chi)$ e também da forma $(\psi' *' \chi')$, então $\psi \equiv \psi'$, $\chi \equiv \chi'$, e $*$ = $*'$.

Então suponha que tanto $\varphi \equiv (\psi * \chi)$ quanto $\varphi \equiv (\psi' *' \chi')$. Então ou $\psi \equiv \psi'$ ou não. Se for, claramente $*$ = $*'$ e $\chi \equiv \chi'$, pois elas então são subcadeias de φ que começam no mesmo lugar e têm o mesmo comprimento. O outro caso é $\psi \not\equiv \psi'$. Como ψ e ψ' são ambas subcadeias de φ que começam no mesmo lugar, uma deve ser um prefixo próprio da outra. Mas isso é impossível por **Lema 15.11**. $\square$

content/first-order-logic/syntax-and-semantics/main-operator.tex

15.5 Operador principal de uma Fórmula

fol:syn:mai:
sec Frequentemente é útil falar sobre o último operador usado na construção de explanation a fórmula φ . Esse operador é chamado de *operador principal* de φ . Intuitivamente, é o operador “mais externo” de φ . Por exemplo, o operador principal de $\neg\varphi$ é $\neg$, o operador principal de $(\varphi \vee \psi)$ é $\vee$, etc.

fol:syn:mai:
def:main-op **Definição 15.14 (Operador principal).** O *operador principal* de a fórmula φ é definido como segue:

1. φ is atomic: φ não tem operador principal.
2. $\varphi \equiv \neg\psi$: o operador principal de φ é $\neg$.
3. $\varphi \equiv (\psi \wedge \chi)$: o operador principal de φ é $\wedge$.
4. $\varphi \equiv (\psi \vee \chi)$: o operador principal de φ é $\vee$.
5. $\varphi \equiv (\psi \rightarrow \chi)$: o operador principal de φ é $\rightarrow$.
6. $\varphi \equiv (\psi \leftrightarrow \chi)$: o operador principal de φ é $\leftrightarrow$.
7. $\varphi \equiv \forall x \psi$: o operador principal de φ é $\forall$.
8. $\varphi \equiv \exists x \psi$: o operador principal de φ é $\exists$.

Em cada caso, pretendemos a ocorrência específica indicada do operador principal na fórmula. Por exemplo, como a fórmula $((\theta \rightarrow \alpha) \rightarrow (\alpha \rightarrow \theta))$ é da forma $(\psi \rightarrow \chi)$ onde ψ é $(\theta \rightarrow \alpha)$ e χ é $(\alpha \rightarrow \theta)$, a segunda ocorrência de $\rightarrow$ é o operador principal.

Esta é uma definição recursiva de uma função que mapeia todas as fórmulas não atômicas para sua ocorrência de operador principal. Por causa da maneira como as fórmulas são definidas indutivamente, toda fórmula φ satisfaz um dos casos em Definição 15.14. Isso garante que, para cada fórmula φ não atômica, existe um operador principal. Como cada fórmula satisfaz apenas uma dessas condições, e como as fórmulas menores a partir das quais φ é construída são determinadas de forma única em cada caso, a ocorrência de operador principal de φ é única, e assim definimos uma função.

Chamamos as fórmulas pelos nomes em Tabela 15.1 dependendo de qual símbolo é seu operador principal.

Operador principal	Tipo de fórmula	Exemplo
nenhum	(fórmula) atômica	$\perp, \top, R(t_1, \dots, t_n), t_1 = t_2$
$\neg$	negação	$\neg\varphi$
$\wedge$	conjunção	$(\varphi \wedge \psi)$
$\vee$	disjunção	$(\varphi \vee \psi)$
$\rightarrow$	condicional	$(\varphi \rightarrow \psi)$
$\leftrightarrow$	bicondicional	$(\varphi \leftrightarrow \psi)$
$\forall$	(fórmula) universal	$\forall x \varphi$
$\exists$	(fórmula) existencial	$\exists x \varphi$

Tabela 15.1: Operador principal e nomes de fórmulas

content/first-order-logic/syntax-and-semantics/subformulas.tex

fol:syn:mai:
tab:main-op

15.6 Subfórmulas

explanation

fol:syn:sbf:
sec

15.6. SUBFÓRMULAS

Frequentemente é útil falar sobre as **fórmulas** que “compõem” uma dada **fórmula**. Chamamos estas de suas **subfórmulas**. Qualquer **fórmula** conta como a **subfórmula** de si mesma; uma subfórmula de φ diferente do próprio φ é uma **subfórmula própria**.

Definição 15.15 (Subfórmula imediata). Se φ é a **fórmula**, as **subfórmulas imediatas** de φ são definidas indutivamente como segue:

1. **Fórmulas** atômicas não têm **subfórmulas** imediatas.
2. $\varphi \equiv \neg\psi$: A única **subfórmula** imediata de φ é ψ .
3. $\varphi \equiv (\psi * \chi)$: As **subfórmulas** imediatas de φ são ψ e χ ($*$ é qualquer um dos conectivos de dois lugares).
4. $\varphi \equiv \forall x \psi$: A única **subfórmula** imediata de φ é ψ .
5. $\varphi \equiv \exists x \psi$: A única **subfórmula** imediata de φ é ψ .

Definição 15.16 (Subfórmula própria). Se φ é a **fórmula**, as **subfórmulas próprias** de φ são definidas recursivamente como segue:

1. **Fórmulas** atômicas não têm **subfórmulas** próprias.
2. $\varphi \equiv \neg\psi$: As **subfórmulas** próprias de φ são ψ juntamente com todas as **subfórmulas** próprias de ψ .
3. $\varphi \equiv (\psi * \chi)$: As **subfórmulas** próprias de φ são ψ , χ , juntamente com todas as **subfórmulas** próprias de ψ e as de χ .
4. $\varphi \equiv \forall x \psi$: As **subfórmulas** próprias de φ são ψ juntamente com todas as **subfórmulas** próprias de ψ .
5. $\varphi \equiv \exists x \psi$: As **subfórmulas** próprias de φ são ψ juntamente com todas as **subfórmulas** próprias de ψ .

Definição 15.17 (Subfórmula). As **subfórmulas** de φ são o próprio φ juntamente com todas as suas **subfórmulas** próprias.

Note a diferença sutil em como definimos **subfórmulas** imediatas e **subfórmulas**[explanation](#) próprias. No primeiro caso, definimos diretamente as **subfórmulas** imediatas de uma fórmula φ para cada forma possível de φ . É uma definição explícita por casos, e os casos espelham a definição indutiva do conjunto de **fórmulas**. No segundo caso, também espelhamos a maneira como o conjunto de todas as **fórmulas** é definido, mas em cada caso também incluímos as **subfórmulas** próprias das **fórmulas** menores ψ , χ além dessas próprias **fórmulas**. Isso torna a definição *recursiva*. Em geral, uma definição de uma função sobre um conjunto definido indutivamente (no nosso caso, **fórmulas**) é recursiva se os casos na definição da função fazem uso da própria função. Para ser bem definida,

devemos garantir, no entanto, que só usamos os valores da função para argumentos que vêm “antes” daquele que estamos definindo—no nosso caso, ao definir “subfórmula própria” para $(\psi * \chi)$ usamos apenas as subfórmulas próprias das fórmulas “anteriores” ψ e χ .

Proposição 15.18. *Suponha que ψ é uma subfórmula de φ e χ é uma subfórmula de ψ . Então χ é uma subfórmula de φ . Em outras palavras, a relação de subfórmula é transitiva.* fol:syn:sbf: prop:subfrm-trans

Problema 15.5. Prove **Proposição 15.18**.

Proposição 15.19. *Suponha que φ é uma fórmula com n conectivos e quantificadores. Então φ tem no máximo $2n + 1$ subfórmulas.* fol:syn:sbf: prop:count-subfrms

Problema 15.6. Prove **Proposição 15.19**.

`content/first-order-logic/syntax-and-semantics/formation-sequences.tex`

15.7 Sequências de Formação

Definir fórmulas via uma definição indutiva, e a técnica complementar de provar propriedades de fórmulas via indução, é uma abordagem elegante e eficiente. No entanto, também pode ser útil considerar uma abordagem mais ascendente, passo a passo, da construção de fórmulas, o que fazemos aqui usando a noção de uma *sequência de formação*. Para mostrar como termos e fórmulas podem ser introduzidos dessa maneira sem precisar referir-se a suas definições indutivas, primeiro introduzimos a noção de uma cadeia arbitrária de símbolos extraída de alguma linguagem $\mathcal{L}$. fol:syn:fseq: sec

Definição 15.20 (Cadeias). Suponha que $\mathcal{L}$ é uma linguagem de primeira ordem. Uma $\mathcal{L}$ -cadeia é uma sequência finita de símbolos de $\mathcal{L}$. Quando a linguagem $\mathcal{L}$ está claramente fixada pelo contexto, frequentemente nos referiremos a uma $\mathcal{L}$ -cadeia simplesmente como uma *cadeia*. fol:syn:fseq: defn:string

Exemplo 15.21. Para qualquer linguagem de primeira ordem $\mathcal{L}$, todas as $\mathcal{L}$ -fórmulas são $\mathcal{L}$ -cadeias, mas não reciprocamente. Por exemplo,

$$)(v_0 \rightarrow \exists$$

é uma $\mathcal{L}$ -cadeia mas não uma $\mathcal{L}$ -fórmula.

Definição 15.22 (Sequências de formação para termos). Uma sequência finita de $\mathcal{L}$ -cadeias $\langle t_0, \dots, t_n \rangle$ é uma *sequência de formação* para um termo t se $t \equiv t_n$ e para todo $i \leq n$, ou t_i é a *variável* ou a *símbolo de constante*, ou $\mathcal{L}$ contém um *símbolo de função* f k -ário e existem $m_0, \dots, m_k < i$ tais que $t_i \equiv f(t_{m_0}, \dots, t_{m_k})$. Quando for necessário distinguir, nos referiremos às sequências de formação para termos como *sequências de formação de termos*. fol:syn:fseq: defn:fseq-trm

15.7. SEQUÊNCIAS DE FORMAÇÃO

Exemplo 15.23. A sequência

$$\langle c_0, v_0, f_0^2(c_0, v_0), f_0^1(f_0^2(c_0, v_0)) \rangle$$

é uma sequência de formação para o termo $f_0^1(f_0^2(c_0, v_0))$, assim como

$$\langle v_0, c_0, f_0^2(c_0, v_0), f_0^1(f_0^2(c_0, v_0)) \rangle.$$

fol:syn:fseq:
defn:fseq-frm

Definição 15.24 (Sequências de formação para fórmulas). Uma sequência finita de $\mathcal{L}$ -cadeias $\langle \varphi_0, \dots, \varphi_n \rangle$ é uma *sequência de formação* para φ se $\varphi \equiv \varphi_n$ e para todo $i \leq n$, ou φ_i é uma **fórmula** atômica ou existem $j, k < i$ e a **variável** x tais que uma das seguintes condições vale:

1. $\varphi_i \equiv \neg \varphi_j$.
2. $\varphi_i \equiv (\varphi_j \wedge \varphi_k)$.
3. $\varphi_i \equiv (\varphi_j \vee \varphi_k)$.
4. $\varphi_i \equiv (\varphi_j \rightarrow \varphi_k)$.
5. $\varphi_i \equiv (\varphi_j \leftrightarrow \varphi_k)$.
6. $\varphi_i \equiv \forall x \varphi_j$.
7. $\varphi_i \equiv \exists x \varphi_j$.

Quando for necessário distinguir, nos referiremos às sequências de formação para fórmulas como *sequências de formação de fórmulas*.

Exemplo 15.25.

$$\langle A_0^1(v_0), A_1^1(c_1), (A_1^1(c_1) \wedge A_0^1(v_0)), \exists v_0 (A_1^1(c_1) \wedge A_0^1(v_0)) \rangle$$

é uma sequência de formação de $\exists v_0 (A_1^1(c_1) \wedge A_0^1(v_0))$, assim como

$$\langle A_0^1(v_0), A_1^1(c_1), (A_1^1(c_1) \wedge A_0^1(v_0)), A_1^1(c_1), \forall v_1 A_0^1(v_0), \exists v_0 (A_1^1(c_1) \wedge A_0^1(v_0)) \rangle.$$

Como pode ser visto no segundo exemplo, sequências de formação podem conter “lixo”: **fórmulas** que são redundantes ou não contribuem para a construção.

fol:syn:fseq:
prop:formed

Proposição 15.26. Toda **fórmula** φ em $\text{Frm}(\mathcal{L})$ tem uma sequência de formação.

Prova. Suponha que φ é atômica. Então a sequência $\langle \varphi \rangle$ é uma sequência de formação para φ . Agora suponha que ψ e χ têm sequências de formação $\langle \psi_0, \dots, \psi_n \rangle$ e $\langle \chi_0, \dots, \chi_m \rangle$ respectivamente.

1. Se $\varphi \equiv \neg \psi$, então $\langle \psi_0, \dots, \psi_n, \neg \psi_n \rangle$ é uma sequência de formação para φ .
2. Se $\varphi \equiv (\psi \wedge \chi)$, então $\langle \psi_0, \dots, \psi_n, \chi_0, \dots, \chi_m, (\psi_n \wedge \chi_m) \rangle$ é uma sequência de formação para φ .

CAPÍTULO 15. SINTAXE DA LÓGICA DE PRIMEIRA ORDEM

3. Se $\varphi \equiv (\psi \vee \chi)$, então $\langle \psi_0, \dots, \psi_n, \chi_0, \dots, \chi_m, (\psi_n \vee \chi_m) \rangle$ é uma sequência de formação para φ .
4. Se $\varphi \equiv (\psi \rightarrow \chi)$, então $\langle \psi_0, \dots, \psi_n, \chi_0, \dots, \chi_m, (\psi_n \rightarrow \chi_m) \rangle$ é uma sequência de formação para φ .
5. Se $\varphi \equiv (\psi \leftrightarrow \chi)$, então $\langle \psi_0, \dots, \psi_n, \chi_0, \dots, \chi_m, (\psi_n \leftrightarrow \chi_m) \rangle$ é uma sequência de formação para φ .
6. Se $\varphi \equiv \forall x \psi$, então $\langle \psi_0, \dots, \psi_n, \forall x \psi_n \rangle$ é uma sequência de formação para φ .
7. Se $\varphi \equiv \exists x \psi$, então $\langle \psi_0, \dots, \psi_n, \exists x \psi_n \rangle$ é uma sequência de formação para φ .

Pelo princípio de indução sobre fórmulas, toda fórmula tem uma sequência de formação. □

Também podemos provar a recíproca. Isso é importante porque mostra que nossas duas maneiras de definir fórmulas são equivalentes: elas dão os mesmos resultados. Isso também significa que podemos provar teoremas sobre fórmulas usando indução ordinária sobre o comprimento das sequências de formação.

Lema 15.27. *Suponha que $\langle \varphi_0, \dots, \varphi_n \rangle$ é uma sequência de formação para φ_n , e que $k \leq n$. Então $\langle \varphi_0, \dots, \varphi_k \rangle$ é uma sequência de formação para φ_k .* fol:syn:fseq;
lem:fseq-init

Prova. Exercício. □

Problema 15.7. Prove **Lema 15.27**.

Teorema 15.28. $\text{Frm}(\mathcal{L})$ é o conjunto de todas as $\mathcal{L}$ -cadeias φ tais que existe uma sequência de formação de fórmulas para φ . fol:syn:fseq;
thm:fseq-frm-equiv

Prova. Seja F o conjunto de todas as cadeias de símbolos na linguagem $\mathcal{L}$ que têm uma sequência de formação. Vimos em **Proposição 15.26** que $\text{Frm}(\mathcal{L}) \subseteq F$, então agora provamos a recíproca.

Suponha que φ tem uma sequência de formação $\langle \varphi_0, \dots, \varphi_n \rangle$. Provamos que $\varphi \in \text{Frm}(\mathcal{L})$ por indução forte sobre n . Nossa hipótese de indução é que toda cadeia de símbolos com uma sequência de formação de comprimento $m < n$ está em $\text{Frm}(\mathcal{L})$. Pela definição de uma sequência de formação, ou $\varphi \equiv \varphi_n$ é atômica ou devem existir $j, k < n$ tais que um dos seguintes casos vale:

1. $\varphi \equiv \neg \varphi_j$.
2. $\varphi \equiv (\varphi_j \wedge \varphi_k)$.
3. $\varphi \equiv (\varphi_j \vee \varphi_k)$.
4. $\varphi \equiv (\varphi_j \rightarrow \varphi_k)$.

15.7. SEQUÊNCIAS DE FORMAÇÃO

5. $\varphi \equiv (\varphi_j \leftrightarrow \varphi_k)$.

6. $\varphi \equiv \forall x \varphi_j$.

7. $\varphi \equiv \exists x \varphi_j$.

Agora raciocinamos por casos. Se φ é atômica então $\varphi_n \in \text{Frm}(\mathcal{L}_0)$. Suponha, em vez disso, que $\varphi \equiv (\varphi_j \wedge \varphi_k)$. Por **Lema 15.27**, $\langle \varphi_0, \dots, \varphi_j \rangle$ e $\langle \varphi_0, \dots, \varphi_k \rangle$ são sequências de formação para φ_j e φ_k , respectivamente. Como estas são subsequências iniciais próprias da sequência de formação para φ , ambas têm comprimento menor que n . Portanto, pela hipótese de indução, φ_j e φ_k estão em $\text{Frm}(\mathcal{L}_0)$, e pela definição de **a fórmula**, $(\varphi_j \wedge \varphi_k)$ também está. Os outros casos seguem por raciocínio análogo. $\square$

Sequências de formação para termos têm propriedades semelhantes às das **fórmulas**.

Proposição 15.29. *$\text{Trm}(\mathcal{L})$ é o conjunto de todas as $\mathcal{L}$ -cadeias t tais que existe uma sequência de formação de termos para t .*

Prova. Exercício. $\square$

Problema 15.8. Prove **Proposição 15.29**. Dica: use uma estratégia semelhante à usada na prova de **Teorema 15.28**.

Há dois tipos de “lixo” que podem aparecer em sequências de formação: elementos repetidos, e elementos que são irrelevantes para a construção da fórmula ou termo. Podemos eliminar ambos olhando para sequências de formação mínimas.

Definição 15.30 (Sequências de formação mínimas). Uma sequência de formação $\langle \varphi_0, \dots, \varphi_n \rangle$ para uma fórmula φ é uma *sequência de formação mínima* para φ se, para toda outra sequência de formação s para φ , o comprimento de s é maior ou igual a $n + 1$.

Similarmente, uma sequência de formação $\langle t_0, \dots, t_n \rangle$ para um termo t é uma *sequência de formação mínima* para t se, para toda outra sequência de formação s para t , o comprimento de s é maior ou igual a $n + 1$.

Note que uma fórmula ou termo pode ter mais de uma sequência de formação mínima, mas elas conterão exatamente as mesmas cadeias.

Proposição 15.31. *As seguintes condições são equivalentes:*

1. ψ é uma sub-fórmula de φ .
2. ψ ocorre em toda sequência de formação de φ .
3. ψ ocorre em uma sequência de formação mínima de φ .

Prova. Exercício. $\square$

Problema 15.9. Prove **Proposição 15.31**.

Observações históricas Sequências de formação foram introduzidas por Raymond Smullyan em seu livro-texto *First-Order Logic* (Smullyan, 1968). Propriedades adicionais de sequências de formação foram estabelecidas por Zuckerman (1973).

content/first-order-logic/syntax-and-semantics/free-vars-sentences.tex

15.8 Variáveis Livres e Sentenças

Definição 15.32 (Ocorrências livres de a variável). As ocorrências *livres* de a variável em a fórmula são definidas indutivamente como segue:

fol:syn:fvs:
sec
fol:syn:fvs:
defn:free-occ

1. φ is atomic: todas as ocorrências de variável em φ são livres.
2. $\varphi \equiv \neg\psi$: as ocorrências livres de variável de φ são exatamente as de ψ .
3. $\varphi \equiv (\psi*\chi)$: as ocorrências livres de variável de φ são as em ψ juntamente com as em χ .
4. $\varphi \equiv \forall x\psi$: as ocorrências livres de variável em φ são todas as em ψ exceto as ocorrências de x .
5. $\varphi \equiv \exists x\psi$: as ocorrências livres de variável em φ são todas as em ψ exceto as ocorrências de x .

Definição 15.33 (Variáveis Ligadas). Uma ocorrência de a variável em uma fórmula φ é *ligada* se não é livre.

Problema 15.10. Dê uma definição indutiva das ocorrências de variáveis ligadas nos moldes de Definição 15.32.

Definição 15.34 (Escopo). Se $\forall x\psi$ é uma ocorrência de uma subfórmula em uma fórmula φ , então a ocorrência correspondente de ψ em φ é chamada de *escopo* da ocorrência correspondente de $\forall x$. Analogamente para $\exists x$.

Se ψ é o escopo de uma ocorrência de quantificador $\forall x$ ou $\exists x$ em φ , então as ocorrências livres de x em ψ são ligadas em $\forall x\psi$ e $\exists x\psi$. Dizemos que essas ocorrências são *ligadas pela* ocorrência de quantificador mencionada.

Exemplo 15.35. Considere a seguinte fórmula:

$$\exists v_0 \underbrace{A_0^2(v_0, v_1)}_{\psi}$$

ψ representa o escopo de $\exists v_0$. O quantificador liga a ocorrência de v_0 em ψ , mas não liga a ocorrência de v_1 . Então v_1 é uma variável livre neste caso.

15.9. SUBSTITUIÇÃO

Agora podemos ver como isso poderia funcionar em uma fórmula φ mais complicada:

$$\forall v_0 \underbrace{(A_0^1(v_0) \rightarrow A_0^2(v_0, v_1))}_{\psi} \rightarrow \exists v_1 \underbrace{(A_1^2(v_0, v_1) \vee \forall v_0 \overbrace{\neg A_1^1(v_0)}^{\theta})}_{\chi}$$

ψ é o escopo do primeiro $\forall v_0$, χ é o escopo de $\exists v_1$, e θ é o escopo do segundo $\forall v_0$. O primeiro $\forall v_0$ liga as ocorrências de v_0 em ψ , $\exists v_1$ liga a ocorrência de v_1 em χ , e o segundo $\forall v_0$ liga a ocorrência de v_0 em θ . A primeira ocorrência de v_1 e a quarta ocorrência de v_0 são livres em φ . A última ocorrência de v_0 é livre em θ , mas ligada em χ e φ .

Definição 15.36 (Sentença). A fórmula φ é a *sentença* sse não contém ocorrências livres de *variáveis*.

`content/first-order-logic/syntax-and-semantics/substitution.tex`

15.9 Substituição

fol:syn:sub:
sec

Definição 15.37 (Substituição em um termo). Definimos $s[t/x]$, o resultado de *substituir* t por toda ocorrência de x em s , recursivamente:

1. $s \equiv c$: $s[t/x]$ é simplesmente s .
2. $s \equiv y$: $s[t/x]$ é também simplesmente s , desde que y seja uma variável e $y \neq x$.
3. $s \equiv x$: $s[t/x]$ é t .
4. $s \equiv f(t_1, \dots, t_n)$: $s[t/x]$ é $f(t_1[t/x], \dots, t_n[t/x])$.

Definição 15.38. Um termo t está *livre para* x em φ se nenhuma das ocorrências livres de x em φ ocorre no escopo de um quantificador que liga uma variável em t .

Exemplo 15.39.

1. v_8 está livre para v_1 em $\exists v_3 A_4^2(v_3, v_1)$
2. $f_1^2(v_1, v_2)$ não está livre para v_0 em $\forall v_2 A_4^2(v_0, v_2)$

Definição 15.40 (Substituição em a fórmula). Se φ é a fórmula, x é a variável, e t é um termo livre para x em φ , então $\varphi[t/x]$ é o resultado de substituir t por todas as ocorrências livres de x em φ .

1. $\varphi \equiv \perp$: $\varphi[t/x]$ é $\perp$.

2. $\varphi \equiv \top$: $\varphi[t/x]$ é $\top$.
3. $\varphi \equiv P(t_1, \dots, t_n)$: $\varphi[t/x]$ é $P(t_1[t/x], \dots, t_n[t/x])$.
4. $\varphi \equiv t_1 = t_2$: $\varphi[t/x]$ é $t_1[t/x] = t_2[t/x]$.
5. $\varphi \equiv \neg\psi$: $\varphi[t/x]$ é $\neg\psi[t/x]$.
6. $\varphi \equiv (\psi \wedge \chi)$: $\varphi[t/x]$ é $(\psi[t/x] \wedge \chi[t/x])$.
7. $\varphi \equiv (\psi \vee \chi)$: $\varphi[t/x]$ é $(\psi[t/x] \vee \chi[t/x])$.
8. $\varphi \equiv (\psi \rightarrow \chi)$: $\varphi[t/x]$ é $(\psi[t/x] \rightarrow \chi[t/x])$.
9. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\varphi[t/x]$ é $(\psi[t/x] \leftrightarrow \chi[t/x])$.
10. $\varphi \equiv \forall y \psi$: $\varphi[t/x]$ é $\forall y \psi[t/x]$, desde que y seja uma variável diferente de x ; caso contrário $\varphi[t/x]$ é simplesmente φ .
11. $\varphi \equiv \exists y \psi$: $\varphi[t/x]$ é $\exists y \psi[t/x]$, desde que y seja uma variável diferente de x ; caso contrário $\varphi[t/x]$ é simplesmente φ .

explanation

Note que a substituição pode ser vácuca: Se x não ocorre de modo algum em φ , então $\varphi[t/x]$ é simplesmente φ .

A restrição de que t deve estar livre para x em φ é necessária para excluir casos como o seguinte. Se $\varphi \equiv \exists y x < y$ e $t \equiv y$, então $\varphi[t/x]$ seria $\exists y y < y$. Neste caso a variável livre y é “capturada” pelo quantificador $\exists y$ ao substituir, e isso é indesejável. Por exemplo, gostaríamos que fosse o caso que, sempre que $\forall x \psi$ vale, $\psi[t/x]$ também vale. Mas considere $\forall x \exists y x < y$ (aqui ψ é $\exists y x < y$). É uma sentença que é verdadeira a respeito, por exemplo, dos números naturais: para todo número x há um número y maior que ele. Se permitíssemos y como uma possível substituição para x , terminaríamos com $\psi[y/x] \equiv \exists y y < y$, que é falsa. Prevenimos isso exigindo que nenhuma das variáveis livres em t acabe sendo ligada por um quantificador em φ .

Frequentemente usamos a seguinte convenção para evitar notação incômoda: Se φ é a fórmula que pode conter a variável x livre, também escrevemos $\varphi(x)$ para indicar isso. Quando está claro qual φ e x temos em mente, e t é um termo (suposto estar livre para x em $\varphi(x)$), então escrevemos $\varphi(t)$ como abreviação de $\varphi[t/x]$. Assim, por exemplo, poderíamos dizer, “chamamos $\varphi(t)$ de uma instância de $\forall x \varphi(x)$.” Com isso queremos dizer que se φ é qualquer fórmula,

x a **variável**, e t um termo que está livre para x em φ , então $\varphi[t/x]$ é uma instância de $\forall x \varphi$.

Capítulo 16

Semântica da Lógica de Primeira Ordem

`content/first-order-logic/syntax-and-semantics/intro-semantics.tex`

16.1 Introdução

fol:syn:its:
sec

Dar o significado das expressões é o domínio da semântica. O conceito central em semântica é o de satisfação em **a estrutura**. **A estrutura** dá significado aos blocos de construção da linguagem: **a domínio** é um conjunto não vazio de objetos. Os quantificadores são interpretados como variando sobre esse domínio, aos **símbolos de constante** são atribuídos elementos no domínio, aos **símbolos de função** são atribuídas funções do **domínio** para si mesmo, e aos **símbolos de predicado** são atribuídas relações sobre o **domínio**. O **domínio** junto com as atribuições ao vocabulário básico constitui **a estrutura**. **Variáveis** podem aparecer em **fórmulas**, e para dar uma semântica, também temos que atribuir **membros** do **domínio** a elas—isto é uma atribuição de variáveis. A relação de satisfação, finalmente, reúne tudo isso. **A fórmula** pode ser satisfeita em **a estrutura** $\mathfrak{M}$ relativamente a uma atribuição de variáveis s , escrita como $\mathfrak{M}, s \models \varphi$. Essa relação também é definida por indução sobre a estrutura de φ , usando as tabelas de verdade para os conectivos lógicos para definir, por exemplo, a satisfação de $(\varphi \wedge \psi)$ em termos da satisfação (ou não) de φ e ψ . Acontece então que a atribuição de variáveis é irrelevante se **a fórmula** φ é **a sentença**, isto é, não tem variáveis livres, e assim podemos falar de **sentenças** sendo simplesmente satisfeitas (ou não) em **estruturas**.

Com base na relação de satisfação $\mathfrak{M} \models \varphi$ para **sentenças** podemos então definir as noções semânticas básicas de validade, consequência e satisfatibilidade. **A sentença** é válida, $\models \varphi$, se toda **estrutura** a satisfaz. Ela é consequência de um conjunto de **sentenças**, $\Gamma \models \varphi$, se toda **estrutura** que satisfaz todas as **sentenças** em Γ também satisfaz φ . E um conjunto de **sentenças** é satisfatível se alguma

estrutura satisfaz todas as **sentenças** nele ao mesmo tempo. Como **fórmulas** são definidas indutivamente, e a satisfação é por sua vez definida por indução sobre a estrutura das **fórmulas**, podemos usar indução para provar propriedades de nossa semântica e para relacionar as noções semânticas definidas.

`content/first-order-logic/syntax-and-semantics/structures.tex`

16.2 Estruturas para Linguagens de Primeira Ordem

explanation As linguagens de primeira ordem são, por si mesmas, *não interpretadas*: os fol:syn:str:sec **símbolos de constante**, **símbolos de função** e **símbolos de predicado** não têm nenhum significado específico associado a eles. Os significados são dados especificando-se a **estrutura**. Ela especifica o *domínio*, isto é, os objetos que os **símbolos de constante** denotam, sobre os quais os **símbolos de função** operam e sobre os quais os quantificadores variam. Além disso, ela especifica quais **símbolos de constante** denotam quais objetos, como a **símbolo de função** mapeia objetos para objetos, e a quais objetos os **símbolos de predicado** se aplicam. **Estruturas** são a base para as noções *semânticas* em lógica, por exemplo, a noção de consequência, validade, satisfatibilidade. Elas são chamadas de várias formas na literatura: “estruturas”, “interpretações” ou “modelos”.

Definição 16.1 (Estruturas). A **estrutura** $\mathfrak{M}$, para uma linguagem $\mathcal{L}$ de lógica de primeira ordem consiste nos seguintes elementos:

1. *Domínio*: um conjunto não vazio, $|\mathfrak{M}|$
2. *Interpretação dos símbolos de constante*: para cada **símbolo de constante** c de $\mathcal{L}$, um **membro** $c^{\mathfrak{M}} \in |\mathfrak{M}|$
3. *Interpretação dos símbolos de predicado*: para cada **símbolo de predicado** R de n lugares de $\mathcal{L}$ (distinto de $=$), uma relação de n lugares $R^{\mathfrak{M}} \subseteq |\mathfrak{M}|^n$
4. *Interpretação dos símbolos de função*: para cada **símbolo de função** f de n lugares de $\mathcal{L}$, uma função de n lugares $f^{\mathfrak{M}}: |\mathfrak{M}|^n \rightarrow |\mathfrak{M}|$

Exemplo 16.2. A **estrutura** $\mathfrak{N}$ para a linguagem da aritmética consiste em um conjunto, um elemento de $|\mathfrak{N}|$, $0^{\mathfrak{N}}$, como interpretação do **símbolo de constante** 0 , uma função de um lugar $\iota^{\mathfrak{N}}: |\mathfrak{N}| \rightarrow |\mathfrak{N}|$, duas funções de dois lugares $+^{\mathfrak{N}}$ e $\times^{\mathfrak{N}}$, ambas $|\mathfrak{N}|^2 \rightarrow |\mathfrak{N}|$, e uma relação de dois lugares $<^{\mathfrak{N}} \subseteq |\mathfrak{N}|^2$.

Um exemplo óbvio de tal estrutura é o seguinte:

1. $|\mathfrak{N}| = \mathbb{N}$
2. $0^{\mathfrak{N}} = 0$
3. $\iota^{\mathfrak{N}}(n) = n + 1$ para todos $n \in \mathbb{N}$

16.3. ESTRUTURAS COBERTAS PARA LINGUAGENS DE PRIMEIRA ORDEM

4. $+^{\mathfrak{N}}(n, m) = n + m$ para todos $n, m \in \mathbb{N}$
5. $\times^{\mathfrak{N}}(n, m) = n \cdot m$ para todos $n, m \in \mathbb{N}$
6. $<^{\mathfrak{N}} = \{\langle n, m \rangle : n \in \mathbb{N}, m \in \mathbb{N}, n < m\}$

A estrutura $\mathfrak{N}$ para $\mathcal{L}_A$ assim definida é chamada de *modelo padrão da aritmética*, porque interpreta as constantes não lógicas de $\mathcal{L}_A$ exatamente como se esperaria.

No entanto, há muitas outras **estruturas** possíveis para $\mathcal{L}_A$. Por exemplo, poderíamos tomar como domínio o conjunto $\mathbb{Z}$ dos inteiros em vez de $\mathbb{N}$, e definir as interpretações de 0 , 1 , $+$, $\times$, $<$ adequadamente. Mas também podemos definir estruturas para $\mathcal{L}_A$ que não têm nada, nem mesmo remotamente, a ver com números.

Exemplo 16.3. Uma estrutura $\mathfrak{M}$ para a linguagem $\mathcal{L}_Z$ da teoria dos conjuntos requer apenas um conjunto e uma única relação de dois lugares. Então, tecnicamente, por exemplo, o conjunto das pessoas mais a relação “ x é mais velho que y ” poderia ser usado como **a estrutura** para $\mathcal{L}_Z$, assim como $\mathbb{N}$ junto com $n \geq m$ para $n, m \in \mathbb{N}$.

Uma **estrutura** particularmente interessante para $\mathcal{L}_Z$ na qual os **membros** do domínio são de fato conjuntos, e a interpretação de $\in$ é de fato a relação “ x é um membro de y ” é a **estrutura** $\mathfrak{H}\mathfrak{F}$ dos *conjuntos hereditariamente finitos*:

1. $|\mathfrak{H}\mathfrak{F}| = \emptyset \cup \wp(\emptyset) \cup \wp(\wp(\emptyset)) \cup \wp(\wp(\wp(\emptyset))) \cup \dots;$
2. $\in^{\mathfrak{H}\mathfrak{F}} = \{\langle x, y \rangle : x, y \in |\mathfrak{H}\mathfrak{F}|, x \in y\}.$

As estipulações que fazemos sobre o que conta como **a estrutura** impactam digression nossa lógica. Por exemplo, a escolha de impedir domínios vazios garante, dada a explicação usual de satisfação (ou verdade) para sentenças quantificadas, que $\exists x (\varphi(x) \vee \neg \varphi(x))$ é válida—isto é, uma verdade lógica. E a estipulação de que todos os **símbolos de constante** devem referir-se a um objeto no domínio garante que a generalização existencial é um padrão de inferência correto: $\varphi(a)$, portanto $\exists x \varphi(x)$. Se permitíssemos que nomes se referissem a algo fora do domínio, ou que não se referissem, então estaríamos a caminho de uma *lógica livre*, na qual a generalização existencial requer uma premissa adicional: $\varphi(a)$ e $\exists x x = a$, portanto $\exists x \varphi(x)$.

content/first-order-logic/syntax-and-semantics/covered-structures.tex

16.3 Estruturas Cobertas para Linguagens de Primeira Ordem

fol:syn:cov:
sec

Lembre-se de que um termo é *fechado* se não contém **variáveis**.

explanation

Definição 16.4 (Valor de termos fechados). Se t é um termo fechado da linguagem $\mathcal{L}$ e $\mathfrak{M}$ é **a estrutura** para $\mathcal{L}$, o **valor** $\text{Val}^{\mathfrak{M}}(t)$ é definido como segue:

1. Se t é apenas o **símbolo de constante** c , então $\text{Val}^{\mathfrak{M}}(c) = c^{\mathfrak{M}}$.
2. Se t é da forma $f(t_1, \dots, t_n)$, então

$$\text{Val}^{\mathfrak{M}}(t) = f^{\mathfrak{M}}(\text{Val}^{\mathfrak{M}}(t_1), \dots, \text{Val}^{\mathfrak{M}}(t_n)).$$

Definição 16.5 (Estrutura Coberta). A estrutura é *coberta* se todo elemento do domínio é o **valor** de algum termo fechado.

Exemplo 16.6. Seja $\mathcal{L}$ a linguagem com os **símbolos de constante** *zero*, *um*, *dois*, ..., o **símbolo de predicado** binário $<$, e os **símbolos de função** binários $+$ e $\times$. Então a estrutura $\mathfrak{M}$ para $\mathcal{L}$ é aquela com domínio $|\mathfrak{M}| = \{0, 1, 2, \dots\}$ e atribuições $\text{zero}^{\mathfrak{M}} = 0$, $\text{um}^{\mathfrak{M}} = 1$, $\text{dois}^{\mathfrak{M}} = 2$, e assim por diante. Para o símbolo de relação binário $<$, o conjunto $<^{\mathfrak{M}}$ é o conjunto de todos os pares $\langle c_1, c_2 \rangle \in |\mathfrak{M}|^2$ tais que c_1 é menor que c_2 : por exemplo, $\langle 1, 3 \rangle \in <^{\mathfrak{M}}$ mas $\langle 2, 2 \rangle \notin <^{\mathfrak{M}}$. Para o **símbolo de função** binário $+$, defina $+^{\mathfrak{M}}$ da maneira usual—por exemplo, $+^{\mathfrak{M}}(2, 3)$ resulta em 5, e analogamente para o **símbolo de função** binário $\times$. Portanto, o **valor** de *quatro* é apenas 4, e o **valor** de $\times(\text{dois}, +(\text{tres}, \text{zero}))$ (ou em notação infixa, $\text{dois} \times (\text{tres} + \text{zero})$) é

$$\begin{aligned} \text{Val}^{\mathfrak{M}}(\times(\text{dois}, +(\text{tres}, \text{zero}))) &= \\ &= \times^{\mathfrak{M}}(\text{Val}^{\mathfrak{M}}(\text{dois}), \text{Val}^{\mathfrak{M}}(+(\text{tres}, \text{zero}))) \\ &= \times^{\mathfrak{M}}(\text{Val}^{\mathfrak{M}}(\text{dois}), +^{\mathfrak{M}}(\text{Val}^{\mathfrak{M}}(\text{tres}), \text{Val}^{\mathfrak{M}}(\text{zero}))) \\ &= \times^{\mathfrak{M}}(\text{dois}^{\mathfrak{M}}, +^{\mathfrak{M}}(\text{tres}^{\mathfrak{M}}, \text{zero}^{\mathfrak{M}})) \\ &= \times^{\mathfrak{M}}(2, +^{\mathfrak{M}}(3, 0)) \\ &= \times^{\mathfrak{M}}(2, 3) \\ &= 6 \end{aligned}$$

Problema 16.1. $\mathfrak{N}$, o modelo padrão da aritmética, é coberto? Explique.

`content/first-order-logic/syntax-and-semantics/satisfaction.tex`

16.4 Satisfação de a Fórmula em a Estrutura

explanation A noção básica que relaciona expressões como termos e **fórmulas**, por um lado, e **estruturas**, por outro, são as de **valor** de um termo e **satisfação** de a fórmula. Informalmente, o **valor** de um termo é um **membro** de a estrutura—se o termo é apenas uma constante, seu **valor** é o objeto atribuído à constante pela estrutura, e se ele é construído usando **símbolos de função**, o **valor** é calculado a partir dos **valores** das constantes e das funções atribuídas às funções no termo. A **fórmula** é *satisfeita* em a estrutura se a interpretação dada aos predicados torna a **fórmula** verdadeira no domínio da estrutura. Essa noção de satisfação é especificada indutivamente: a especificação da estrutura afirma diretamente

fol:syn:sat:
sec

16.4. SATISFAÇÃO DE A FÓRMULA EM A ESTRUTURA

quando **fórmulas** atômicas são satisfeitas, e definimos quando uma **fórmula** complexa é satisfeita dependendo do conectivo principal ou quantificador e de se as **subfórmulas** imediatas são ou não satisfeitas.

O caso dos quantificadores aqui é um pouco delicado, pois a **subfórmula** imediata de uma **fórmula** quantificada tem uma **variável** livre, e **estruturas** não especificam os **valores** das **variáveis**. Para lidar com essa dificuldade, também introduzimos *atribuições de variáveis* e definimos a satisfação não em relação a a **estrutura** sozinha, mas em relação a a **estrutura** mais uma atribuição de variáveis.

Definição 16.7 (Atribuição de Variáveis). Uma *atribuição de variáveis* s para a **estrutura** $\mathfrak{M}$ é uma função que mapeia cada **variável** para um **membro** de $|\mathfrak{M}|$, isto é, $s: \text{Var} \rightarrow |\mathfrak{M}|$.

A **estrutura** atribui a **valor** a cada **símbolo de constante**, e uma atribuição explanation de variáveis a cada variável. Mas queremos usar termos construídos a partir deles para também nomear **membros** do **domínio**. Para isso, definimos o **valor** dos termos indutivamente. Para **símbolos de constante** e variáveis, o valor é exatamente como a **estrutura** ou a atribuição de variáveis o especifica; para termos mais complexos, ele é calculado recursivamente usando as funções que a **estrutura** atribui aos **símbolos de função**.

Definição 16.8 (Valor de Termos). Se t é um termo da linguagem $\mathcal{L}$, $\mathfrak{M}$ é a **estrutura** para $\mathcal{L}$, e s é uma atribuição de variáveis para $\mathfrak{M}$, o **valor** $\text{Val}_s^{\mathfrak{M}}(t)$ é definido como segue:

$$1. t \equiv c: \text{Val}_s^{\mathfrak{M}}(t) = c^{\mathfrak{M}}.$$

$$2. t \equiv x: \text{Val}_s^{\mathfrak{M}}(t) = s(x).$$

$$3. t \equiv f(t_1, \dots, t_n):$$

$$\text{Val}_s^{\mathfrak{M}}(t) = f^{\mathfrak{M}}(\text{Val}_s^{\mathfrak{M}}(t_1), \dots, \text{Val}_s^{\mathfrak{M}}(t_n)).$$

Definição 16.9 (x -Variante). Se s é uma atribuição de variáveis para a **estrutura** $\mathfrak{M}$, então qualquer atribuição de variáveis s' para $\mathfrak{M}$ que difere de s no máximo no que atribui a x é chamada de x -variante de s . Se s' é uma x -variante de s escrevemos $s' \sim_x s$.

Note que uma x -variante de uma atribuição s não *precisa* atribuir algo explanation diferente a x . De fato, toda atribuição conta como uma x -variante de si mesma.

Definição 16.10. Se s é uma atribuição de variáveis para a **estrutura** $\mathfrak{M}$ e $m \in |\mathfrak{M}|$, então a atribuição $s[m/x]$ é a atribuição de variáveis definida por

$$s[m/x](y) = \begin{cases} m & \text{se } y \equiv x \\ s(y) & \text{caso contrário.} \end{cases}$$

Em outras palavras, $s[m/x]$ é a x -variante particular de s que atribui o **membro** m do domínio a x , e atribui as mesmas coisas às **variáveis** diferentes de x que s atribui.

Definição 16.11 (Satisfação). A satisfação de a **fórmula** φ em a **estrutura** $\mathfrak{M}$ relativamente a uma atribuição de variáveis s , em símbolos: $\mathfrak{M}, s \models \varphi$, é definida recursivamente como segue. (Escrevemos $\mathfrak{M}, s \not\models \varphi$ para significar “não $\mathfrak{M}, s \models \varphi$ ”.)

fol: syn: sat:
defn: satisfaction

1. $\varphi \equiv \perp$: $\mathfrak{M}, s \not\models \varphi$.
2. $\varphi \equiv \top$: $\mathfrak{M}, s \models \varphi$.
3. $\varphi \equiv R(t_1, \dots, t_n)$: $\mathfrak{M}, s \models \varphi$ sse $\langle \text{Val}_s^{\mathfrak{M}}(t_1), \dots, \text{Val}_s^{\mathfrak{M}}(t_n) \rangle \in R^{\mathfrak{M}}$.
4. $\varphi \equiv t_1 = t_2$: $\mathfrak{M}, s \models \varphi$ sse $\text{Val}_s^{\mathfrak{M}}(t_1) = \text{Val}_s^{\mathfrak{M}}(t_2)$.
5. $\varphi \equiv \neg\psi$: $\mathfrak{M}, s \models \varphi$ sse $\mathfrak{M}, s \not\models \psi$.
6. $\varphi \equiv (\psi \wedge \chi)$: $\mathfrak{M}, s \models \varphi$ sse $\mathfrak{M}, s \models \psi$ e $\mathfrak{M}, s \models \chi$.
7. $\varphi \equiv (\psi \vee \chi)$: $\mathfrak{M}, s \models \varphi$ sse $\mathfrak{M}, s \models \psi$ ou $\mathfrak{M}, s \models \chi$ (ou ambos).
8. $\varphi \equiv (\psi \rightarrow \chi)$: $\mathfrak{M}, s \models \varphi$ sse $\mathfrak{M}, s \not\models \psi$ ou $\mathfrak{M}, s \models \chi$ (ou ambos).
9. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\mathfrak{M}, s \models \varphi$ sse ou ambos $\mathfrak{M}, s \models \psi$ e $\mathfrak{M}, s \models \chi$, ou nem $\mathfrak{M}, s \models \psi$ nem $\mathfrak{M}, s \models \chi$.
10. $\varphi \equiv \forall x \psi$: $\mathfrak{M}, s \models \varphi$ sse para todo **membro** $m \in |\mathfrak{M}|$, $\mathfrak{M}, s[m/x] \models \psi$.
11. $\varphi \equiv \exists x \psi$: $\mathfrak{M}, s \models \varphi$ sse para pelo menos um **membro** $m \in |\mathfrak{M}|$, $\mathfrak{M}, s[m/x] \models \psi$.

explanation

As atribuições de variáveis são importantes nas últimas duas cláusulas. Não podemos definir a satisfação de $\forall x \psi(x)$ por “para todo $m \in |\mathfrak{M}|$, $\mathfrak{M} \models \psi(m)$ ”. Não podemos definir a satisfação de $\exists x \psi(x)$ por “para pelo menos um $m \in |\mathfrak{M}|$, $\mathfrak{M} \models \psi(m)$ ”. A razão é que, se $m \in |\mathfrak{M}|$, ele não é um símbolo da linguagem, e portanto $\psi(m)$ não é a **fórmula** (isto é, $\psi[m/x]$ é indefinido). Também não podemos supor que temos **símbolos de constante** ou termos disponíveis que nomeiem cada **membro** de $\mathfrak{M}$, já que não há nada na definição de **estruturas** que o exija. Na linguagem padrão, o conjunto de **símbolos de constante** é **infinito enumerável**, então se $|\mathfrak{M}|$ não é **enumerável** não há sequer **símbolos de constante** suficientes para nomear cada objeto.

Resolvemos esse problema introduzindo atribuições de variáveis, que nos permitem ligar variáveis diretamente a **membros** do domínio. Então, em vez de dizer que, por exemplo, $\exists x \psi(x)$ é satisfeita em $\mathfrak{M}$ sse para pelo menos um $m \in |\mathfrak{M}|$, dizemos que ela é satisfeita em $\mathfrak{M}$ *relativamente a* s sse $\psi(x)$ é satisfeita relativamente a $s[m/x]$ para pelo menos um $m \in |\mathfrak{M}|$.

16.4. SATISFAÇÃO DE A FÓRMULA EM A ESTRUTURA

Exemplo 16.12. Seja $\mathcal{L} = \{a, b, f, R\}$ onde a e b são **símbolos de constante**, f é um **símbolo de função** de dois lugares, e R é um **símbolo de predicado** de dois lugares. Considere a **estrutura** $\mathfrak{M}$ definida por:

1. $|\mathfrak{M}| = \{1, 2, 3, 4\}$
2. $a^{\mathfrak{M}} = 1$
3. $b^{\mathfrak{M}} = 2$
4. $f^{\mathfrak{M}}(x, y) = x + y$ se $x + y \leq 3$ e $= 3$ caso contrário.
5. $R^{\mathfrak{M}} = \{\langle 1, 1 \rangle, \langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 2, 4 \rangle\}$

A função $s(x) = 1$ que atribui $1 \in |\mathfrak{M}|$ a cada **variável** é uma atribuição de variáveis para $\mathfrak{M}$.

Então

$$\text{Val}_s^{\mathfrak{M}}(f(a, b)) = f^{\mathfrak{M}}(\text{Val}_s^{\mathfrak{M}}(a), \text{Val}_s^{\mathfrak{M}}(b)).$$

Como a e b são **símbolos de constante**, $\text{Val}_s^{\mathfrak{M}}(a) = a^{\mathfrak{M}} = 1$ e $\text{Val}_s^{\mathfrak{M}}(b) = b^{\mathfrak{M}} = 2$.
Então

$$\text{Val}_s^{\mathfrak{M}}(f(a, b)) = f^{\mathfrak{M}}(1, 2) = 1 + 2 = 3.$$

Para calcular o valor de $f(f(a, b), a)$ temos que considerar

$$\text{Val}_s^{\mathfrak{M}}(f(f(a, b), a)) = f^{\mathfrak{M}}(\text{Val}_s^{\mathfrak{M}}(f(a, b)), \text{Val}_s^{\mathfrak{M}}(a)) = f^{\mathfrak{M}}(3, 1) = 3,$$

já que $3 + 1 > 3$. Como $s(x) = 1$ e $\text{Val}_s^{\mathfrak{M}}(x) = s(x)$, também temos

$$\text{Val}_s^{\mathfrak{M}}(f(f(a, b), x)) = f^{\mathfrak{M}}(\text{Val}_s^{\mathfrak{M}}(f(a, b)), \text{Val}_s^{\mathfrak{M}}(x)) = f^{\mathfrak{M}}(3, 1) = 3,$$

Uma **fórmula** atômica $R(t_1, t_2)$ é satisfeita se a tupla de valores de seus argumentos, isto é, $\langle \text{Val}_s^{\mathfrak{M}}(t_1), \text{Val}_s^{\mathfrak{M}}(t_2) \rangle$, é **um membro** de $R^{\mathfrak{M}}$. Assim, por exemplo, temos $\mathfrak{M}, s \models R(b, f(a, b))$ já que $\langle \text{Val}_s^{\mathfrak{M}}(b), \text{Val}_s^{\mathfrak{M}}(f(a, b)) \rangle = \langle 2, 3 \rangle \in R^{\mathfrak{M}}$, mas $\mathfrak{M}, s \not\models R(x, f(a, b))$ já que $\langle 1, 3 \rangle \notin R^{\mathfrak{M}}[s]$.

Para determinar se uma fórmula não atômica φ é satisfeita, aplicam-se as cláusulas da definição indutiva que se aplicam ao conectivo principal. Por exemplo, o conectivo principal em $R(a, a) \rightarrow (R(b, x) \vee R(x, b))$ é o $\rightarrow$, e

$$\begin{aligned} \mathfrak{M}, s \models R(a, a) \rightarrow (R(b, x) \vee R(x, b)) \text{ sse} \\ \mathfrak{M}, s \not\models R(a, a) \text{ ou } \mathfrak{M}, s \models R(b, x) \vee R(x, b) \end{aligned}$$

Como $\mathfrak{M}, s \models R(a, a)$ (porque $\langle 1, 1 \rangle \in R^{\mathfrak{M}}$) ainda não podemos determinar a resposta e devemos primeiro descobrir se $\mathfrak{M}, s \models R(b, x) \vee R(x, b)$:

$$\begin{aligned} \mathfrak{M}, s \models R(b, x) \vee R(x, b) \text{ sse} \\ \mathfrak{M}, s \models R(b, x) \text{ ou } \mathfrak{M}, s \models R(x, b) \end{aligned}$$

CAPÍTULO 16. SEMÂNTICA DA LÓGICA DE PRIMEIRA ORDEM

E este é o caso, já que $\mathfrak{M}, s \models R(x, b)$ (porque $\langle 1, 2 \rangle \in R^{\mathfrak{M}}$).

Lembre-se de que uma x -variante de s é uma atribuição de variáveis que difere de s no máximo no que atribui a x . Para cada **membro** de $|\mathfrak{M}|$, há uma x -variante de s :

$$\begin{aligned} s_1 &= s[1/x], & s_2 &= s[2/x], \\ s_3 &= s[3/x], & s_4 &= s[4/x]. \end{aligned}$$

Assim, por exemplo, $s_2(x) = 2$ e $s_2(y) = s(y) = 1$ para todas as variáveis y diferentes de x . Estas são todas as x -variantes de s para a estrutura $\mathfrak{M}$, já que $|\mathfrak{M}| = \{1, 2, 3, 4\}$. Note, em particular, que $s_1 = s$ (s é sempre uma x -variante de si mesma).

Para determinar se uma **fórmula** quantificada existencialmente $\exists x \varphi(x)$ é satisfeita, temos que determinar se $\mathfrak{M}, s[m/x] \models \varphi(x)$ para pelo menos um $m \in |\mathfrak{M}|$. Assim,

$$\mathfrak{M}, s \models \exists x (R(b, x) \vee R(x, b)),$$

já que $\mathfrak{M}, s[1/x] \models R(b, x) \vee R(x, b)$ ($s[3/x]$ também serviria). Mas,

$$\mathfrak{M}, s \not\models \exists x (R(b, x) \wedge R(x, b))$$

já que, qualquer que seja o $m \in |\mathfrak{M}|$ que escolhamos, $\mathfrak{M}, s[m/x] \not\models R(b, x) \wedge R(x, b)$.

Para determinar se uma **fórmula** quantificada universalmente $\forall x \varphi(x)$ é satisfeita, temos que determinar se $\mathfrak{M}, s[m/x] \models \varphi(x)$ para todo $m \in |\mathfrak{M}|$. Assim,

$$\mathfrak{M}, s \models \forall x (R(x, a) \rightarrow R(a, x)),$$

já que $\mathfrak{M}, s[m/x] \models R(x, a) \rightarrow R(a, x)$ para todo $m \in |\mathfrak{M}|$. Para $m = 1$, temos $\mathfrak{M}, s[1/x] \models R(a, x)$ então o consequente é verdadeiro; para $m = 2, 3$, e 4 , temos $\mathfrak{M}, s[m/x] \not\models R(x, a)$, então o antecedente é falso. Mas,

$$\mathfrak{M}, s \not\models \forall x (R(a, x) \rightarrow R(x, a))$$

já que $\mathfrak{M}, s[2/x] \not\models R(a, x) \rightarrow R(x, a)$ (porque $\mathfrak{M}, s[2/x] \models R(a, x)$ e $\mathfrak{M}, s[2/x] \not\models R(x, a)$).

Para um caso mais complicado, considere

$$\forall x (R(a, x) \rightarrow \exists y R(x, y)).$$

Como $\mathfrak{M}, s[3/x] \not\models R(a, x)$ e $\mathfrak{M}, s[4/x] \not\models R(a, x)$, os casos interessantes em que temos que nos preocupar com o consequente do condicional são apenas $m = 1$ e $m = 2$. $\mathfrak{M}, s[1/x] \models \exists y R(x, y)$ vale? Vale se há pelo menos um $n \in |\mathfrak{M}|$ tal que $\mathfrak{M}, s[1/x][n/y] \models R(x, y)$. De fato, se tomarmos $n = 1$, temos $s[1/x][n/y] = s[1/y] = s$. Como $s(x) = 1$, $s(y) = 1$, e $\langle 1, 1 \rangle \in R^{\mathfrak{M}}$, a resposta é sim.

16.5. ATRIBUIÇÕES DE VARIÁVEIS

Para determinar se $\mathfrak{M}, s[2/x] \models \exists y R(x, y)$, temos que olhar para as atribuições de variáveis $s[2/x][n/y]$. Aqui, para $n = 1$, essa atribuição é $s_2 = s[2/x]$, que não satisfaz $R(x, y)$ ($s_2(x) = 2$, $s_2(y) = 1$, e $\langle 2, 1 \rangle \notin R^{\mathfrak{M}}$). No entanto, considere $s[2/x][3/y] = s_2[3/y]$. $\mathfrak{M}, s_2[3/y] \models R(x, y)$ já que $\langle 2, 3 \rangle \in R^{\mathfrak{M}}$, e assim $\mathfrak{M}, s_2 \models \exists y R(x, y)$.

Assim, para todo $n \in |\mathfrak{M}|$, ou $\mathfrak{M}, s[m/x] \not\models R(a, x)$ (se $m = 3, 4$) ou $\mathfrak{M}, s[m/x] \models \exists y R(x, y)$ (se $m = 1, 2$), e assim

$$\mathfrak{M}, s \models \forall x (R(a, x) \rightarrow \exists y R(x, y)).$$

Por outro lado,

$$\mathfrak{M}, s \not\models \exists x (R(a, x) \wedge \forall y R(x, y)).$$

Temos $\mathfrak{M}, s[m/x] \models R(a, x)$ apenas para $m = 1$ e $m = 2$. Mas para ambos esses valores de m , há por sua vez um $n \in |\mathfrak{M}|$, a saber $n = 4$, tal que $\mathfrak{M}, s[m/x][n/y] \not\models R(x, y)$ e assim $\mathfrak{M}, s[m/x] \not\models \forall y R(x, y)$ para $m = 1$ e $m = 2$. Em suma, não há $m \in |\mathfrak{M}|$ tal que $\mathfrak{M}, s[m/x] \models R(a, x) \wedge \forall y R(x, y)$.

Problema 16.2. Seja $\mathcal{L} = \{c, f, A\}$ com um **símbolo de constante**, um **símbolo de função** de um lugar e um **símbolo de predicado** de dois lugares, e seja a **estrutura** $\mathfrak{M}$ dada por

1. $|\mathfrak{M}| = \{1, 2, 3\}$
2. $c^{\mathfrak{M}} = 3$
3. $f^{\mathfrak{M}}(1) = 2, f^{\mathfrak{M}}(2) = 3, f^{\mathfrak{M}}(3) = 2$
4. $A^{\mathfrak{M}} = \{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 3 \rangle\}$

(a) Seja $s(v) = 1$ para todas as **variáveis** v . Descubra se

$$\mathfrak{M}, s \models \exists x (A(f(z), c) \rightarrow \forall y (A(y, x) \vee A(f(y), x)))$$

Explique por que sim ou por que não.

(b) Dê uma estrutura e uma atribuição de variáveis diferentes nas quais a **fórmula** não seja satisfeita.

content/first-order-logic/syntax-and-semantics/assignments.tex

16.5 Atribuições de Variáveis

fol:syn:ass:
sec

Uma atribuição de variáveis s fornece um valor para *toda* variável—e há infinitas delas. Isso, é claro, não é necessário. Exigimos que as atribuições de variáveis atribuam valores a todas as **variáveis** simplesmente porque isso torna as coisas muito mais fáceis. O valor de um termo t , e se a **fórmula** φ é satisfeita em a **estrutura** com respeito a s , dependem apenas das atribuições que s faz às **variáveis** em t e às **variáveis** livres de φ . Esse é o conteúdo das próximas

explanation

duas proposições. Para tornar precisa a ideia de “depende de”, mostramos que quaisquer duas atribuições de variáveis que concordam em todas as variáveis em t dão o mesmo valor, e que φ é satisfeita relativamente a uma sse é satisfeita relativamente à outra se duas atribuições de variáveis concordam em todas as variáveis livres de φ .

Proposição 16.13. *Se as **variáveis** em um termo t estão entre $x_1, \dots, x_n$, e $s_1(x_i) = s_2(x_i)$ para $i = 1, \dots, n$, então $\text{Val}_{s_1}^{\mathfrak{M}}(t) = \text{Val}_{s_2}^{\mathfrak{M}}(t)$.* fol:syn:ass:
prop:valindep

Prova. Por indução sobre a complexidade de t . Para o caso base, t pode ser **a símbolo de constante** ou uma das variáveis $x_1, \dots, x_n$. Se $t = c$, então $\text{Val}_{s_1}^{\mathfrak{M}}(t) = c^{\mathfrak{M}} = \text{Val}_{s_2}^{\mathfrak{M}}(t)$. Se $t = x_i$, $s_1(x_i) = s_2(x_i)$ pela hipótese da proposição, e assim $\text{Val}_{s_1}^{\mathfrak{M}}(t) = s_1(x_i) = s_2(x_i) = \text{Val}_{s_2}^{\mathfrak{M}}(t)$.

Para o passo indutivo, suponha que $t = f(t_1, \dots, t_k)$ e que a afirmação vale para $t_1, \dots, t_k$. Então

$$\begin{aligned} \text{Val}_{s_1}^{\mathfrak{M}}(t) &= \text{Val}_{s_1}^{\mathfrak{M}}(f(t_1, \dots, t_k)) \\ &= f^{\mathfrak{M}}(\text{Val}_{s_1}^{\mathfrak{M}}(t_1), \dots, \text{Val}_{s_1}^{\mathfrak{M}}(t_k)). \end{aligned}$$

Para $j = 1, \dots, k$, as **variáveis** de t_j estão entre $x_1, \dots, x_n$. Pela hipótese de indução, $\text{Val}_{s_1}^{\mathfrak{M}}(t_j) = \text{Val}_{s_2}^{\mathfrak{M}}(t_j)$. Assim,

$$\begin{aligned} \text{Val}_{s_1}^{\mathfrak{M}}(t) &= \text{Val}_{s_1}^{\mathfrak{M}}(f(t_1, \dots, t_k)) \\ &= f^{\mathfrak{M}}(\text{Val}_{s_1}^{\mathfrak{M}}(t_1), \dots, \text{Val}_{s_1}^{\mathfrak{M}}(t_k)) \\ &= f^{\mathfrak{M}}(\text{Val}_{s_2}^{\mathfrak{M}}(t_1), \dots, \text{Val}_{s_2}^{\mathfrak{M}}(t_k)) \\ &= \text{Val}_{s_2}^{\mathfrak{M}}(f(t_1, \dots, t_k)) = \text{Val}_{s_2}^{\mathfrak{M}}(t). \quad \square \end{aligned}$$

Proposição 16.14. *Se as **variáveis** livres em φ estão entre $x_1, \dots, x_n$, e $s_1(x_i) = s_2(x_i)$ para $i = 1, \dots, n$, então $\mathfrak{M}, s_1 \models \varphi$ sse $\mathfrak{M}, s_2 \models \varphi$.* fol:syn:ass:
prop:satindep

Prova. Usamos indução sobre a complexidade de φ . Para o caso base, onde φ é atômica, φ pode ser: $\top$, $\perp$, $R(t_1, \dots, t_k)$ para um predicado R de k lugares e termos $t_1, \dots, t_k$, ou $t_1 = t_2$ para termos t_1 e t_2 . Nos dois últimos casos, demonstramos apenas a direção direta do **bicondicional**, já que a prova da recíproca é simétrica.

1. $\varphi \equiv \top$: tanto $\mathfrak{M}, s_1 \models \varphi$ quanto $\mathfrak{M}, s_2 \models \varphi$.
2. $\varphi \equiv \perp$: tanto $\mathfrak{M}, s_1 \not\models \varphi$ quanto $\mathfrak{M}, s_2 \not\models \varphi$.
3. $\varphi \equiv R(t_1, \dots, t_k)$: seja $\mathfrak{M}, s_1 \models \varphi$. Então

$$\langle \text{Val}_{s_1}^{\mathfrak{M}}(t_1), \dots, \text{Val}_{s_1}^{\mathfrak{M}}(t_k) \rangle \in R^{\mathfrak{M}}.$$

Para $i = 1, \dots, k$, $\text{Val}_{s_1}^{\mathfrak{M}}(t_i) = \text{Val}_{s_2}^{\mathfrak{M}}(t_i)$ por **Proposição 16.13**. Então também temos $\langle \text{Val}_{s_2}^{\mathfrak{M}}(t_1), \dots, \text{Val}_{s_2}^{\mathfrak{M}}(t_k) \rangle \in R^{\mathfrak{M}}$, e portanto $\mathfrak{M}, s_2 \models \varphi$.

16.5. ATRIBUIÇÕES DE VARIÁVEIS

4. $\varphi \equiv t_1 = t_2$: suponha $\mathfrak{M}, s_1 \models \varphi$. Então $\text{Val}_{s_1}^{\mathfrak{M}}(t_1) = \text{Val}_{s_1}^{\mathfrak{M}}(t_2)$. Assim,

$$\begin{aligned} \text{Val}_{s_2}^{\mathfrak{M}}(t_1) &= \text{Val}_{s_1}^{\mathfrak{M}}(t_1) && \text{(por Proposição 16.13)} \\ &= \text{Val}_{s_1}^{\mathfrak{M}}(t_2) && \text{(já que } \mathfrak{M}, s_1 \models t_1 = t_2 \text{)} \\ &= \text{Val}_{s_2}^{\mathfrak{M}}(t_2) && \text{(por Proposição 16.13),} \end{aligned}$$

então $\mathfrak{M}, s_2 \models t_1 = t_2$.

Agora suponha $\mathfrak{M}, s_1 \models \psi$ sse $\mathfrak{M}, s_2 \models \psi$ para todas as fórmulas ψ menos complexas que φ . O passo de indução procede por casos determinados pelo operador principal de φ . Em cada caso, demonstramos apenas a direção direta do **bicondicional**; a prova da direção reversa é simétrica. Em todos os casos exceto os dos quantificadores, aplicamos a hipótese de indução às sub-fórmulas ψ de φ . As variáveis livres de ψ estão entre as de φ . Assim, se s_1 e s_2 concordam nas variáveis livres de φ , elas também concordam nas de ψ , e a hipótese de indução se aplica a ψ .

1. $\varphi \equiv \neg\psi$: se $\mathfrak{M}, s_1 \models \varphi$, então $\mathfrak{M}, s_1 \not\models \psi$, então pela hipótese de indução, $\mathfrak{M}, s_2 \not\models \psi$, portanto $\mathfrak{M}, s_2 \models \varphi$.
2. $\varphi \equiv \psi \wedge \chi$: se $\mathfrak{M}, s_1 \models \varphi$, então $\mathfrak{M}, s_1 \models \psi$ e $\mathfrak{M}, s_1 \models \chi$, então pela hipótese de indução, $\mathfrak{M}, s_2 \models \psi$ e $\mathfrak{M}, s_2 \models \chi$. Portanto, $\mathfrak{M}, s_2 \models \varphi$.
3. $\varphi \equiv \psi \vee \chi$: se $\mathfrak{M}, s_1 \models \varphi$, então $\mathfrak{M}, s_1 \models \psi$ ou $\mathfrak{M}, s_1 \models \chi$. Pela hipótese de indução, $\mathfrak{M}, s_2 \models \psi$ ou $\mathfrak{M}, s_2 \models \chi$, então $\mathfrak{M}, s_2 \models \varphi$.
4. $\varphi \equiv \psi \rightarrow \chi$: se $\mathfrak{M}, s_1 \models \varphi$, então $\mathfrak{M}, s_1 \not\models \psi$ ou $\mathfrak{M}, s_1 \models \chi$. Pela hipótese de indução, $\mathfrak{M}, s_2 \not\models \psi$ ou $\mathfrak{M}, s_2 \models \chi$, então $\mathfrak{M}, s_2 \models \varphi$.
5. $\varphi \equiv \psi \leftrightarrow \chi$: se $\mathfrak{M}, s_1 \models \varphi$, então ou $\mathfrak{M}, s_1 \models \psi$ e $\mathfrak{M}, s_1 \models \chi$, ou $\mathfrak{M}, s_1 \not\models \psi$ e $\mathfrak{M}, s_1 \not\models \chi$. Pela hipótese de indução, ou $\mathfrak{M}, s_2 \models \psi$ e $\mathfrak{M}, s_2 \models \chi$ ou $\mathfrak{M}, s_2 \not\models \psi$ e $\mathfrak{M}, s_2 \not\models \chi$. Em qualquer caso, $\mathfrak{M}, s_2 \models \varphi$.
6. $\varphi \equiv \exists x \psi$: se $\mathfrak{M}, s_1 \models \varphi$, há um $m \in |\mathfrak{M}|$ tal que $\mathfrak{M}, s_1[m/x] \models \psi$. Sejam $s'_1 = s_1[m/x]$ e $s'_2 = s_2[m/x]$. As variáveis livres de ψ estão entre $x_1, \dots, x_n$, e x . $s'_1(x_i) = s'_2(x_i)$, já que s'_1 e s'_2 são x -variantes de s_1 e s_2 , respectivamente, e por hipótese $s_1(x_i) = s_2(x_i)$. $s'_1(x) = s'_2(x) = m$ pela forma como definimos s'_1 e s'_2 . Então a hipótese de indução se aplica a ψ e s'_1, s'_2 , então $\mathfrak{M}, s'_2 \models \psi$. Portanto, já que $s'_2 = s_2[m/x]$, há um $m \in |\mathfrak{M}|$ tal que $\mathfrak{M}, s_2[m/x] \models \psi$, e assim $\mathfrak{M}, s_2 \models \varphi$.
7. $\varphi \equiv \forall x \psi$: se $\mathfrak{M}, s_1 \models \varphi$, então para todo $m \in |\mathfrak{M}|$, $\mathfrak{M}, s_1[m/x] \models \psi$. Queremos mostrar que também, para todo $m \in |\mathfrak{M}|$, $\mathfrak{M}, s_2[m/x] \models \psi$. Então seja $m \in |\mathfrak{M}|$ arbitrário, e considere $s'_1 = s_1[m/x]$ e $s'_2 = s_2[m/x]$. Temos que $\mathfrak{M}, s'_1 \models \psi$. As variáveis livres de ψ estão entre $x_1, \dots, x_n$, e x . $s'_1(x_i) = s'_2(x_i)$, já que s'_1 e s'_2 são x -variantes de s_1 e s_2 , respectivamente, e por hipótese $s_1(x_i) = s_2(x_i)$. $s'_1(x) = s'_2(x) = m$ pela forma como definimos s'_1 e s'_2 . Então a hipótese de indução se aplica a ψ e s'_1, s'_2 , e

CAPÍTULO 16. SEMÂNTICA DA LÓGICA DE PRIMEIRA ORDEM

temos $\mathfrak{M}, s'_2 \models \psi$. Isso se aplica a todo $m \in |\mathfrak{M}|$, isto é, $\mathfrak{M}, s_2[m/x] \models \psi$ para todo $m \in |\mathfrak{M}|$, então $\mathfrak{M}, s_2 \models \varphi$.

Por indução, obtemos que $\mathfrak{M}, s_1 \models \varphi$ sse $\mathfrak{M}, s_2 \models \varphi$ sempre que as **variáveis** livres em φ estão entre $x_1, \dots, x_n$ e $s_1(x_i) = s_2(x_i)$ para $i = 1, \dots, n$. $\square$

Problema 16.3. Complete a prova de **Proposição 16.14**.

explanation **Sentenças** não têm variáveis livres, então quaisquer duas atribuições de variáveis atribuem as mesmas coisas a todas as (zero) variáveis livres de qualquer sentença. A proposição que acabamos de provar significa, então, que se **a sentença** é satisfeita ou não em uma estrutura relativamente a uma atribuição de variáveis é completamente independente da atribuição. Registraremos esse fato. Ele justifica a definição de satisfação de **a sentença** em **a estrutura** (sem mencionar uma atribuição de variáveis) que segue.

Corolário 16.15. *Se φ é **a sentença** e s uma atribuição de variáveis, então $\mathfrak{M}, s \models \varphi$ sse $\mathfrak{M}, s' \models \varphi$ para toda atribuição de variáveis s' .*

fol:syn:ass:
cor:sat-sentence

Prova. Seja s' uma atribuição de variáveis qualquer. Como φ é **a sentença**, ela não tem variáveis livres, e assim toda atribuição de variáveis s' trivialmente atribui as mesmas coisas a todas as variáveis livres de φ que faz s . Então a condição de **Proposição 16.14** é satisfeita, e temos $\mathfrak{M}, s \models \varphi$ sse $\mathfrak{M}, s' \models \varphi$. $\square$

Definição 16.16. Se φ é **a sentença**, dizemos que **a estrutura** $\mathfrak{M}$ *satisfaz* φ , $\mathfrak{M} \models \varphi$, sse $\mathfrak{M}, s \models \varphi$ para todas as atribuições de variáveis s .

fol:syn:ass:
defn:satisfaction

Se $\mathfrak{M} \models \varphi$, também dizemos simplesmente que φ é *verdadeira em* $\mathfrak{M}$. A noção de satisfação se estende naturalmente de **sentenças** individuais para conjuntos de **sentenças**.

Definição 16.17. Se Γ é um conjunto de **sentenças** Γ , dizemos que **a estrutura** $\mathfrak{M}$ *satisfaz* Γ , $\mathfrak{M} \models \Gamma$, sse $\mathfrak{M} \models \varphi$ para todo $\varphi \in \Gamma$.

fol:syn:ass:
defn:sat

Proposição 16.18. *Seja $\mathfrak{M}$ **a estrutura**, φ **a sentença**, e s uma atribuição de variáveis. $\mathfrak{M} \models \varphi$ sse $\mathfrak{M}, s \models \varphi$.*

fol:syn:ass:
prop:sentence-sat-true

Prova. Exercício. $\square$

Problema 16.4. Prove **Proposição 16.18**

Proposição 16.19. *Suponha que $\varphi(x)$ contém apenas x livre, e $\mathfrak{M}$ é **a estrutura**. Então:*

fol:syn:ass:
prop:sat-quant

1. $\mathfrak{M} \models \exists x \varphi(x)$ sse $\mathfrak{M}, s \models \varphi(x)$ para pelo menos uma atribuição de variáveis s .
2. $\mathfrak{M} \models \forall x \varphi(x)$ sse $\mathfrak{M}, s \models \varphi(x)$ para todas as atribuições de variáveis s .

Prova. Exercício. $\square$

16.5. ATRIBUIÇÕES DE VARIÁVEIS

Problema 16.5. Prove [Proposição 16.19](#).

Problema 16.6. Suponha que $\mathcal{L}$ é uma linguagem sem [símbolos de função](#). Dada uma [estrutura](#) $\mathfrak{M}$, c a [símbolo de constante](#) e $a \in |\mathfrak{M}|$, defina $\mathfrak{M}[a/c]$ como a [estrutura](#) que é exatamente como $\mathfrak{M}$, exceto que $c^{\mathfrak{M}[a/c]} = a$. Defina $\mathfrak{M} \models \varphi$ para [sentenças](#) φ por:

1. $\varphi \equiv \perp$: não $\mathfrak{M} \models \varphi$.
2. $\varphi \equiv \top$: $\mathfrak{M} \models \varphi$.
3. $\varphi \equiv R(d_1, \dots, d_n)$: $\mathfrak{M} \models \varphi$ sse $\langle d_1^{\mathfrak{M}}, \dots, d_n^{\mathfrak{M}} \rangle \in R^{\mathfrak{M}}$.
4. $\varphi \equiv d_1 = d_2$: $\mathfrak{M} \models \varphi$ sse $d_1^{\mathfrak{M}} = d_2^{\mathfrak{M}}$.
5. $\varphi \equiv \neg\psi$: $\mathfrak{M} \models \varphi$ sse não $\mathfrak{M} \models \psi$.
6. $\varphi \equiv (\psi \wedge \chi)$: $\mathfrak{M} \models \varphi$ sse $\mathfrak{M} \models \psi$ e $\mathfrak{M} \models \chi$.
7. $\varphi \equiv (\psi \vee \chi)$: $\mathfrak{M} \models \varphi$ sse $\mathfrak{M} \models \psi$ ou $\mathfrak{M} \models \chi$ (ou ambos).
8. $\varphi \equiv (\psi \rightarrow \chi)$: $\mathfrak{M} \models \varphi$ sse não $\mathfrak{M} \models \psi$ ou $\mathfrak{M} \models \chi$ (ou ambos).
9. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\mathfrak{M} \models \varphi$ sse ou ambos $\mathfrak{M} \models \psi$ e $\mathfrak{M} \models \chi$, ou nem $\mathfrak{M} \models \psi$ nem $\mathfrak{M} \models \chi$.
10. $\varphi \equiv \forall x \psi$: $\mathfrak{M} \models \varphi$ sse para todo $a \in |\mathfrak{M}|$, $\mathfrak{M}[a/c] \models \psi[c/x]$, se c não ocorre em ψ .
11. $\varphi \equiv \exists x \psi$: $\mathfrak{M} \models \varphi$ sse há um $a \in |\mathfrak{M}|$ tal que $\mathfrak{M}[a/c] \models \psi[c/x]$, se c não ocorre em ψ .

Sejam $x_1, \dots, x_n$ todas as [variáveis](#) livres em φ , $c_1, \dots, c_n$ símbolos de constante não presentes em φ , $a_1, \dots, a_n \in |\mathfrak{M}|$, e $s(x_i) = a_i$.

Mostre que $\mathfrak{M}, s \models \varphi$ sse $\mathfrak{M}[a_1/c_1, \dots, a_n/c_n] \models \varphi[c_1/x_1] \dots [c_n/x_n]$.

(Este problema mostra que é possível dar uma semântica para a lógica de primeira ordem que dispensa atribuições de variáveis.)

Problema 16.7. Suponha que f é um símbolo de função não presente em $\varphi(x, y)$. Mostre que há a [estrutura](#) $\mathfrak{M}$ tal que $\mathfrak{M} \models \forall x \exists y \varphi(x, y)$ sse há uma $\mathfrak{M}'$ tal que $\mathfrak{M}' \models \forall x \varphi(x, f(x))$.

(Este problema é um caso especial do que é conhecido como Teorema de Skolem; $\forall x \varphi(x, f(x))$ é chamado de *forma normal de Skolem* de $\forall x \exists y \varphi(x, y)$.)

16.6 Extensionalidade

explanation A extensionalidade, às vezes chamada de relevância, pode ser expressa informalmente da seguinte forma: os únicos fatores que influenciam a satisfação de fórmula φ em a estrutura $\mathfrak{M}$ em relação a uma atribuição de variáveis s são o tamanho do domínio e as atribuições feitas por $\mathfrak{M}$ e s aos elementos da linguagem que de fato aparecem em φ . fol:syn:ext:sec

Uma consequência imediata da extensionalidade é que, quando duas estruturas $\mathfrak{M}$ e $\mathfrak{M}'$ concordam em todos os elementos da linguagem que aparecem em uma sentença φ e têm o mesmo domínio, $\mathfrak{M}$ e $\mathfrak{M}'$ também devem concordar quanto a se φ em si é verdadeira ou não.

Proposição 16.20 (Extensionalidade). *Seja φ a fórmula, e $\mathfrak{M}_1$ e $\mathfrak{M}_2$ estruturas com $|\mathfrak{M}_1| = |\mathfrak{M}_2|$, e s uma atribuição de variáveis sobre $|\mathfrak{M}_1| = |\mathfrak{M}_2|$. Se $c^{\mathfrak{M}_1} = c^{\mathfrak{M}_2}$, $R^{\mathfrak{M}_1} = R^{\mathfrak{M}_2}$, e $f^{\mathfrak{M}_1} = f^{\mathfrak{M}_2}$ para todo símbolo de constante c , símbolo de relação R , e símbolo de função f que ocorre em φ , então $\mathfrak{M}_1, s \models \varphi$ sse $\mathfrak{M}_2, s \models \varphi$.* fol:syn:ext:prop:extensionality

Prova. Primeiro prove (por indução sobre t) que, para todo termo, $\text{Val}_s^{\mathfrak{M}_1}(t) = \text{Val}_s^{\mathfrak{M}_2}(t)$. Então prove a proposição por indução sobre φ , fazendo uso da afirmação que acabamos de provar para a base da indução (onde φ é atômica).□

Problema 16.8. Faça a prova de **Proposição 16.20** em detalhe.

Corolário 16.21 (Extensionalidade para Sentenças). *Seja φ a sentença e $\mathfrak{M}_1, \mathfrak{M}_2$ como em **Proposição 16.20**. Então $\mathfrak{M}_1 \models \varphi$ sse $\mathfrak{M}_2 \models \varphi$.* fol:syn:ext:cor:extensionality-sent

Prova. Segue de **Proposição 16.20** por **Corolário 16.15**. □

Além disso, o valor de um termo, e se a estrutura satisfaz ou não a fórmula, dependem apenas dos valores de seus subtermos.

Proposição 16.22. *Seja $\mathfrak{M}$ a estrutura, t e t' termos, e s uma atribuição de variáveis. Então $\text{Val}_s^{\mathfrak{M}}(t[t'/x]) = \text{Val}_{s[\text{Val}_s^{\mathfrak{M}}(t')/x]}^{\mathfrak{M}}(t)$.* fol:syn:ext:prop:ext-terms

Prova. Por indução sobre t .

1. Se t é uma constante, digamos, $t \equiv c$, então $t[t'/x] = c$, e $\text{Val}_s^{\mathfrak{M}}(c) = c^{\mathfrak{M}} = \text{Val}_{s[\text{Val}_s^{\mathfrak{M}}(t')/x]}^{\mathfrak{M}}(c)$.
2. Se t é uma variável diferente de x , digamos, $t \equiv y$, então $t[t'/x] = y$, e $\text{Val}_s^{\mathfrak{M}}(y) = \text{Val}_{s[\text{Val}_s^{\mathfrak{M}}(t')/x]}^{\mathfrak{M}}(y)$ já que $s \sim_x s[\text{Val}_s^{\mathfrak{M}}(t')/x]$.
3. Se $t \equiv x$, então $t[t'/x] = t'$. Mas $\text{Val}_{s[\text{Val}_s^{\mathfrak{M}}(t')/x]}^{\mathfrak{M}}(x) = \text{Val}_s^{\mathfrak{M}}(t')$ pela definição de $s[\text{Val}_s^{\mathfrak{M}}(t')/x]$.

16.7. NOÇÕES SEMÂNTICAS

4. Se $t \equiv f(t_1, \dots, t_n)$ então temos:

$$\begin{aligned}
 \text{Val}_s^{\mathfrak{M}}(t[t'/x]) &= \\
 &= \text{Val}_s^{\mathfrak{M}}(f(t_1[t'/x], \dots, t_n[t'/x])) \\
 &\quad \text{pela definição de } t[t'/x] \\
 &= f^{\mathfrak{M}}(\text{Val}_s^{\mathfrak{M}}(t_1[t'/x]), \dots, \text{Val}_s^{\mathfrak{M}}(t_n[t'/x])) \\
 &\quad \text{pela definição de } \text{Val}_s^{\mathfrak{M}}(f(\dots)) \\
 &= f^{\mathfrak{M}}(\text{Val}_{s[\text{Val}_s^{\mathfrak{M}}(t')/x]}^{\mathfrak{M}}(t_1), \dots, \text{Val}_{s[\text{Val}_s^{\mathfrak{M}}(t')/x]}^{\mathfrak{M}}(t_n)) \\
 &\quad \text{pela hipótese de indução} \\
 &= \text{Val}_{s[\text{Val}_s^{\mathfrak{M}}(t')/x]}^{\mathfrak{M}}(t) \text{ pela definição de } \text{Val}_{s[\text{Val}_s^{\mathfrak{M}}(t')/x]}^{\mathfrak{M}}(f(\dots)) \quad \square
 \end{aligned}$$

fol:syn:ext:
prop:ext-formulas

Proposição 16.23. *Seja $\mathfrak{M}$ a estrutura, φ a fórmula, t' um termo, e s uma atribuição de variáveis. Então $\mathfrak{M}, s \models \varphi[t'/x]$ sse $\mathfrak{M}, s[\text{Val}_s^{\mathfrak{M}}(t')/x] \models \varphi$.*

Prova. Exercício. $\square$

Problema 16.9. Prove [Proposição 16.23](#)

O ponto de [Proposições 16.22](#) e [16.23](#) é o seguinte. Suponha que temos um termo t ou a fórmula φ e algum termo t' , e queremos saber o valor de $t[t'/x]$ ou se $\varphi[t'/x]$ é ou não satisfeita em a estrutura $\mathfrak{M}$ em relação a uma atribuição de variáveis s . Então podemos ou realizar a substituição primeiro e depois considerar o valor ou a satisfação em relação a $\mathfrak{M}$ e s , ou podemos primeiro determinar o valor $m = \text{Val}_s^{\mathfrak{M}}(t')$ de t' em $\mathfrak{M}$ em relação a s , mudar a atribuição de variáveis para $s[m/x]$ e então considerar o valor de t em $\mathfrak{M}$ e $s[m/x]$, ou se $\mathfrak{M}, s[m/x] \models \varphi$. [Proposições 16.22](#) e [16.23](#) garantem que a resposta será a mesma, independentemente de como o fazamos.

explanation

content/first-order-logic/syntax-and-semantics/semantic-notions.tex

16.7 Noções Semânticas

fol:syn:sem:
sec

Dada a definição de estruturas para linguagens de primeira ordem, podemos definir algumas propriedades semânticas básicas de sentenças e relações entre elas. A mais simples destas é a noção de *validade* de uma sentença. Uma sentença é válida se é satisfeita em toda estrutura. Sentenças válidas são aquelas que são satisfeitas independentemente de como os símbolos não lógicos nela são interpretados. Sentenças válidas são, portanto, também chamadas de *verdades lógicas*—elas são verdadeiras, isto é, satisfeitas, em qualquer estrutura e, portanto, sua verdade depende apenas dos símbolos lógicos que nelas ocorrem e de sua estrutura sintática, mas não dos símbolos não lógicos ou de sua interpretação.

explanation

Definição 16.24 (Validade). Uma sentença φ é *válida*, $\models \varphi$, sse $\mathfrak{M} \models \varphi$ para toda *estrutura* $\mathfrak{M}$.

Definição 16.25 (Consequência). Uma sentença φ é *consequência* de um conjunto de sentenças Γ , $\Gamma \models \varphi$, sse para toda *estrutura* $\mathfrak{M}$ com $\mathfrak{M} \models \Gamma$, $\mathfrak{M} \models \varphi$.

Definição 16.26 (Satisfatibilidade). Um conjunto de sentenças Γ é *satisfatível* se $\mathfrak{M} \models \Gamma$ para alguma *estrutura* $\mathfrak{M}$. Se Γ não é satisfatível ele é chamado de *insatisfatível*.

Proposição 16.27. Uma sentença φ é *válida* sse $\Gamma \models \varphi$ para todo conjunto de sentenças Γ .

Prova. Para a direção direta, seja φ válida, e seja Γ um conjunto de sentenças. Seja $\mathfrak{M}$ a *estrutura* tal que $\mathfrak{M} \models \Gamma$. Como φ é válida, $\mathfrak{M} \models \varphi$, portanto $\Gamma \models \varphi$.

Para a contrapositiva da direção reversa, seja φ inválida, então há a *estrutura* $\mathfrak{M}$ com $\mathfrak{M} \not\models \varphi$. Quando $\Gamma = \{\top\}$, como $\top$ é válida, $\mathfrak{M} \models \Gamma$. Portanto, há a *estrutura* $\mathfrak{M}$ tal que $\mathfrak{M} \models \Gamma$ mas $\mathfrak{M} \not\models \varphi$, portanto φ não é consequência de Γ . $\square$

Proposição 16.28. $\Gamma \models \varphi$ sse $\Gamma \cup \{\neg\varphi\}$ é *insatisfatível*.

*fol:syn:sem:
prop:entails-unsat*

Prova. Para a direção direta, suponha $\Gamma \models \varphi$ e suponha, ao contrário, que há a *estrutura* $\mathfrak{M}$ tal que $\mathfrak{M} \models \Gamma \cup \{\neg\varphi\}$. Como $\mathfrak{M} \models \Gamma$ e $\Gamma \models \varphi$, $\mathfrak{M} \models \varphi$. Além disso, como $\mathfrak{M} \models \Gamma \cup \{\neg\varphi\}$, $\mathfrak{M} \models \neg\varphi$, então temos tanto $\mathfrak{M} \models \varphi$ quanto $\mathfrak{M} \not\models \varphi$, uma contradição. Portanto, não pode haver tal *estrutura* $\mathfrak{M}$, então $\Gamma \cup \{\neg\varphi\}$ é *insatisfatível*.

Para a direção reversa, suponha que $\Gamma \cup \{\neg\varphi\}$ é *insatisfatível*. Então, para toda *estrutura* $\mathfrak{M}$, ou $\mathfrak{M} \not\models \Gamma$ ou $\mathfrak{M} \models \varphi$. Portanto, para toda *estrutura* $\mathfrak{M}$ com $\mathfrak{M} \models \Gamma$, $\mathfrak{M} \models \varphi$, então $\Gamma \models \varphi$. $\square$

Problema 16.10. 1. Mostre que $\Gamma \models \perp$ sse Γ é *insatisfatível*.

2. Mostre que $\Gamma \cup \{\varphi\} \models \perp$ sse $\Gamma \models \neg\varphi$.

3. Suponha que c não ocorre em φ nem em Γ . Mostre que $\Gamma \models \forall x \varphi$ sse $\Gamma \models \varphi[c/x]$.

Proposição 16.29. Se $\Gamma \subseteq \Gamma'$ e $\Gamma \models \varphi$, então $\Gamma' \models \varphi$.

Prova. Suponha que $\Gamma \subseteq \Gamma'$ e $\Gamma \models \varphi$. Seja $\mathfrak{M}$ uma *estrutura* tal que $\mathfrak{M} \models \Gamma'$; então $\mathfrak{M} \models \Gamma$, e como $\Gamma \models \varphi$, obtemos que $\mathfrak{M} \models \varphi$. Portanto, sempre que $\mathfrak{M} \models \Gamma'$, $\mathfrak{M} \models \varphi$, então $\Gamma' \models \varphi$. $\square$

Teorema 16.30 (Teorema da Dedução Semântica). $\Gamma \cup \{\varphi\} \models \psi$ sse $\Gamma \models \varphi \rightarrow \psi$.

*fol:syn:sem:
thm:sem-deduction*

Prova. Para a direção direta, seja $\Gamma \cup \{\varphi\} \models \psi$ e seja $\mathfrak{M}$ a estrutura tal que $\mathfrak{M} \models \Gamma$. Se $\mathfrak{M} \models \varphi$, então $\mathfrak{M} \models \Gamma \cup \{\varphi\}$, então, como ψ é consequência de $\Gamma \cup \{\varphi\}$, obtemos $\mathfrak{M} \models \psi$. Portanto, $\mathfrak{M} \models \varphi \rightarrow \psi$, então $\Gamma \models \varphi \rightarrow \psi$.

Para a direção reversa, seja $\Gamma \models \varphi \rightarrow \psi$ e $\mathfrak{M}$ a estrutura tal que $\mathfrak{M} \models \Gamma \cup \{\varphi\}$. Então $\mathfrak{M} \models \Gamma$, então $\mathfrak{M} \models \varphi \rightarrow \psi$, e como $\mathfrak{M} \models \varphi$, $\mathfrak{M} \models \psi$. Portanto, sempre que $\mathfrak{M} \models \Gamma \cup \{\varphi\}$, $\mathfrak{M} \models \psi$, então $\Gamma \cup \{\varphi\} \models \psi$. $\square$

*fol:syn:sem:
prop:quant-terms*

Proposição 16.31. *Seja $\mathfrak{M}$ a estrutura, e $\varphi(x)$ a fórmula com uma variável livre x , e t um termo fechado. Então:*

1. $\varphi(t) \models \exists x \varphi(x)$
2. $\forall x \varphi(x) \models \varphi(t)$

Prova. 1. Suponha $\mathfrak{M} \models \varphi(t)$. Seja s uma atribuição de variáveis com $s(x) = \text{Val}^{\mathfrak{M}}(t)$. Então $\mathfrak{M}, s \models \varphi(t)$ já que $\varphi(t)$ é a sentença. Por **Proposição 16.23**, $\mathfrak{M}, s \models \varphi(x)$. Por **Proposição 16.19**, $\mathfrak{M} \models \exists x \varphi(x)$.

2. Suponha $\mathfrak{M} \models \forall x \varphi(x)$. Seja s uma atribuição de variáveis com $s(x) = \text{Val}^{\mathfrak{M}}(t)$. Por **Proposição 16.19**, $\mathfrak{M}, s \models \varphi(x)$. Por **Proposição 16.23**, $\mathfrak{M}, s \models \varphi(t)$. Por **Proposição 16.18**, $\mathfrak{M} \models \varphi(t)$ já que $\varphi(t)$ é a sentença. $\square$

Problema 16.11. Complete a prova de **Proposição 16.31**.

Capítulo 17

Teorias e Seus Modelos

`content/first-order-logic/models-theories/introduction.tex`

17.1 Introdução

*fol:mat:int:
sec*

O desenvolvimento do método axiomático é uma conquista significativa na história da ciência, e é de especial importância na história da matemática. Um desenvolvimento axiomático de um campo envolve o esclarecimento de muitas questões: Do que trata o campo? Quais são os conceitos mais fundamentais?

explanation

Como eles se relacionam? Todos os conceitos do campo podem ser definidos em termos desses conceitos fundamentais? Que leis esses conceitos obedecem, e devem obedecer?

O método axiomático e a lógica foram feitos um para o outro. A lógica formal fornece as ferramentas para formular teorias axiomáticas, para provar teoremas a partir dos axiomas da teoria de uma maneira precisamente especificada, para estudar as propriedades de todos os sistemas que satisfazem os axiomas de uma maneira sistemática.

Definição 17.1. Um conjunto de **sentenças** Γ é *fechado* sse, sempre que $\Gamma \models \varphi$ então $\varphi \in \Gamma$. O *fecho* de um conjunto de **sentenças** Γ é $\{\varphi : \Gamma \models \varphi\}$.

Dizemos que Γ é *axiomatizado por* um conjunto de sentenças Δ se Γ é o fecho de Δ .

explanation

Podemos pensar em uma teoria axiomática como o conjunto de sentenças que é axiomatizado por seu conjunto de axiomas Δ . Em outras palavras, quando temos uma linguagem de primeira ordem que contém símbolos não lógicos para os primitivos da ciência desenvolvida axiomáticamente que desejamos estudar, juntamente com um conjunto de **sentenças** que expressam as leis fundamentais da ciência, podemos pensar na teoria como representada por todas as **sentenças** nessa linguagem que são consequência dos axiomas. Isso varia de exemplos simples com apenas um único primitivo e axiomas simples, como a teoria das ordens parciais, até teorias complexas como a mecânica newtoniana.

Os fatos lógicos importantes que tornam essa abordagem formal do método axiomático tão importante são os seguintes. Suponha que Γ é um sistema de axiomas para uma teoria, isto é, um conjunto de sentenças.

1. Podemos afirmar precisamente quando um sistema de axiomas captura uma classe pretendida de **estruturas**. Isto é, se estamos interessados em uma certa classe de **estruturas**, capturaremos com sucesso essa classe por um sistema de axiomas Γ sse as **estruturas** são exatamente aquelas $\mathfrak{M}$ tais que $\mathfrak{M} \models \Gamma$.
2. Podemos falhar nesse aspecto porque há $\mathfrak{M}$ tais que $\mathfrak{M} \models \Gamma$, mas $\mathfrak{M}$ não é uma das **estruturas** que pretendemos. Isso pode nos levar a adicionar axiomas que não são verdadeiros em $\mathfrak{M}$.
3. Se somos bem-sucedidos pelo menos no aspecto de que Γ é verdadeiro em todas as **estruturas** pretendidas, então uma sentença φ é verdadeira em todas as **estruturas** pretendidas sempre que $\Gamma \models \varphi$. Assim, podemos usar ferramentas lógicas (como métodos de **derivação**) para mostrar que sentenças são verdadeiras em todas as **estruturas** pretendidas simplesmente mostrando que elas são consequência dos axiomas.
4. Às vezes não temos **estruturas** pretendidas em mente, mas em vez disso partimos dos próprios axiomas: começamos com alguns primitivos que queremos que satisfaçam certas leis que codificamos em um sistema de axiomas. Uma coisa que gostaríamos de verificar imediatamente é que os

17.2. EXPRESSANDO PROPRIEDADES DE ESTRUTURAS

axiomas não se contradizem: se o fizerem, não pode haver conceitos que obedeçam a essas leis, e teremos tentado estabelecer uma teoria incoerente. Podemos verificar que isso não acontece encontrando um modelo de Γ . E se há modelos da nossa teoria, podemos usar métodos lógicos para investigá-los, e também podemos usar métodos lógicos para construir modelos.

5. A independência dos axiomas é, igualmente, uma questão importante. Pode acontecer que um dos axiomas seja, na verdade, uma consequência dos outros, e seja, portanto, redundante. Podemos provar que um axioma φ em Γ é redundante provando $\Gamma \setminus \{\varphi\} \models \varphi$. Também podemos provar que um axioma não é redundante mostrando que $(\Gamma \setminus \{\varphi\}) \cup \{\neg\varphi\}$ é satisfatível. Por exemplo, foi assim que se mostrou que o postulado das paralelas é independente dos outros axiomas da geometria.
6. Outra questão importante é a da definibilidade de conceitos em uma teoria: A escolha da linguagem determina em que consistem os modelos de uma teoria. Mas nem todo aspecto de uma teoria deve ser representado separadamente em seus modelos. Por exemplo, toda ordenação $\leq$ determina uma ordenação estrita correspondente $<$ —dada uma, nós podemos definir a outra. Então não é necessário que um modelo de uma teoria que envolve tal ordem deva conter *também* a ordenação estrita correspondente. Quando é o caso, em geral, de que uma relação pode ser definida em termos de outras? Quando é impossível definir uma relação em termos de outras (e, portanto, deve-se adicioná-la aos primitivos da linguagem)?

`content/first-order-logic/models-theories/expressing-props-of-structures.tex`

17.2 Expressando Propriedades de Estruturas

fol:mat:exs:
sec Frequentemente é útil e importante expressar condições sobre funções e relações, explanation ou, mais geralmente, que as funções e relações em uma estrutura satisfaçam essas condições. Por exemplo, nós gostaríamos de ter maneiras de distinguir aquelas **estruturas** para uma linguagem que “capturam” o que queremos que os **símbolos de predicado** “signifiquem” daquelas que não capturam. É claro que somos completamente livres para especificar quais **estruturas** “pretendemos”, por exemplo, podemos especificar que a interpretação do **símbolo de predicado** $\leq$ deve ser uma ordenação, ou que estamos interessados apenas em interpretações de $\mathcal{L}$ nas quais o domínio consiste em conjuntos e $\in$ é interpretado pela relação “é **um membro** de”. Mas podemos fazer isso com **sentenças** da linguagem? Em outras palavras, quais condições sobre **a estrutura** $\mathfrak{M}$ podemos expressar por **a sentença** (ou talvez um conjunto de **sentenças**) na linguagem de $\mathfrak{M}$? Há algumas condições que não conseguiremos expressar. Por exemplo, não há sentença de $\mathcal{L}_A$ que seja verdadeira em uma **estrutura** $\mathfrak{M}$ somente se

$|\mathfrak{M}| = \mathbb{N}$. Não podemos expressar “o domínio contém apenas números naturais”. Mas há “propriedades estruturais” de **estruturas** que talvez possamos expressar. Quais propriedades de **estruturas** podemos expressar por **sentenças**? Ou, dito de outra forma, quais coleções de **estruturas** podemos descrever como aquelas que tornam a **sentença** (ou conjunto de **sentenças**) verdadeira?

Definição 17.2 (Modelo de um conjunto). Seja Γ um conjunto de **sentenças** em uma linguagem $\mathcal{L}$. Nós dizemos que a **estrutura** $\mathfrak{M}$ é um **modelo** de Γ se $\mathfrak{M} \models \varphi$ para todo $\varphi \in \Gamma$.

Exemplo 17.3. A sentença $\forall x x \leq x$ é verdadeira em $\mathfrak{M}$ sse $\leq^{\mathfrak{M}}$ é uma relação reflexiva. A sentença $\forall x \forall y ((x \leq y \wedge y \leq x) \rightarrow x = y)$ é verdadeira em $\mathfrak{M}$ sse $\leq^{\mathfrak{M}}$ é antissimétrica. A sentença $\forall x \forall y \forall z ((x \leq y \wedge y \leq z) \rightarrow x \leq z)$ é verdadeira em $\mathfrak{M}$ sse $\leq^{\mathfrak{M}}$ é transitiva. Assim, os modelos de

$$\left\{ \begin{array}{l} \forall x x \leq x, \\ \forall x \forall y ((x \leq y \wedge y \leq x) \rightarrow x = y), \\ \forall x \forall y \forall z ((x \leq y \wedge y \leq z) \rightarrow x \leq z) \end{array} \right\}$$

são exatamente aquelas estruturas nas quais $\leq^{\mathfrak{M}}$ é reflexiva, antissimétrica e transitiva, isto é, uma ordem parcial. Portanto, podemos tomá-las como axiomas para a *teoria de primeira ordem das ordens parciais*.

content/first-order-logic/models-theories/theories.tex

17.3 Exemplos de Teorias de Primeira Ordem

Exemplo 17.4. A teoria das ordens lineares estritas na linguagem $\mathcal{L}_{<}$ é axiomatizada pelo conjunto

$$\left\{ \begin{array}{l} \forall x \neg x < x, \\ \forall x \forall y ((x < y \vee y < x) \vee x = y), \\ \forall x \forall y \forall z ((x < y \wedge y < z) \rightarrow x < z) \end{array} \right\}$$

Ela captura completamente as **estruturas** pretendidas: toda ordem linear estrita é um modelo deste sistema de axiomas e, vice-versa, se R é uma ordem linear sobre um conjunto X , então a estrutura $\mathfrak{M}$ com $|\mathfrak{M}| = X$ e $<^{\mathfrak{M}} = R$ é um modelo desta teoria.

Exemplo 17.5. A teoria dos grupos na linguagem $\mathbf{1}$ (**símbolo de constante**), $\cdot$ (**símbolo de função** de dois lugares) é axiomatizada por

$$\begin{array}{l} \forall x (x \cdot \mathbf{1}) = x \\ \forall x \forall y \forall z (x \cdot (y \cdot z)) = ((x \cdot y) \cdot z) \\ \forall x \exists y (x \cdot y) = \mathbf{1} \end{array}$$

fol:mat:the:
sec

17.3. EXEMPLOS DE TEORIAS DE PRIMEIRA ORDEM

Exemplo 17.6. A teoria da aritmética de Peano é axiomatizada pelas seguintes sentenças na linguagem da aritmética $\mathcal{L}_A$.

$$\begin{aligned} &\forall x \forall y (x' = y' \rightarrow x = y) \\ &\forall x \, 0 \neq x' \\ &\forall x (x + 0) = x \\ &\forall x \forall y (x + y') = (x + y)' \\ &\forall x (x \times 0) = 0 \\ &\forall x \forall y (x \times y') = ((x \times y) + x) \\ &\forall x \forall y (x < y \leftrightarrow \exists z (z' + x) = y) \end{aligned}$$

mais todas as sentenças da forma

$$(\varphi(0) \wedge \forall x (\varphi(x) \rightarrow \varphi(x'))) \rightarrow \forall x \varphi(x)$$

Como há infinitas sentenças desta última forma, este sistema de axiomas é infinito. Esta última forma é chamada de *esquema de indução*. (Na verdade, o esquema de indução é um pouco mais complicado do que deixamos transparecer aqui.)

O último axioma é uma *definição explícita* de $<$.

Exemplo 17.7. A teoria dos conjuntos puros desempenha um papel importante nos fundamentos (e na filosofia) da matemática. Um conjunto é puro se todos os seus **membros** são também conjuntos puros. O conjunto vazio conta, portanto, como puro, mas um conjunto que tem como **um membro** algo que não é um conjunto não seria puro. Então os conjuntos puros são aqueles que são formados apenas a partir do conjunto vazio e sem “urelementos”, isto é, objetos que não são, eles próprios, conjuntos.

O seguinte poderia ser considerado um sistema de axiomas para uma teoria dos conjuntos puros:

$$\begin{aligned} &\exists x \neg \exists y y \in x \\ &\forall x \forall y (\forall z (z \in x \leftrightarrow z \in y) \rightarrow x = y) \\ &\forall x \forall y \exists z \forall u (u \in z \leftrightarrow (u = x \vee u = y)) \\ &\forall x \exists y \forall z (z \in y \leftrightarrow \exists u (z \in u \wedge u \in x)) \end{aligned}$$

mais todas as sentenças da forma

$$\exists x \forall y (y \in x \leftrightarrow \varphi(y))$$

O primeiro axioma diz que há um conjunto sem **membros** (isto é, $\emptyset$ existe); o segundo diz que conjuntos são extensionais; o terceiro que, para quaisquer conjuntos X e Y , o conjunto $\{X, Y\}$ existe; o quarto que, para qualquer conjunto X , o conjunto $\cup X$ existe, onde $\cup X$ é a união de todos os elementos de X .

As **sentenças** mencionadas por último são chamadas coletivamente de *esquema de compreensão ingênuo*. Ele essencialmente diz que, para todo $\varphi(x)$, o conjunto $\{x : \varphi(x)\}$ existe—então, à primeira vista, um axioma verdadeiro, útil e talvez até necessário. É chamado de “ingênuo” porque, como se vê, ele torna esta teoria insatisfatível: se você toma $\varphi(y)$ como $\neg y \in y$, obtém a **sentença**

$$\exists x \forall y (y \in x \leftrightarrow \neg y \in y)$$

e esta **sentença** não é satisfeita em nenhuma **estrutura**.

Exemplo 17.8. Na área da *mereologia*, a relação de *parte* é uma relação fundamental. Assim como as teorias de conjuntos, há teorias de parte que axiomatizam várias concepções (às vezes conflitantes) dessa relação.

A linguagem da mereologia contém um único símbolo de predicado de dois lugares P , e $P(x, y)$ “significa” que x é uma parte de y . Quando temos essa interpretação em mente, a **estrutura** para essa linguagem é chamada de *estrutura de parte*. É claro que nem toda estrutura para um único predicado de dois lugares realmente merecerá esse nome. Para ter uma chance de capturar “parte”, P^M deve satisfazer algumas condições, que podemos estabelecer como axiomas para uma teoria de parte. Por exemplo, a parte é uma ordem parcial sobre objetos: todo objeto é uma parte (ainda que uma parte *imprópria*) de si mesmo; dois objetos diferentes não podem ser partes um do outro; uma parte de uma parte de um objeto é, ela própria, parte daquele objeto. Note que, nesse sentido, “é uma parte de” se assemelha a “é um subconjunto de”, mas não se assemelha a “é um elemento de”, que não é nem reflexiva nem transitiva.

$$\begin{aligned} &\forall x P(x, x) \\ &\forall x \forall y ((P(x, y) \wedge P(y, x)) \rightarrow x = y) \\ &\forall x \forall y \forall z ((P(x, y) \wedge P(y, z)) \rightarrow P(x, z)) \end{aligned}$$

Além disso, quaisquer dois objetos têm uma soma mereológica (um objeto que tem esses dois objetos como partes, e é minimal nesse aspecto).

$$\forall x \forall y \exists z \forall u (P(z, u) \leftrightarrow (P(x, u) \wedge P(y, u)))$$

Estes são apenas alguns dos princípios básicos de parte considerados pelos metafísicos. Princípios adicionais, no entanto, rapidamente se tornam difíceis de formular ou escrever sem primeiro introduzir algumas relações definidas. Por exemplo, a maioria dos metafísicos interessados em mereologia também considera o seguinte como um princípio válido: sempre que um objeto x tem uma parte própria y , ele também tem uma parte z que não tem partes em comum com y , e tal que a fusão de y e z é x .

17.4 Expressando Relações em a Estrutura

fol:mat:exr:
sec

Um uso principal que se pode dar às **fórmulas** é expressar propriedades e relações em a estrutura $\mathfrak{M}$ em termos dos primitivos da linguagem $\mathcal{L}$ de $\mathfrak{M}$. Com isso queremos dizer o seguinte: o **domínio** de $\mathfrak{M}$ é um conjunto de objetos. Os **símbolos de constante**, **símbolos de função** e **símbolos de predicado** são interpretados em $\mathfrak{M}$ por alguns objetos em $|\mathfrak{M}|$, funções sobre $|\mathfrak{M}|$, e relações sobre $|\mathfrak{M}|$. Por exemplo, se A_0^2 está em $\mathcal{L}$, então $\mathfrak{M}$ atribui a ele uma relação $R = A_0^2$. Então a fórmula $A_0^2(v_1, v_2)$ *expressa* exatamente essa relação, no seguinte sentido: se uma atribuição de variáveis s mapeia v_1 para $a \in |\mathfrak{M}|$ e v_2 para $b \in |\mathfrak{M}|$, então

$$Rab \text{ sse } \mathfrak{M}, s \models A_0^2(v_1, v_2).$$

Note que temos que envolver atribuições de variáveis aqui: não podemos simplesmente dizer “ $Rab \text{ sse } \mathfrak{M} \models A_0^2(a, b)$ ” porque a e b não são símbolos da nossa linguagem: eles são **membros** de $|\mathfrak{M}|$.

Como não temos apenas **fórmulas** atômicas, mas podemos combiná-las usando os conectivos lógicos e os quantificadores, **fórmulas** mais complexas podem definir outras relações que não estão diretamente embutidas em $\mathfrak{M}$. Estamos interessados em como fazer isso e, especificamente, quais relações podemos definir em a estrutura.

Definição 17.9. Seja $\varphi(v_1, \dots, v_n)$ a **fórmula** de $\mathcal{L}$ na qual apenas $v_1, \dots, v_n$ ocorrem livres, e seja $\mathfrak{M}$ a estrutura para $\mathcal{L}$. $\varphi(v_1, \dots, v_n)$ *expressa a relação* $R \subseteq |\mathfrak{M}|^n$ sse

$$Ra_1 \dots a_n \text{ sse } \mathfrak{M}, s \models \varphi(v_1, \dots, v_n)$$

para qualquer atribuição de variáveis s com $s(v_i) = a_i$ ($i = 1, \dots, n$).

Exemplo 17.10. No modelo padrão da aritmética $\mathfrak{N}$, a **fórmula** $v_1 < v_2 \vee v_1 = v_2$ expressa a relação $\leq$ sobre $\mathbb{N}$. A **fórmula** $v_2 = v_1'$ expressa a relação sucessora, isto é, a relação $R \subseteq \mathbb{N}^2$ onde Rnm vale se m é o sucessor de n . A fórmula $v_1 = v_2'$ expressa a relação predecessora. As **fórmulas** $\exists v_3 (v_3 \neq 0 \wedge v_2 = (v_1 + v_3))$ e $\exists v_3 (v_1 + v_3' = v_2)$ ambas expressam a relação $<$. Isso significa que o símbolo de predicado $<$ é, na verdade, supérfluo na linguagem da aritmética; ele pode ser definido.

Essa ideia não é interessante apenas em **estruturas** específicas, mas geralmente sempre que usamos uma linguagem para descrever um modelo ou modelos pretendidos, isto é, quando consideramos teorias. Essas teorias frequentemente contêm apenas alguns **símbolos de predicado** como símbolos básicos, mas no domínio que elas são usadas para descrever muitas outras relações frequentemente desempenham um papel importante. Se essas outras relações podem ser sistematicamente expressas pelas relações que interpretam os **símbolos de predicado** básicos da linguagem, nós dizemos que podemos *defini-las* na linguagem.

explanation

explanation

Problema 17.1. Encontre fórmulas em $\mathcal{L}_A$ que definam as seguintes relações:

1. n está entre i e j ;
2. n divide m exatamente (isto é, m é um múltiplo de n);
3. n é um número primo (isto é, nenhum número além de 1 e n divide n exatamente).

Problema 17.2. Suponha que a fórmula $\varphi(v_1, v_2)$ expressa a relação $R \subseteq |\mathfrak{M}|^2$ em a estrutura $\mathfrak{M}$. Encontre fórmulas que expressem as seguintes relações:

1. a inversa R^{-1} de R ;
2. o produto relativo $R \mid R$;

Você consegue encontrar uma maneira de expressar R^+ , o fecho transitivo de R ?

Problema 17.3. Seja $\mathcal{L}$ a linguagem contendo apenas um símbolo de predicado de 2 lugares $<$ (sem outros símbolos de constante, símbolos de função ou símbolos de predicado— exceto, é claro, $=$). Seja $\mathfrak{N}$ a estrutura tal que $|\mathfrak{N}| = \mathbb{N}$, e $<^{\mathfrak{N}} = \{\langle n, m \rangle : n < m\}$. Prove o seguinte:

1. $\{0\}$ é definível em $\mathfrak{N}$;
2. $\{1\}$ é definível em $\mathfrak{N}$;
3. $\{2\}$ é definível em $\mathfrak{N}$;
4. para cada $n \in \mathbb{N}$, o conjunto $\{n\}$ é definível em $\mathfrak{N}$;
5. todo subconjunto finito de $|\mathfrak{N}|$ é definível em $\mathfrak{N}$;
6. todo subconjunto cofinito de $|\mathfrak{N}|$ é definível em $\mathfrak{N}$ (onde $X \subseteq \mathbb{N}$ é cofinito sse $\mathbb{N} \setminus X$ é finito).

content/first-order-logic/models-theories/set-theory.tex

17.5 A Teoria dos Conjuntos

Quase toda a matemática pode ser desenvolvida na teoria dos conjuntos. Desenvolver a matemática nessa teoria envolve uma série de coisas. Primeiro, requer um conjunto de axiomas para a relação $\in$. Vários sistemas de axiomas diferentes foram desenvolvidos, às vezes com propriedades conflitantes de $\in$. O sistema de axiomas conhecido como **ZFC**, a teoria dos conjuntos de Zermelo–Fraenkel com o axioma da escolha, destaca-se: é de longe o mais amplamente usado e estudado, porque acontece que seus axiomas bastam para provar quase todas as coisas que os matemáticos esperam ser capazes de provar. Mas antes que isso possa ser estabelecido, é primeiro necessário deixar claro como

fol:mat:set:
sec

17.5. A TEORIA DOS CONJUNTOS

podemos sequer *expressar* todas as coisas que os matemáticos gostariam de expressar. Para começar, a linguagem não contém **símbolos de constante** nem **símbolos de função**, então parece, à primeira vista, pouco claro que possamos falar sobre conjuntos particulares (como $\emptyset$ ou $\mathbb{N}$), falar sobre operações sobre conjuntos (como $X \cup Y$ e $\wp(X)$), muito menos outras construções que envolvem coisas diferentes de conjuntos, como relações e funções.

Para começar, “é um elemento de” não é a única relação na qual estamos interessados: “é um subconjunto de” parece quase tão importante. Mas nós podemos *definir* “é um subconjunto de” em termos de “é um elemento de”. Para fazer isso, temos que encontrar a **fórmula** $\varphi(x, y)$ na linguagem da teoria dos conjuntos que é satisfeita por um par de conjuntos $\langle X, Y \rangle$ sse $X \subseteq Y$. Mas X é um subconjunto de Y exatamente quando todos os **membros** de X são também **membros** de Y . Então podemos definir $\subseteq$ pela fórmula

$$\forall z (z \in x \rightarrow z \in y)$$

Agora, sempre que quisermos usar a relação $\subseteq$ em uma fórmula, nós poderíamos, em vez disso, usar aquela fórmula (com x e y adequadamente substituídos, e a variável ligada z renomeada se necessário). Por exemplo, a extensionalidade dos conjuntos significa que se quaisquer conjuntos x e y estão contidos um no outro, então x e y devem ser o mesmo conjunto. Isso pode ser expresso por $\forall x \forall y ((x \subseteq y \wedge y \subseteq x) \rightarrow x = y)$, ou, se substituirmos $\subseteq$ pela definição acima, por

$$\forall x \forall y ((\forall z (z \in x \rightarrow z \in y) \wedge \forall z (z \in y \rightarrow z \in x)) \rightarrow x = y).$$

Este é, de fato, um dos axiomas de **ZFC**, o “axioma da extensionalidade”.

Não há **símbolo de constante** para $\emptyset$, mas podemos expressar “ x é vazio” por $\neg \exists y y \in x$. Então “ $\emptyset$ existe” torna-se a **sentença** $\exists x \neg \exists y y \in x$. Este é outro axioma de **ZFC**. (Note que o axioma da extensionalidade implica que há apenas um conjunto vazio.) Sempre que quisermos falar sobre $\emptyset$ na linguagem da teoria dos conjuntos, nós escreveríamos isso como “há um conjunto que é vazio e ...” Como exemplo, para expressar o fato de que $\emptyset$ é um subconjunto de todo conjunto, poderíamos escrever

$$\exists x (\neg \exists y y \in x \wedge \forall z x \subseteq z)$$

onde, é claro, $x \subseteq z$ teria, por sua vez, que ser substituído por sua definição.

Para falar sobre operações sobre conjuntos, como $X \cup Y$ e $\wp(X)$, nós temos que usar um truque semelhante. Não há símbolos de função na linguagem da teoria dos conjuntos, mas podemos expressar as relações funcionais $X \cup Y = Z$ e $\wp(X) = Y$ por

$$\begin{aligned} \forall u ((u \in x \vee u \in y) &\leftrightarrow u \in z) \\ \forall u (u \subseteq x &\leftrightarrow u \in y) \end{aligned}$$

já que os **membros** de $X \cup Y$ são exatamente os conjuntos que são ou **membros** de X ou **membros** de Y , e os **membros** de $\wp(X)$ são exatamente os subconjuntos

de X . No entanto, isso não nos permite usar $x \cup y$ ou $\wp(x)$ como se fossem termos: só podemos usar as **fórmulas** inteiras que definem as relações $X \cup Y = Z$ e $\wp(X) = Y$. De fato, não sabemos que essas relações são alguma vez satisfeitas, isto é, não sabemos que uniões e conjuntos potência sempre existem. Por exemplo, a **sentença** $\forall x \exists y \wp(x) = y$ é outro axioma de **ZFC** (o axioma do conjunto potência).

E quanto a falar de pares ordenados ou funções? Aqui temos que explicar como podemos pensar em pares ordenados e funções como tipos especiais de conjuntos. Uma maneira de definir o par ordenado $\langle x, y \rangle$ é como o conjunto $\{\{x\}, \{x, y\}\}$. Mas, como antes, não podemos introduzir a **símbolo de função** que nomeie esse conjunto; só podemos definir a relação $\langle x, y \rangle = z$, isto é, $\{\{x\}, \{x, y\}\} = z$:

$$\forall u (u \in z \leftrightarrow (\forall v (v \in u \leftrightarrow v = x) \vee \forall v (v \in u \leftrightarrow (v = x \vee v = y))))$$

Isso diz que os **membros** u de z são exatamente aqueles conjuntos que ou têm x como seu único **membro** ou têm x e y como seus únicos **membros** (em outras palavras, aqueles conjuntos que são ou idênticos a $\{x\}$ ou idênticos a $\{x, y\}$). Uma vez que temos isso, podemos dizer outras coisas, por exemplo, que $X \times Y = Z$:

$$\forall z (z \in Z \leftrightarrow \exists x \exists y (x \in X \wedge y \in Y \wedge \langle x, y \rangle = z))$$

Uma função $f: X \rightarrow Y$ pode ser pensada como a relação $f(x) = y$, isto é, como o conjunto de pares $\{\langle x, y \rangle : f(x) = y\}$. Nós podemos então dizer que um conjunto f é uma função de X em Y se (a) é uma relação $\subseteq X \times Y$, (b) é total, isto é, para todo $x \in X$ há algum $y \in Y$ tal que $\langle x, y \rangle \in f$ e (c) é funcional, isto é, sempre que $\langle x, y \rangle, \langle x, y' \rangle \in f$, $y = y'$ (porque os valores de funções devem ser únicos). Então “ f é uma função de X em Y ” pode ser escrito como:

$$\begin{aligned} & \forall u (u \in f \rightarrow \exists x \exists y (x \in X \wedge y \in Y \wedge \langle x, y \rangle = u)) \wedge \\ & \forall x (x \in X \rightarrow (\exists y (y \in Y \wedge \text{mapeia}(f, x, y)) \wedge \\ & \quad (\forall y \forall y' ((\text{mapeia}(f, x, y) \wedge \text{mapeia}(f, x, y')) \rightarrow y = y')))) \end{aligned}$$

onde $\text{mapeia}(f, x, y)$ abrevia $\exists v (v \in f \wedge \langle x, y \rangle = v)$ (esta **fórmula** expressa “ $f(x) = y$ ”).

Agora também não é difícil expressar que $f: X \rightarrow Y$ é **injetiva**, por exemplo:

$$\begin{aligned} f: X \rightarrow Y \wedge \forall x \forall x' ((x \in X \wedge x' \in X \wedge \\ \exists y (\text{mapeia}(f, x, y) \wedge \text{mapeia}(f, x', y))) \rightarrow x = x') \end{aligned}$$

Uma função $f: X \rightarrow Y$ é **injetiva** sse, sempre que f mapeia $x, x' \in X$ para um único y , $x = x'$. Se abreviarmos esta fórmula como $\text{inj}(f, X, Y)$, já estamos em posição de afirmar na linguagem da teoria dos conjuntos algo tão não trivial quanto o teorema de Cantor: não há função **injetiva** de $\wp(X)$ em X :

$$\forall X \forall Y (\wp(X) = Y \rightarrow \neg \exists f \text{inj}(f, Y, X))$$

17.6. EXPRESSANDO O TAMANHO DE ESTRUTURAS

Poderíamos pensar que a teoria dos conjuntos requer outro axioma que garanta a existência de um conjunto para toda propriedade definidora. Se $\varphi(x)$ é uma fórmula da teoria dos conjuntos com a variável x livre, podemos considerar a **sentença**

$$\exists y \forall x (x \in y \leftrightarrow \varphi(x)).$$

Esta **sentença** afirma que há um conjunto y cujos **membros** são todos e somente aqueles x que satisfazem $\varphi(x)$. Este esquema é chamado de “princípio da compreensão”. Parece muito útil; infelizmente é inconsistente. Tome $\varphi(x) \equiv \neg x \in x$, então o princípio da compreensão afirma

$$\exists y \forall x (x \in y \leftrightarrow x \notin x),$$

isto é, afirma a existência de um conjunto de todos os conjuntos que não são **membros** de si mesmos. Nenhum tal conjunto pode existir—este é o Paradoxo de Russell. **ZFC**, de fato, contém uma versão restrita—e consistente—deste princípio, o princípio da separação:

$$\forall z \exists y \forall x (x \in y \leftrightarrow (x \in z \wedge \varphi(x))).$$

Problema 17.4. Mostre que o princípio da compreensão é inconsistente dando a **derivação** que mostre

$$\exists y \forall x (x \in y \leftrightarrow x \notin x) \vdash \perp.$$

Pode ajudar mostrar primeiro $(A \rightarrow \neg A) \wedge (\neg A \rightarrow A) \vdash \perp$.

content/first-order-logic/models-theories/size-of-structures.tex

17.6 Expressando o Tamanho de Estruturas

fol:mat:siz:
sec Há algumas propriedades de estruturas que podemos expressar mesmo sem explanation usar os símbolos não lógicos de uma linguagem. Por exemplo, há **sentenças** que são verdadeiras em a **estrutura** se, e somente se, o **domínio** da **estrutura** tem pelo menos, no máximo ou exatamente um certo número n de **membros**.

Proposição 17.11. A **sentença**

$$\begin{aligned} \varphi_{\geq n} \equiv & \exists x_1 \exists x_2 \dots \exists x_n \\ & (x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_1 \neq x_4 \wedge \dots \wedge x_1 \neq x_n \wedge \\ & x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge \dots \wedge x_2 \neq x_n \wedge \\ & \vdots \\ & x_{n-1} \neq x_n) \end{aligned}$$

é verdadeira em a **estrutura** $\mathfrak{M}$ se, e somente se, $|\mathfrak{M}|$ contém pelo menos n **membros**. Consequentemente, $\mathfrak{M} \models \neg \varphi_{\geq n+1}$ se, e somente se, $|\mathfrak{M}|$ contém no máximo n **membros**.

Proposição 17.12. A *sentença*

$$\begin{aligned} \varphi_{=n} \equiv \exists x_1 \exists x_2 \dots \exists x_n \\ (x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_1 \neq x_4 \wedge \dots \wedge x_1 \neq x_n \wedge \\ x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge \dots \wedge x_2 \neq x_n \wedge \\ \vdots \\ x_{n-1} \neq x_n \wedge \\ \forall y (y = x_1 \vee \dots \vee y = x_n)) \end{aligned}$$

é verdadeira em a *estrutura* $\mathfrak{M}$ se, e somente se, $|\mathfrak{M}|$ contém exatamente n *membros*.

Proposição 17.13. Uma *estrutura* é infinita se, e somente se, é um modelo de

$$\{\varphi_{\geq 1}, \varphi_{\geq 2}, \varphi_{\geq 3}, \dots\}.$$

Não há uma única sentença puramente lógica que seja verdadeira em $\mathfrak{M}$ se, e somente se, $|\mathfrak{M}|$ é infinito. No entanto, pode-se dar *sentenças* com *símbolos de predicado* não lógicos que só têm modelos infinitos (embora nem toda *estrutura* infinita seja modelo delas). A propriedade de ser uma estrutura finita e a propriedade de ser uma estrutura *não enumerável* nem sequer podem ser expressas com um conjunto infinito de *sentenças*. Estes fatos seguem dos teoremas da compacidade e de Löwenheim–Skolem.

Capítulo 18

Sistemas de Derivação

Este capítulo reúne material geral sobre sistemas de *derivação*. Um livro-texto que usa um sistema específico pode inserir a seção de introdução mais a seção de panorama relevante no início do capítulo que introduz esse sistema.

18.1. INTRODUÇÃO

18.1 Introdução

fol:prf:int:
sec

As lógicas comumente têm tanto uma semântica quanto um sistema de **derivação**. A semântica trata de conceitos como verdade, satisfatibilidade, validade e consequência. O propósito dos sistemas de **derivação** é fornecer um método puramente sintático de estabelecer consequência e validade. Eles são puramente sintáticos no sentido de que uma **derivação** em tal sistema é um objeto sintático finito, geralmente uma sequência (ou outra disposição finita) de **sentenças** ou **fórmulas**. Bons sistemas de **derivação** têm a propriedade de que qualquer sequência ou disposição dada de **sentenças** ou **fórmulas** pode ser verificada mecanicamente como “correta”.

Os sistemas de **derivação** mais simples (e historicamente os primeiros) para a lógica de primeira ordem foram *axiomáticos*. Uma sequência de **fórmulas** conta como uma **derivação** em tal sistema se cada **fórmula** individual nela é ou está entre um conjunto fixo de “axiomas” ou segue de **fórmulas** anteriores na sequência por uma entre um número fixo de “regras de inferência”—e pode ser verificado mecanicamente se uma **fórmula** é um axioma e se segue corretamente de outras **fórmulas** por uma das regras de inferência. Sistemas de **derivação** axiomáticos são fáceis de descrever—e também fáceis de tratar meta-teoricamente—mas as **derivações** neles são difíceis de ler e entender, e também difíceis de produzir.

Outros sistemas de **derivação** foram desenvolvidos com o objetivo de facilitar a construção de **derivações** ou facilitar a compreensão de **derivações** uma vez completas. Exemplos são a dedução natural, árvores de verdade, também conhecidas como provas por tableaux, e o cálculo de sequentes. Alguns sistemas de **derivação** são projetados especialmente com a mecanização em mente; por exemplo, o método da resolução é fácil de implementar em software (mas suas **derivações** são essencialmente impossíveis de entender). A maioria desses outros sistemas de **derivação** representa **derivações** como árvores de **fórmulas** em vez de sequências. Isso facilita ver quais partes de uma **derivação** dependem de quais outras partes.

Assim, para uma dada lógica, como a lógica de primeira ordem, os diferentes sistemas de **derivação** darão diferentes explicações do que significa uma **sentença** ser um *teorema* e o que significa uma **sentença** ser **derivável** a partir de outras. Seja como for feito isso (via **derivações** axiomáticas, deduções naturais, **derivações** de sequentes, árvores de verdade, refutações por resolução), queremos que essas relações correspondam às noções semânticas de validade e consequência. Escrevamos $\vdash \varphi$ para “ φ é um teorema” e “ $\Gamma \vdash \varphi$ ” para “ φ é **derivável** de Γ .” Seja como for $\vdash$ definido, queremos que corresponda a $\models$, isto é:

1. $\vdash \varphi$ se, e somente se, $\models \varphi$
2. $\Gamma \vdash \varphi$ se, e somente se, $\Gamma \models \varphi$

A direção “somente se” acima é chamada de *correção*. Um sistema de **derivação** é correto se a **derivabilidade** garante consequência (ou validade). Todo sistema

de *derivação* decente tem que ser correto; sistemas de *derivação* incorretos não são úteis de modo algum. Afinal, o propósito inteiro de uma *derivação* é fornecer uma garantia sintática de validade ou consequência. Provaremos a correção para os sistemas de *derivação* que apresentamos.

A direção convers “se” também é importante: é chamada de *completude*. Um sistema de *derivação* completo é forte o suficiente para mostrar que φ é um teorema sempre que φ é válida, e que $\Gamma \vdash \varphi$ sempre que $\Gamma \models \varphi$. A completude é mais difícil de estabelecer, e algumas lógicas não têm sistemas de *derivação* completos. A lógica de primeira ordem tem. Kurt Gödel foi o primeiro a provar a completude para um sistema de *derivação* da lógica de primeira ordem em sua dissertação de 1929.

Outro conceito ligado aos sistemas de *derivação* é o de *consistência*. Um conjunto de *sentenças* é chamado inconsistente se qualquer coisa pode ser *derivada* dele, e consistente caso contrário. A inconsistência é a contrapartida sintática da insatisfatibilidade: como conjuntos insatisfatíveis, conjuntos inconsistentes de *sentenças* não fazem boas teorias; são defeituosos de maneira fundamental. Conjuntos consistentes de *sentenças* podem não ser verdadeiros ou úteis, mas pelo menos passam esse limiar mínimo de utilidade lógica. Para diferentes sistemas de *derivação*, a definição específica de consistência de conjuntos de *sentenças* pode diferir, mas como $\vdash$, queremos que a consistência coincida com sua contrapartida semântica, a satisfatibilidade. Queremos que seja sempre o caso que Γ é consistente se, e somente se, é satisfatível. Aqui, a direção “somente se” corresponde à completude (consistência garante satisfatibilidade), e a direção “se” corresponde à correção (satisfatibilidade garante consistência). De fato, para a lógica clássica de primeira ordem, as duas versões de correção e completude são equivalentes.

[content/first-order-logic/proof-systems/sequent-calculus.tex](#)

18.2 O Cálculo de Sequentes

Enquanto muitos sistemas de *derivação* operam com disposições de *sentenças*, o cálculo de sequentes opera com *sequentes*. Um sequente é uma expressão da forma

$$\varphi_1, \dots, \varphi_m \Rightarrow \psi_1, \dots, \psi_n,$$

isto é, um par de sequências de *sentenças*, separadas pelo símbolo de sequente $\Rightarrow$. Qualquer uma das sequências pode ser vazia. Uma *derivação* no cálculo de sequentes é uma árvore de sequentes, em que os sequentes mais altos são de uma forma especial (são chamados de “sequentes iniciais” ou “axiomas”) e todo outro sequente segue dos sequentes imediatamente acima dele por uma das regras de inferência. As regras de inferência ou manipulam as *sentenças* nos sequentes (adicionando, removendo ou rearranjando-as à esquerda ou à direita), ou introduzem uma *fórmula* complexa na conclusão da regra. Por exemplo, a regra $\wedge L$ permite a inferência de $\varphi, \Gamma \Rightarrow \Delta$ para $\varphi \wedge \psi, \Gamma \Rightarrow \Delta$, e a

fol:prf:seq:
sec

18.3. DEDUÇÃO NATURAL

$\rightarrow R$ permite a inferência de $\varphi, \Gamma \Rightarrow \Delta, \psi$ para $\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi$, para quaisquer Γ, Δ, φ e ψ . (Em particular, Γ e Δ podem ser vazios.)

A relação $\vdash$ baseada no cálculo de seqüentes é definida da seguinte forma: $\Gamma \vdash \varphi$ se, e somente se, há alguma seqüência Γ_0 tal que toda φ em Γ_0 está em Γ e há uma **derivação** com o seqüente $\Gamma_0 \Rightarrow \varphi$ em sua raiz. φ é um teorema no cálculo de seqüentes se o seqüente $\Rightarrow \varphi$ tem uma **derivação**. Por exemplo, aqui está uma **derivação** que mostra que $\vdash (\varphi \wedge \psi) \rightarrow \varphi$:

$$\frac{\frac{\varphi \Rightarrow \varphi}{\varphi \wedge \psi \Rightarrow \varphi} \wedge L}{\Rightarrow (\varphi \wedge \psi) \rightarrow \varphi} \rightarrow R$$

Um conjunto Γ é inconsistente no cálculo de seqüentes se há uma **derivação** de $\Gamma_0 \Rightarrow$ (em que toda $\varphi \in \Gamma_0$ está em Γ e o lado direito do seqüente é vazio). Usando a regra WR, qualquer **sentença** pode ser **derivada** a partir de um conjunto inconsistente.

O cálculo de seqüentes foi inventado na década de 1930 por Gerhard Gentzen. Por causa de seu desenho sistemático e simétrico, é um formalismo muito útil para desenvolver uma teoria de **derivções**. É relativamente fácil encontrar **derivções** no cálculo de seqüentes, mas essas **derivções** são frequentemente difíceis de ler e sua conexão com provas às vezes não é fácil de ver. Provou ser uma abordagem muito elegante a sistemas de **derivação**, no entanto, e muitas lógicas têm sistemas de cálculo de seqüentes.

[content/first-order-logic/proof-systems/natural-deduction.tex](#)

18.3 Dedução Natural

fol:prf:ntd:
sec

A dedução natural é um sistema de **derivação** destinado a espelhar o raciocínio real (especialmente o tipo de raciocínio regimentado empregado por matemáticos). O raciocínio real procede por vários padrões “naturais”. Por exemplo, a prova por casos nos permite estabelecer uma conclusão com base em uma premissa disjuntiva, estabelecendo que a conclusão segue de qualquer um dos disjuntos. A prova indireta nos permite estabelecer uma conclusão mostrando que sua negação leva a uma contradição. A prova condicional estabelece uma afirmação condicional “se ... então ...” mostrando que o conseqüente segue do antecedente. A dedução natural é uma formalização de algumas dessas inferências naturais. Cada um dos conectivos lógicos e quantificadores vem com duas regras, uma de introdução e uma de eliminação, e cada uma corresponde a um desses padrões de inferência natural. Por exemplo, $\rightarrow$ Intro corresponde à prova condicional, e $\vee$ Elim à prova por casos. Uma regra particularmente simples é $\wedge$ Elim, que permite a inferência de $\varphi \wedge \psi$ para φ (ou ψ).

Uma característica que distingue a dedução natural de outros sistemas de **derivação** é o uso de hipóteses. Uma **derivação** em dedução natural é uma árvore de **fórmulas**. Uma única **fórmula** fica na raiz da árvore de **fórmulas**, e as “folhas” da árvore são **fórmulas** das quais a conclusão é derivada. Na dedução

natural, algumas fórmulas folha desempenham um papel dentro da derivação, mas são “consumidas” quando a derivação atinge a conclusão. Isso corresponde à prática, no raciocínio real, de introduzir hipóteses que só permanecem em vigor por um curto período. Por exemplo, em uma prova por casos, assumimos a verdade de cada um dos disjuntos; na prova condicional, assumimos a verdade do antecedente; na prova indireta, assumimos a verdade da negação da conclusão. Essa maneira de introduzir hipóteses hipotéticas e depois descartá-las a serviço de estabelecer um passo intermediário é uma marca da dedução natural. As fórmulas nas folhas de uma derivação em dedução natural são chamadas de hipóteses, e algumas das regras de inferência podem descarregá-las. Por exemplo, se temos uma derivação de ψ a partir de algumas hipóteses que incluem φ , então a regra $\rightarrow$ Intro nos permite inferir $\varphi \rightarrow \psi$ e descartar qualquer hipótese da forma φ . (Para acompanhar quais hipóteses são descartadas em quais inferências, rotulamos a inferência e as hipóteses que ela descarta com um número.) As hipóteses que permanecem não descarregadas no final da derivação são, juntas, suficientes para a verdade da conclusão, de modo que uma derivação estabelece que suas hipóteses não descarregadas implicam sua conclusão.

A relação $\Gamma \vdash \varphi$ baseada na dedução natural vale se, e somente se, há uma derivação em que φ é a última sentença na árvore, e toda folha que está não descarregada está em Γ . φ é um teorema na dedução natural se, e somente se, há uma derivação em que φ é a última sentença e todas as hipóteses são descarregadas. Por exemplo, aqui está uma derivação que mostra que $\vdash (\varphi \wedge \psi) \rightarrow \varphi$:

$$1 \frac{\frac{[\varphi \wedge \psi]^1}{\varphi} \wedge \text{Elim}}{(\varphi \wedge \psi) \rightarrow \varphi} \rightarrow \text{Intro}$$

O rótulo 1 indica que a hipótese $\varphi \wedge \psi$ é descarregada na inferência $\rightarrow$ Intro.

Um conjunto Γ é inconsistente se, e somente se, $\Gamma \vdash \perp$ na dedução natural. A regra $\perp_I$ faz com que, a partir de um conjunto inconsistente, qualquer sentença possa ser derivada.

Sistemas de dedução natural foram desenvolvidos por Gerhard Gentzen e Stanisław Jaśkowski na década de 1930, e depois por Dag Prawitz e Frederic Fitch. Porque suas inferências espelham métodos naturais de prova, é favorecida por filósofos. As versões desenvolvidas por Fitch são frequentemente usadas em livros introdutórios de lógica. Na filosofia da lógica, as regras da dedução natural às vezes foram tomadas como dando os significados dos operadores lógicos (“semântica da Teoria da Prova”).

`content/first-order-logic/proof-systems/tableaux.tex`

18.4 Tableaux

fol:prf:tab:
sec

18.4. TABLEAUX

Enquanto muitos sistemas de **derivação** operam com disposições de **sentenças**, os **tableaux** operam com **fórmulas assinadas**. A **fórmula assinada** é um par consistindo em um sinal de valor de verdade ($\mathbb{T}$ ou $\mathbb{F}$) e a **sentença**

$$\mathbb{T}\varphi \text{ ou } \mathbb{F}\varphi.$$

A **tableau** consiste em **fórmulas assinadas** dispostas em uma árvore ramificada para baixo. Começa com várias **hipóteses** e continua com **fórmulas assinadas** que resultam de uma das **fórmulas assinadas** acima dela pela aplicação de uma das regras de inferência. Cada regra nos permite adicionar uma ou mais **fórmulas assinadas** ao final de um ramo, ou duas **fórmulas assinadas** lado a lado—neste caso um ramo se divide em dois, com as duas **fórmulas assinadas** adicionadas formando os finais dos dois ramos.

Uma regra aplicada a uma **fórmula assinada** complexa resulta na adição de **fórmulas assinadas** que são sub-**fórmulas** imediatas. Elas vêm em pares, uma regra para cada um dos dois sinais. Por exemplo, a regra $\wedge\mathbb{T}$ se aplica a $\mathbb{T}\varphi \wedge \psi$, e permite a adição tanto das duas **fórmulas assinadas** $\mathbb{T}\varphi$ quanto de $\mathbb{T}\psi$ ao final de qualquer ramo que contém $\mathbb{T}\varphi \wedge \psi$, e a regra $\varphi \wedge \psi\mathbb{F}$ permite que um ramo seja dividido adicionando $\mathbb{F}\varphi$ e $\mathbb{F}\psi$ lado a lado. A **tableau** é fechado se cada um de seus ramos contém um par correspondente de **fórmulas assinadas** $\mathbb{T}\varphi$ e $\mathbb{F}\varphi$.

A relação $\vdash$ baseada em **tableaux** é definida da seguinte forma: $\Gamma \vdash \varphi$ se, e somente se, há algum conjunto finito $\Gamma_0 = \{\psi_1, \dots, \psi_n\} \subseteq \Gamma$ tal que há um **tableau** fechado para as hipóteses

$$\{\mathbb{F}\varphi, \mathbb{T}\psi_1, \dots, \mathbb{T}\psi_n\}$$

Por exemplo, aqui está um **tableau** fechado que mostra que $\vdash (\varphi \wedge \psi) \rightarrow \varphi$:

1.	$\mathbb{F}(\varphi \wedge \psi) \rightarrow \varphi$	Assumption
2.	$\mathbb{T}\varphi \wedge \psi$	$\rightarrow\mathbb{F}1$
3.	$\mathbb{F}\varphi$	$\rightarrow\mathbb{F}1$
4.	$\mathbb{T}\varphi$	$\rightarrow\mathbb{T}2$
5.	$\mathbb{T}\psi$	$\rightarrow\mathbb{T}2$
	$\otimes$	

Um conjunto Γ é inconsistente no cálculo de **tableau** se há um **tableau** fechado para hipóteses

$$\{\mathbb{T}\psi_1, \dots, \mathbb{T}\psi_n\}$$

para algum $\psi_i \in \Gamma$.

Os **Tableaux** foram inventados na década de 1950 independentemente por Evert Beth e Jaakko Hintikka, e simplificados e popularizados por Raymond Smullyan. São muito fáceis de usar, pois construir um **tableau** é um procedimento muito sistemático. Por causa da natureza sistemática dos **tableaux**, eles também se prestam à implementação por computador. No entanto, um **tableau** é frequentemente difícil de ler e sua conexão com provas às vezes não

é fácil de ver. A abordagem também é bastante geral, e muitas lógicas diferentes têm sistemas de **tableau**. Os **Tableaux** também nos ajudam a encontrar **estruturas** que satisfazem **sentenças** (ou conjuntos de **sentenças**) dadas: se o conjunto é satisfatível, ele não terá um **tableau** fechado, isto é, qualquer **tableau** terá um ramo aberto. A **estrutura** satisfatória pode ser “lida” de um ramo aberto, desde que toda regra que seja possível aplicar tenha sido aplicada naquele ramo. Há também uma conexão muito estreita com o cálculo de seqüentes: essencialmente, um **tableau** fechado é uma **derivação** condensada no cálculo de seqüentes, escrita de cabeça para baixo.

`content/first-order-logic/proof-systems/axiomatic-deduction.tex`

18.5 Dedução Axiomática

As **derivações** axiomáticas são os sistemas de **derivação** lógicos mais antigos e mais simples. Suas **derivações** são simplesmente seqüências de **sentenças**. Uma seqüência de **sentenças** conta como uma **derivação** correta se toda **sentença** φ nela satisfaz uma das seguintes condições:

fol:prf:axd:
sec

1. φ é um axioma, ou
2. φ é um **membro** de um dado conjunto Γ de **sentenças**, ou
3. φ é justificada por uma regra de inferência.

Para ser um axioma, φ tem que ter a forma de um entre vários esquemas fixos de **sentença**. Há muitos conjuntos de esquemas de axiomas que fornecem um sistema de **derivação** satisfatório (correto e completo) para a lógica de primeira ordem. Alguns são organizados de acordo com os conectivos que governam, por exemplo, os esquemas

$$\varphi \rightarrow (\psi \rightarrow \varphi) \quad \psi \rightarrow (\psi \vee \chi) \quad (\psi \wedge \chi) \rightarrow \psi$$

são axiomas comuns que governam $\rightarrow$, $\vee$ e $\wedge$. Alguns sistemas de axiomas visam um número mínimo de axiomas. Dependendo dos conectivos tomados como primitivos, é até possível encontrar sistemas de axiomas que consistem em um único axioma.

Uma regra de inferência é uma afirmação condicional que dá uma condição suficiente para que uma **sentença** em uma **derivação** seja justificada. O modus ponens é uma regra muito comum desse tipo: diz que se φ e $\varphi \rightarrow \psi$ já estão justificados, então ψ está justificado. Isso significa que uma linha em uma **derivação** que contém a **sentença** ψ está justificada, desde que tanto φ quanto $\varphi \rightarrow \psi$ (para alguma **sentença** φ) apareçam na **derivação** antes de ψ .

A relação $\vdash$ baseada em **derivações** axiomáticas é definida da seguinte forma: $\Gamma \vdash \varphi$ se, e somente se, há uma **derivação** com a **sentença** φ como sua última fórmula (e Γ é tomado como o conjunto de **sentenças** nessa **derivação** que são justificadas por (2) acima). φ é um teorema se φ tem uma **derivação** em que Γ

é vazio, isto é, toda **sentença** na **derivação** é justificada por (1) ou (3). Por exemplo, aqui está uma **derivação** que mostra que $\vdash \varphi \rightarrow (\psi \rightarrow (\psi \vee \varphi))$:

1. $\psi \rightarrow (\psi \vee \varphi)$
2. $(\psi \rightarrow (\psi \vee \varphi)) \rightarrow (\varphi \rightarrow (\psi \rightarrow (\psi \vee \varphi)))$
3. $\varphi \rightarrow (\psi \rightarrow (\psi \vee \varphi))$

A **sentença** na linha 1 tem a forma do axioma $\varphi \rightarrow (\varphi \vee \psi)$ (com os papéis de φ e ψ invertidos). A sentença na linha 2 tem a forma do axioma $\varphi \rightarrow (\psi \rightarrow \varphi)$. Assim, ambas as linhas estão justificadas. A linha 3 é justificada por *modus ponens*: se a abreviarmos como θ , então a linha 2 tem a forma $\chi \rightarrow \theta$, onde χ é $\psi \rightarrow (\psi \vee \varphi)$, isto é, a linha 1.

Um conjunto Γ é inconsistente se $\Gamma \vdash \perp$. Um sistema de axiomas completo também provará que $\perp \rightarrow \varphi$ para qualquer φ , e assim, se Γ é inconsistente, então $\Gamma \vdash \varphi$ para qualquer φ .

Sistemas de **derivações** axiomáticas para lógica foram dados pela primeira vez por Gottlob Frege em sua *Begriffsschrift* de 1879, que por essa razão é frequentemente considerada a primeira obra da lógica moderna. Foram aperfeiçoados em *Principia Mathematica* de Alfred North Whitehead e Bertrand Russell e por David Hilbert e seus alunos na década de 1920. São assim frequentemente chamados de “sistemas de Frege” ou “sistemas de Hilbert”. São muito versáteis no sentido de que é frequentemente fácil encontrar um sistema axiomático para uma lógica. Porque as **derivações** têm uma estrutura muito simples e apenas uma ou duas regras de inferência, também é relativamente fácil provar coisas *sobre* eles. No entanto, são muito difíceis de usar na prática, isto é, é difícil encontrar e escrever provas.

Capítulo 19

O Cálculo de Sequentes

Este capítulo apresenta o cálculo de sequentes padrão LK de Gentzen para a lógica clássica de primeira ordem. Ele poderia ter mais exemplos e exercícios. Para incluir ou excluir material relevante ao cálculo de sequentes como sistema de prova, use a tag “prfLK”.

content/first-order-logic/sequent-calculus/rules-and-proofs.tex

19.1 Regras e Derivações

Para o que segue, sejam $\Gamma, \Delta, \Pi, \Lambda$ representando sequências finitas de **sentenças**. fol:seq:rul:
sec

Definição 19.1 (Sequente). Um *sequente* é uma expressão da forma

$$\Gamma \Rightarrow \Delta$$

em que Γ e Δ são sequências finitas (possivelmente vazias) de **sentenças** da linguagem $\mathcal{L}$. Γ é chamado de *antecedente*, enquanto Δ é o *sucedente*.

explanation A ideia intuitiva por trás de um sequente é: se todas as **sentenças** no antecedente valem, então ao menos uma das **sentenças** no sucedente vale. Isto é, se $\Gamma = \langle \varphi_1, \dots, \varphi_m \rangle$ e $\Delta = \langle \psi_1, \dots, \psi_n \rangle$, então $\Gamma \Rightarrow \Delta$ vale se, e somente se,

$$(\varphi_1 \wedge \dots \wedge \varphi_m) \rightarrow (\psi_1 \vee \dots \vee \psi_n)$$

vale. Há dois casos especiais: quando Γ é vazio e quando Δ é vazio. Quando Γ é vazio, isto é, $m = 0$, $\Rightarrow \Delta$ vale se, e somente se, $\psi_1 \vee \dots \vee \psi_n$ vale. Quando Δ é vazio, isto é, $n = 0$, $\Gamma \Rightarrow$ vale se, e somente se, $\neg(\varphi_1 \wedge \dots \wedge \varphi_m)$ vale. Dizemos que um sequente é válido se, e somente se, a **sentença** correspondente é válida.

Se Γ é uma sequência de **sentenças**, escrevemos Γ, φ para o resultado de acrescentar φ à extremidade direita de Γ (e φ, Γ para o resultado de acrescentar φ à extremidade esquerda de Γ). Se Δ também é uma sequência de **sentenças**, então Γ, Δ é a concatenação das duas sequências.

Definição 19.2 (Sequente Inicial). Um *sequente inicial* é um sequente de uma das seguintes formas:

1. $\varphi \Rightarrow \varphi$
2. $\Rightarrow \top$
3. $\perp \Rightarrow$

para qualquer **sentença** φ na linguagem.

Derivações no cálculo de sequentes são certas árvores de sequentes, em que os sequentes no topo são sequentes iniciais e, se um sequente está abaixo de um ou dois outros sequentes, ele deve seguir corretamente por uma regra de inferência. As regras para **LK** dividem-se em dois tipos principais: regras *lógicas* e regras *estruturais*. As regras lógicas recebem o nome do **operador principal** da **sentença** que contém φ e/ou ψ no sequente inferior. Cada uma vem em duas versões, uma para inferir um sequente com a **sentença** que contém o **operador lógico** à esquerda, e uma com a **sentença** à direita.

content/first-order-logic/sequent-calculus/propositional-rules.tex

19.2. REGRAS PROPOSICIONAIS

19.2 Regras Proposicionais

fol:seq:prl:
sec

Regras para $\neg$

$$\frac{\Gamma \Rightarrow \Delta, \varphi}{\neg \varphi, \Gamma \Rightarrow \Delta} \neg L \qquad \frac{\varphi, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg \varphi} \neg R$$

Regras para $\wedge$

$$\frac{\varphi, \Gamma \Rightarrow \Delta}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta} \wedge L \qquad \frac{\Gamma \Rightarrow \Delta, \varphi \quad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi} \wedge R$$

$$\frac{\psi, \Gamma \Rightarrow \Delta}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta} \wedge L$$

Regras para $\vee$

$$\frac{\varphi, \Gamma \Rightarrow \Delta \quad \psi, \Gamma \Rightarrow \Delta}{\varphi \vee \psi, \Gamma \Rightarrow \Delta} \vee L \qquad \frac{\Gamma \Rightarrow \Delta, \varphi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi} \vee R$$

$$\frac{\Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi} \vee R$$

Regras para $\rightarrow$

$$\frac{\Gamma \Rightarrow \Delta, \varphi \quad \psi, \Pi \Rightarrow \Lambda}{\varphi \rightarrow \psi, \Gamma, \Pi \Rightarrow \Delta, \Lambda} \rightarrow L \qquad \frac{\varphi, \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi} \rightarrow R$$

content/first-order-logic/sequent-calculus/quantifier-rules.tex

19.3 Regras de Quantificadores

fol:seq:qrl:
sec

Regras para $\forall$

$$\frac{\varphi(t), \Gamma \Rightarrow \Delta}{\forall x \varphi(x), \Gamma \Rightarrow \Delta} \forall L \qquad \frac{\Gamma \Rightarrow \Delta, \varphi(a)}{\Gamma \Rightarrow \Delta, \forall x \varphi(x)} \forall R$$

Em $\forall L$, t é um termo fechado (isto é, um termo sem variáveis). Em $\forall R$, a é **a símbolo de constante** que não pode ocorrer em lugar algum do sequente inferior da regra $\forall R$. Chamamos a de *variável própria* da inferência $\forall R$.¹

Regras para $\exists$

$$\boxed{\frac{\varphi(a), \Gamma \Rightarrow \Delta}{\exists x \varphi(x), \Gamma \Rightarrow \Delta} \exists L \qquad \frac{\Gamma \Rightarrow \Delta, \varphi(t)}{\Gamma \Rightarrow \Delta, \exists x \varphi(x)} \exists R}$$

Novamente, t é um termo fechado, e a é **a símbolo de constante** que não ocorre no sequente inferior da regra $\exists L$. Chamamos a de *variável própria* da inferência $\exists L$.

A condição de que uma variável própria não ocorra no sequente inferior da inferência $\forall R$ ou $\exists L$ é chamada de *condição de variável própria*.

explanation Lembre-se da convenção de que, quando φ é **a fórmula** com a **variável** x livre, indicamos isso escrevendo $\varphi(x)$. No mesmo contexto, $\varphi(t)$ é então uma abreviação de $\varphi[t/x]$. Assim, poderíamos também escrever a regra $\exists R$ como:

$$\frac{\Gamma \Rightarrow \Delta, \varphi[t/x]}{\Gamma \Rightarrow \Delta, \exists x \varphi} \exists R$$

Note que t pode já ocorrer em φ , por exemplo, φ poderia ser $P(t, x)$. Assim, inferir $\Gamma \Rightarrow \Delta, \exists x P(t, x)$ de $\Gamma \Rightarrow \Delta, P(t, t)$ é uma aplicação correta de $\exists R$ —pode-se “substituir” uma ou mais, e não necessariamente todas, as ocorrências de t na premissa pela **variável** x ligada. Contudo, as condições de variável própria em $\forall R$ e $\exists L$ exigem que **a símbolo de constante** a não ocorra em φ . Portanto, não se pode inferir corretamente $\Gamma \Rightarrow \Delta, \forall x P(a, x)$ de $\Gamma \Rightarrow \Delta, P(a, a)$ usando $\forall R$.

explanation Em $\exists R$ e $\forall L$ não há restrições sobre o termo t . Por outro lado, nas regras $\exists L$ e $\forall R$, a condição de variável própria exige que **a símbolo de constante** a não ocorra em lugar algum fora de $\varphi(a)$ no sequente superior. Isso é necessário para garantir que o sistema seja correto, isto é, que só **deriva** sequentes que sejam válidos. Sem essa condição, o seguinte seria permitido:

$$\frac{\frac{\varphi(a) \Rightarrow \varphi(a)}{\exists x \varphi(x) \Rightarrow \varphi(a)} * \exists L}{\exists x \varphi(x) \Rightarrow \forall x \varphi(x)} \forall R \qquad \frac{\frac{\varphi(a) \Rightarrow \varphi(a)}{\varphi(a) \Rightarrow \forall x \varphi(x)} * \forall R}{\exists x \varphi(x) \Rightarrow \forall x \varphi(x)} \exists L$$

Contudo, $\exists x \varphi(x) \Rightarrow \forall x \varphi(x)$ não é válido.

content/first-order-logic/sequent-calculus/structural-rules.tex

¹Usamos o termo “variável própria” embora a na regra acima seja **a símbolo de constante**. Isso tem razões históricas.

19.4 Regras Estruturais

fol:seq:srl:
sec

Precisamos também de algumas regras que nos permitam reorganizar **sentenças** nos lados esquerdo e direito de um sequente. Como as regras lógicas exigem que as **sentenças** na premissa sobre as quais a regra atua estejam ou na extremidade esquerda ou na extremidade direita, precisamos de uma regra de “troca” que nos permita mover **sentenças** para a posição correta. Às vezes também é importante poder combinar duas **sentenças** idênticas em uma só, e acrescentar **a sentença** em qualquer um dos lados.

Enfraquecimento

$$\frac{\Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta} \text{WL} \qquad \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \varphi} \text{WR}$$

Contração

$$\frac{\varphi, \varphi, \Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta} \text{CL} \qquad \frac{\Gamma \Rightarrow \Delta, \varphi, \varphi}{\Gamma \Rightarrow \Delta, \varphi} \text{CR}$$

Troca

$$\frac{\Gamma, \varphi, \psi, \Pi \Rightarrow \Delta}{\Gamma, \psi, \varphi, \Pi \Rightarrow \Delta} \text{XL} \qquad \frac{\Gamma \Rightarrow \Delta, \varphi, \psi, \Lambda}{\Gamma \Rightarrow \Delta, \psi, \varphi, \Lambda} \text{XR}$$

Uma série de inferências de enfraquecimento, contração e troca será frequentemente indicada por linhas de inferência duplas.

A regra a seguir, chamada de “corte”, não é estritamente necessária, mas torna muito mais fácil reutilizar e combinar **derivações**.

$$\frac{\Gamma \Rightarrow \Delta, \varphi \quad \varphi, \Pi \Rightarrow \Lambda}{\Gamma, \Pi \Rightarrow \Delta, \Lambda} \text{Cut}$$

19.5 Derivações

explanation Já dissemos como é um sequente inicial e demos as regras de inferência. fol:seq:der:sec **Derivações** no cálculo de sequentes são geradas indutivamente a partir disso: cada **derivação** ou é um sequente inicial por si só, ou consiste em uma ou duas **derivações** seguidas de uma inferência.

Definição 19.3 (LK-derivação). Uma **LK-derivação** de um sequente S é uma árvore finita de sequentes que satisfaz as seguintes condições:

1. Os sequentes no topo da árvore são sequentes iniciais.
2. O sequente na base da árvore é S .
3. Todo sequente na árvore exceto S é uma premissa de uma aplicação correta de uma regra de inferência cuja conclusão está diretamente abaixo daquele sequente na árvore.

Dizemos então que S é o *sequente final* da **derivação** e que S é **derivável em LK** (ou **LK-derivável**).

Exemplo 19.4. Todo sequente inicial, por exemplo, $\chi \Rightarrow \chi$, é a **derivação**. Podemos obter uma nova **derivação** a partir disso aplicando, digamos, a regra WL,

$$\frac{\Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta} \text{ WL}$$

A regra, contudo, deve ser geral: podemos substituir o φ na regra por qualquer **sentença**, por exemplo, também por θ . Se a premissa corresponde ao nosso sequente inicial $\chi \Rightarrow \chi$, isso significa que tanto Γ quanto Δ são apenas χ , e a conclusão seria então $\theta, \chi \Rightarrow \chi$. Assim, o seguinte é a **derivação**:

$$\frac{\chi \Rightarrow \chi}{\theta, \chi \Rightarrow \chi} \text{ WL}$$

Podemos agora aplicar outra regra, digamos XL, que nos permite trocar duas **sentenças** à esquerda. Assim, o seguinte também é uma **derivação** correta:

$$\frac{\frac{\chi \Rightarrow \chi}{\theta, \chi \Rightarrow \chi} \text{ WL}}{\chi, \theta \Rightarrow \chi} \text{ XL}$$

Nesta aplicação da regra, que foi dada como

$$\frac{\Gamma, \varphi, \psi, \Pi \Rightarrow \Delta}{\Gamma, \psi, \varphi, \Pi \Rightarrow \Delta} \text{ XL}$$

tanto Γ quanto Π eram vazios, Δ é χ , e os papéis de φ e ψ são desempenhados por θ e χ , respectivamente. De modo muito semelhante, vemos também que

19.6. EXEMPLOS DE DERIVAÇÕES

$$\frac{\theta \Rightarrow \theta}{\chi, \theta \Rightarrow \theta} \text{WL}$$

é a **derivação**. Agora podemos pegar essas duas **derivações** e combiná-las usando $\wedge R$. Aquela regra era

$$\frac{\Gamma \Rightarrow \Delta, \varphi \quad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi} \wedge R$$

No nosso caso, as premissas devem corresponder aos últimos sequentes das **derivações** que terminam nas premissas. Isso significa que Γ é χ, θ , Δ é vazio, φ é χ e ψ é θ . Assim, a conclusão, se a inferência for correta, é $\chi, \theta \Rightarrow \chi \wedge \theta$.

$$\frac{\frac{\frac{\chi \Rightarrow \chi}{\theta, \chi \Rightarrow \chi} \text{WL}}{\chi, \theta \Rightarrow \chi} \text{XL} \quad \frac{\theta \Rightarrow \theta}{\chi, \theta \Rightarrow \theta} \text{WL}}{\chi, \theta \Rightarrow \chi \wedge \theta} \wedge R$$

É claro que também podemos inverter as premissas; então φ seria θ e ψ seria χ .

$$\frac{\frac{\theta \Rightarrow \theta}{\chi, \theta \Rightarrow \theta} \text{WL} \quad \frac{\frac{\chi \Rightarrow \chi}{\theta, \chi \Rightarrow \chi} \text{WL}}{\chi, \theta \Rightarrow \chi} \text{XL}}{\chi, \theta \Rightarrow \theta \wedge \chi} \wedge R$$

content/first-order-logic/sequent-calculus/proving-things.tex

19.6 Exemplos de Derivações

fol:seq:pro:
sec

Exemplo 19.5. Dê uma **LK-derivação** para o sequente $\varphi \wedge \psi \Rightarrow \varphi$.

Começamos escrevendo o sequente final desejado na base da **derivação**.

$$\overline{\varphi \wedge \psi \Rightarrow \varphi}$$

Em seguida, precisamos descobrir que tipo de inferência poderia ter um sequente inferior dessa forma. Poderia ser uma regra estrutural, mas é uma boa ideia começar procurando uma regra lógica. O único conectivo lógico que ocorre no sequente inferior é $\wedge$, então procuramos uma regra de $\wedge$ e, como o símbolo $\wedge$ ocorre no antecedente, estamos diante da regra $\wedge L$.

$$\overline{\varphi \wedge \psi \Rightarrow \varphi} \wedge L$$

Há duas opções para o que poderia ter sido o sequente superior da inferência $\wedge L$: poderíamos ter um sequente superior $\varphi \Rightarrow \varphi$ ou $\psi \Rightarrow \varphi$. Claramente, $\varphi \Rightarrow \varphi$ é um sequente inicial (o que é bom), enquanto $\psi \Rightarrow \varphi$ não é derivável em geral. Preenchemos o sequente superior:

CAPÍTULO 19. O CÁLCULO DE SEQUENTES

$$\frac{\varphi \Rightarrow \varphi}{\varphi \wedge \psi \Rightarrow \varphi} \wedge L$$

Agora temos uma **LK-derivação** correta do sequente $\varphi \wedge \psi \Rightarrow \varphi$.

Exemplo 19.6. Dê uma **LK-derivação** para o sequente $\neg\varphi \vee \psi \Rightarrow \varphi \rightarrow \psi$.

Comece escrevendo o sequente final desejado na base da **derivação**.

$$\overline{\neg\varphi \vee \psi \Rightarrow \varphi \rightarrow \psi}$$

Para encontrar uma regra lógica que pudesse nos dar esse sequente final, olhamos para os conectivos lógicos no sequente final: $\neg$, $\vee$ e $\rightarrow$. No momento, só nos importam $\vee$ e $\rightarrow$, porque são **operadores principais** de **sentenças** no sequente final, enquanto $\neg$ está dentro do escopo de outro conectivo, de modo que cuidaremos dele depois. Nossas opções de regras lógicas para a inferência final são, portanto, a regra $\vee L$ e a regra $\rightarrow R$. Poderíamos escolher qualquer uma das regras, na verdade, mas vamos escolher a regra $\rightarrow R$ (quando não fosse por outro motivo, porque ela nos permite adiar a divisão em dois ramos). De acordo com a forma das inferências $\rightarrow R$ que podem produzir o sequente inferior, isto deve ser assim:

$$\frac{\overline{\varphi, \neg\varphi \vee \psi \Rightarrow \psi}}{\neg\varphi \vee \psi \Rightarrow \varphi \rightarrow \psi} \rightarrow R$$

Se movermos $\neg\varphi \vee \psi$ para a parte externa do antecedente, podemos aplicar a regra $\vee L$. De acordo com o esquema, isto deve se dividir em dois sequentes superiores como segue:

$$\frac{\frac{\overline{\neg\varphi, \varphi \Rightarrow \psi} \quad \overline{\psi, \varphi \Rightarrow \psi}}{\neg\varphi \vee \psi, \varphi \Rightarrow \psi} \vee L \quad \frac{\overline{\varphi, \neg\varphi \vee \psi \Rightarrow \psi}}{\neg\varphi \vee \psi \Rightarrow \varphi \rightarrow \psi} \rightarrow R$$

Lembre-se de que estamos tentando subir até sequentes iniciais; parece que estamos bem perto! O ramo direito está a apenas um enfraquecimento e uma troca de um sequente inicial, e então estará pronto:

$$\frac{\overline{\neg\varphi, \varphi \Rightarrow \psi} \quad \frac{\frac{\overline{\psi \Rightarrow \psi}}{\varphi, \psi \Rightarrow \psi} WL}{\psi, \varphi \Rightarrow \psi} XL}{\neg\varphi \vee \psi, \varphi \Rightarrow \psi} \vee L \quad \frac{\overline{\varphi, \neg\varphi \vee \psi \Rightarrow \psi}}{\neg\varphi \vee \psi \Rightarrow \varphi \rightarrow \psi} \rightarrow R$$

Agora, olhando para o ramo esquerdo, o único conectivo lógico em qualquer **sentença** é o símbolo $\neg$ nas **sentenças** do antecedente, então estamos diante de uma instância da regra $\neg L$.

19.6. EXEMPLOS DE DERIVAÇÕES

$$\begin{array}{c}
 \frac{}{\varphi \Rightarrow \psi, \varphi} \neg L \quad \frac{\psi \Rightarrow \psi}{\varphi, \psi \Rightarrow \psi} WL \\
 \frac{}{\neg \varphi, \varphi \Rightarrow \psi} \neg L \quad \frac{\psi \Rightarrow \psi}{\psi, \varphi \Rightarrow \psi} XL \\
 \frac{}{\neg \varphi \vee \psi, \varphi \Rightarrow \psi} \neg L \quad \frac{}{\varphi, \neg \varphi \vee \psi \Rightarrow \psi} XR \\
 \frac{}{\neg \varphi \vee \psi \Rightarrow \varphi \rightarrow \psi} \rightarrow R
 \end{array}$$

De modo semelhante a como finalizamos o ramo direito, estamos a apenas um enfraquecimento e uma troca de finalizar também este ramo esquerdo.

$$\begin{array}{c}
 \frac{\varphi \Rightarrow \varphi}{\varphi \Rightarrow \varphi, \psi} WR \\
 \frac{\varphi \Rightarrow \varphi, \psi}{\varphi \Rightarrow \psi, \varphi} XR \\
 \frac{}{\neg \varphi, \varphi \Rightarrow \psi} \neg L \quad \frac{\psi \Rightarrow \psi}{\varphi, \psi \Rightarrow \psi} WL \\
 \frac{}{\neg \varphi, \varphi \Rightarrow \psi} \neg L \quad \frac{\psi \Rightarrow \psi}{\psi, \varphi \Rightarrow \psi} XL \\
 \frac{}{\neg \varphi \vee \psi, \varphi \Rightarrow \psi} \neg L \quad \frac{}{\varphi, \neg \varphi \vee \psi \Rightarrow \psi} XR \\
 \frac{}{\neg \varphi \vee \psi \Rightarrow \varphi \rightarrow \psi} \rightarrow R
 \end{array}$$

Exemplo 19.7. Dê uma **LK-derivação** do sequente $\neg \varphi \vee \neg \psi \Rightarrow \neg(\varphi \wedge \psi)$

Usando as técnicas acima, começamos escrevendo o sequente final desejado na base.

$$\frac{}{\neg \varphi \vee \neg \psi \Rightarrow \neg(\varphi \wedge \psi)}$$

Os conectivos principais disponíveis das **sentenças** no sequente final são o símbolo $\vee$ e o símbolo $\neg$. Funcionaria aplicar aqui ou a regra $\vee L$ ou a regra $\neg R$, mas começamos com a regra $\neg R$ porque ela evita, por ora, a divisão em dois ramos:

$$\frac{\frac{}{\varphi \wedge \psi, \neg \varphi \vee \neg \psi \Rightarrow} \neg L}{\neg \varphi \vee \neg \psi \Rightarrow \neg(\varphi \wedge \psi)} \neg R$$

Agora temos a escolha de olhar para a regra $\wedge L$ ou a regra $\vee L$. Vejamos o que acontece quando aplicamos a regra $\wedge L$: temos a escolha de começar com o sequente $\varphi, \neg \varphi \vee \neg \psi \Rightarrow$ ou com o sequente $\psi, \neg \varphi \vee \neg \psi \Rightarrow$. Como a **derivação** é simétrica em relação a φ e ψ , vamos com o primeiro:

$$\frac{\frac{\frac{}{\varphi, \neg \varphi \vee \neg \psi \Rightarrow} \neg L}{\varphi \wedge \psi, \neg \varphi \vee \neg \psi \Rightarrow} \wedge L}{\neg \varphi \vee \neg \psi \Rightarrow \neg(\varphi \wedge \psi)} \neg R$$

Continuando a preencher a **derivação**, vemos que esbarramos em um problema:

$$\begin{array}{c}
 \frac{}{\varphi \Rightarrow \varphi} \neg L \quad \frac{}{\varphi \Rightarrow \psi} \neg L \\
 \frac{}{\neg \varphi, \varphi \Rightarrow} \neg L \quad \frac{}{\neg \psi, \varphi \Rightarrow} \neg L \\
 \frac{}{\neg \varphi \vee \neg \psi, \varphi \Rightarrow} \vee L \\
 \frac{}{\varphi, \neg \varphi \vee \neg \psi \Rightarrow} XL \\
 \frac{}{\varphi \wedge \psi, \neg \varphi \vee \neg \psi \Rightarrow} \wedge L \\
 \frac{}{\neg \varphi \vee \neg \psi \Rightarrow \neg(\varphi \wedge \psi)} \neg R
 \end{array}$$

O topo do ramo direito não pode ser reduzido mais, e ele não pode ser levado, por meio de inferências estruturais, a um sequente inicial, então este não é o caminho certo a tomar. Assim, claramente, foi um erro aplicar a regra $\wedge L$ acima. Voltando ao que tínhamos antes e executando a regra $\vee L$ em vez disso, obtemos

$$\frac{\frac{\frac{\overline{\neg\varphi, \varphi \wedge \psi \Rightarrow}}{\neg\varphi \vee \neg\psi, \varphi \wedge \psi \Rightarrow} \vee L \quad \frac{\overline{\neg\psi, \varphi \wedge \psi \Rightarrow}}{\varphi \wedge \psi, \neg\varphi \vee \neg\psi \Rightarrow} \text{XL}}{\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi)} \neg R$$

Completando cada ramo como fizemos antes, obtemos

$$\frac{\frac{\frac{\overline{\varphi \Rightarrow \varphi}}{\varphi \wedge \psi \Rightarrow \varphi} \wedge L \quad \frac{\overline{\psi \Rightarrow \psi}}{\varphi \wedge \psi \Rightarrow \psi} \wedge L}{\neg\varphi, \varphi \wedge \psi \Rightarrow} \neg L \quad \frac{\overline{\neg\psi, \varphi \wedge \psi \Rightarrow}}{\neg\varphi \vee \neg\psi, \varphi \wedge \psi \Rightarrow} \vee L}{\frac{\frac{\overline{\neg\varphi \vee \neg\psi, \varphi \wedge \psi \Rightarrow}}{\varphi \wedge \psi, \neg\varphi \vee \neg\psi \Rightarrow} \text{XL}}{\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi)} \neg R$$

(Poderíamos ter executado as regras de $\wedge$ mais abaixo do que as regras de $\neg$ nesses passos e ainda assim obtido uma **derivação** correta).

Exemplo 19.8. Até agora não usamos a regra de contração, mas às vezes ela é necessária. Eis um exemplo em que isso acontece. Suponha que queiramos provar $\Rightarrow \varphi \vee \neg\varphi$. Aplicar $\vee R$ de trás para frente nos daria uma destas duas **derivações**:

$$\frac{\overline{\Rightarrow \varphi}}{\Rightarrow \varphi \vee \neg\varphi} \vee R \quad \frac{\frac{\overline{\varphi \Rightarrow}}{\Rightarrow \neg\varphi} \neg R}{\Rightarrow \varphi \vee \neg\varphi} \vee R$$

Nenhuma destas, é claro, termina em um sequente inicial. O truque é perceber que a regra de contração nos permite combinar duas cópias de **a sentença** em uma só—e quando estamos buscando uma prova, isto é, indo de baixo para cima, podemos manter uma cópia de $\varphi \vee \neg\varphi$ na premissa, por exemplo,

$$\frac{\frac{\overline{\Rightarrow \varphi \vee \neg\varphi, \varphi}}{\Rightarrow \varphi \vee \neg\varphi, \varphi \vee \neg\varphi} \vee R}{\Rightarrow \varphi \vee \neg\varphi} \text{CR}$$

Agora podemos aplicar $\vee R$ uma segunda vez, e também obter $\neg\varphi$, o que leva a uma **derivação** completa.

$$\frac{\frac{\frac{\overline{\varphi \Rightarrow \varphi}}{\Rightarrow \varphi, \neg\varphi} \neg R \quad \frac{\overline{\Rightarrow \varphi, \varphi \vee \neg\varphi}}{\Rightarrow \varphi \vee \neg\varphi, \varphi} \vee R}{\Rightarrow \varphi \vee \neg\varphi, \varphi} \text{XR}}{\frac{\overline{\Rightarrow \varphi \vee \neg\varphi, \varphi \vee \neg\varphi}}{\Rightarrow \varphi \vee \neg\varphi} \vee R} \text{CR}$$

19.6. EXEMPLOS DE DERIVAÇÕES

Problema 19.1. Dê *derivações* dos seguintes sequentes:

1. $\varphi \wedge (\psi \wedge \chi) \Rightarrow (\varphi \wedge \psi) \wedge \chi.$
2. $\varphi \vee (\psi \vee \chi) \Rightarrow (\varphi \vee \psi) \vee \chi.$
3. $\varphi \rightarrow (\psi \rightarrow \chi) \Rightarrow \psi \rightarrow (\varphi \rightarrow \chi).$
4. $\varphi \Rightarrow \neg\neg\varphi.$

Problema 19.2. Dê *derivações* dos seguintes sequentes:

1. $(\varphi \vee \psi) \rightarrow \chi \Rightarrow \varphi \rightarrow \chi.$
2. $(\varphi \rightarrow \chi) \wedge (\psi \rightarrow \chi) \Rightarrow (\varphi \vee \psi) \rightarrow \chi.$
3. $\Rightarrow \neg(\varphi \wedge \neg\varphi).$
4. $\psi \rightarrow \varphi \Rightarrow \neg\varphi \rightarrow \neg\psi.$
5. $\Rightarrow (\varphi \rightarrow \neg\varphi) \rightarrow \neg\varphi.$
6. $\Rightarrow \neg(\varphi \rightarrow \psi) \rightarrow \neg\psi.$
7. $\varphi \rightarrow \chi \Rightarrow \neg(\varphi \wedge \neg\chi).$
8. $\varphi \wedge \neg\chi \Rightarrow \neg(\varphi \rightarrow \chi).$
9. $\varphi \vee \psi, \neg\psi \Rightarrow \varphi.$
10. $\neg\varphi \vee \neg\psi \Rightarrow \neg(\varphi \wedge \psi).$
11. $\Rightarrow (\neg\varphi \wedge \neg\psi) \rightarrow \neg(\varphi \vee \psi).$
12. $\Rightarrow \neg(\varphi \vee \psi) \rightarrow (\neg\varphi \wedge \neg\psi).$

Problema 19.3. Dê *derivações* dos seguintes sequentes:

1. $\neg(\varphi \rightarrow \psi) \Rightarrow \varphi.$
2. $\neg(\varphi \wedge \psi) \Rightarrow \neg\varphi \vee \neg\psi.$
3. $\varphi \rightarrow \psi \Rightarrow \neg\varphi \vee \psi.$
4. $\Rightarrow \neg\neg\varphi \rightarrow \varphi.$
5. $\varphi \rightarrow \psi, \neg\varphi \rightarrow \psi \Rightarrow \psi.$
6. $(\varphi \wedge \psi) \rightarrow \chi \Rightarrow (\varphi \rightarrow \chi) \vee (\psi \rightarrow \chi).$
7. $(\varphi \rightarrow \psi) \rightarrow \varphi \Rightarrow \varphi.$
8. $\Rightarrow (\varphi \rightarrow \psi) \vee (\psi \rightarrow \chi).$

(Todos esses requerem a regra CR.)

19.7 Derivações com Quantificadores

Exemplo 19.9. Dê uma **LK-derivação** do sequente $\exists x \neg \varphi(x) \Rightarrow \neg \forall x \varphi(x)$.

fol:seq:prq:
sec

Ao lidar com quantificadores, temos de cuidar para não violar a condição de variável própria, e às vezes isso exige que brinquemos com a ordem de execução de certas inferências. Em geral, ajuda tentar cuidar primeiro das regras sujeitas à condição de variável própria (elas ficarão mais abaixo na prova finalizada). Além disso, é uma boa ideia tentar olhar adiante e adivinhar como o sequente inicial poderia ser. No nosso caso, ele terá de ser algo como $\varphi(a) \Rightarrow \varphi(a)$. Isso significa que, ao “reverter” as regras de quantificador, teremos de escolher o mesmo termo—que chamaremos de a —tanto para a regra $\forall$ quanto para a regra $\exists$. Se escolhêssemos termos diferentes para cada regra, acabaríamos com algo como $\varphi(a) \Rightarrow \varphi(b)$, que, é claro, não é derivável.

Começando como de costume, escrevemos

$$\frac{}{\exists x \neg \varphi(x) \Rightarrow \neg \forall x \varphi(x)}$$

Poderíamos executar a regra $\exists L$ ou a regra $\neg R$. Como a regra $\exists L$ está sujeita à condição de variável própria, é uma boa ideia cuidar dela mais cedo do que mais tarde, então faremos essa primeiro.

$$\frac{\frac{}{\neg \varphi(a) \Rightarrow \neg \forall x \varphi(x)}}{\exists x \neg \varphi(x) \Rightarrow \neg \forall x \varphi(x)} \exists L$$

Aplicando as regras $\neg L$ e $\neg R$ de trás para frente, obtemos

$$\frac{\frac{\frac{\frac{}{\forall x \varphi(x) \Rightarrow \varphi(a)}}{\neg \varphi(a), \forall x \varphi(x) \Rightarrow} \neg L}{\forall x \varphi(x), \neg \varphi(a) \Rightarrow} XL}{\neg \varphi(a) \Rightarrow \neg \forall x \varphi(x)} \neg R}{\exists x \neg \varphi(x) \Rightarrow \neg \forall x \varphi(x)} \exists L$$

Neste ponto, nossa única opção é executar a regra $\forall L$. Como essa regra não está sujeita à restrição de variável própria, estamos livres. Lembre-se, queremos tentar obter um sequente inicial (da forma $\varphi(a) \Rightarrow \varphi(a)$), então devemos escolher a como nosso argumento para φ ao aplicarmos a regra.

$$\frac{\frac{\frac{\frac{\frac{}{\varphi(a) \Rightarrow \varphi(a)}}{\forall x \varphi(x) \Rightarrow \varphi(a)} \forall L}{\neg \varphi(a), \forall x \varphi(x) \Rightarrow} \neg L}{\forall x \varphi(x), \neg \varphi(a) \Rightarrow} XL}{\neg \varphi(a) \Rightarrow \neg \forall x \varphi(x)} \neg R}{\exists x \neg \varphi(x) \Rightarrow \neg \forall x \varphi(x)} \exists L$$

19.8. NOÇÕES DA TEORIA DA PROVA

É importante, especialmente ao lidar com quantificadores, verificar com cuidado neste ponto que a condição de variável própria não foi violada. Como a única regra que aplicamos sujeita à condição de variável própria foi $\exists L$, e a variável própria a não ocorre em seu sequente inferior (o sequente final), esta é uma **derivação** correta.

Problema 19.4. Dê **derivções** dos seguintes sequentes:

1. $\Rightarrow (\forall x \varphi(x) \wedge \forall y \psi(y)) \rightarrow \forall z (\varphi(z) \wedge \psi(z)).$
2. $\Rightarrow (\exists x \varphi(x) \vee \exists y \psi(y)) \rightarrow \exists z (\varphi(z) \vee \psi(z)).$
3. $\forall x (\varphi(x) \rightarrow \psi) \Rightarrow \exists y \varphi(y) \rightarrow \psi.$
4. $\forall x \neg \varphi(x) \Rightarrow \neg \exists x \varphi(x).$
5. $\Rightarrow \neg \exists x \varphi(x) \rightarrow \forall x \neg \varphi(x).$
6. $\Rightarrow \neg \exists x \forall y ((\varphi(x, y) \rightarrow \neg \varphi(y, y)) \wedge (\neg \varphi(y, y) \rightarrow \varphi(x, y))).$

Problema 19.5. Dê **derivções** dos seguintes sequentes:

1. $\Rightarrow \neg \forall x \varphi(x) \rightarrow \exists x \neg \varphi(x).$
2. $(\forall x \varphi(x) \rightarrow \psi) \Rightarrow \exists y (\varphi(y) \rightarrow \psi).$
3. $\Rightarrow \exists x (\varphi(x) \rightarrow \forall y \varphi(y)).$

(Todos esses requerem a regra CR.)

Esta seção reúne as definições da relação de demonstrabilidade e de consistência para a dedução natural.

`content/first-order-logic/sequent-calculus/proof-theoretic-notions.tex`

19.8 Noções da Teoria da Prova

fol:seq:ptn: sec Assim como definimos uma série de noções semânticas importantes (validade, explanation consequência, satisfatibilidade), definimos agora as *noções da teoria da prova* correspondentes. Elas não são definidas por apelo à satisfação de **sentenças** em **estruturas**, mas por apelo à **derivabilidade** ou à **não derivabilidade** de certos sequentes. Foi uma descoberta importante que essas noções coincidem. Que elas coincidem é o conteúdo dos teoremas da *correção* e da *completude*.

Definição 19.10 (Teoremas). A **sentença** φ é um *teorema* se há a **derivção** em **LK** do sequente $\Rightarrow \varphi$. Escrevemos $\vdash \varphi$ se φ é um teorema e $\nvdash \varphi$ se não é.

Definição 19.11 (Derivabilidade). A sentença φ é *derivável* de um conjunto de sentenças Γ , $\Gamma \vdash \varphi$, se, e somente se, há um subconjunto finito $\Gamma_0 \subseteq \Gamma$ e uma sequência Γ'_0 das sentenças em Γ_0 tal que **LK** deriva $\Gamma'_0 \Rightarrow \varphi$. Se φ não é derivável de Γ , escrevemos $\Gamma \nvdash \varphi$.

Por causa das regras de contração, enfraquecimento e troca, a ordem e o número de sentenças em Γ'_0 não importam: se um sequente $\Gamma'_0 \Rightarrow \varphi$ é derivável, então também o é $\Gamma''_0 \Rightarrow \varphi$ para qualquer Γ''_0 que contenha as mesmas sentenças que Γ'_0 . Por exemplo, se $\Gamma_0 = \{\psi, \chi\}$, então tanto $\Gamma'_0 = \langle \psi, \psi, \chi \rangle$ quanto $\Gamma''_0 = \langle \chi, \chi, \psi \rangle$ são sequências que contêm apenas as sentenças em Γ_0 . Se um sequente contendo uma delas é derivável, o outro também é, por exemplo:

$$\begin{array}{c} \vdots \\ \vdots \\ \frac{\psi, \psi, \chi \Rightarrow \varphi}{\psi, \chi \Rightarrow \varphi} \text{CL} \\ \frac{\psi, \chi \Rightarrow \varphi}{\chi, \psi \Rightarrow \varphi} \text{XL} \\ \frac{\chi, \psi \Rightarrow \varphi}{\chi, \chi, \psi \Rightarrow \varphi} \text{WL} \end{array}$$

De agora em diante, diremos que, se Γ_0 é um conjunto finito de sentenças, então $\Gamma_0 \Rightarrow \varphi$ é qualquer sequente em que o antecedente é uma sequência de sentenças em Γ_0 , e incluiremos tacitamente contrações, trocas e enfraquecimentos se necessário.

Definição 19.12 (Consistência). Um conjunto de sentenças Γ é *inconsistente* se, e somente se, há um subconjunto finito $\Gamma_0 \subseteq \Gamma$ tal que **LK** deriva $\Gamma_0 \Rightarrow$. Se Γ não é inconsistente, isto é, se para todo $\Gamma_0 \subseteq \Gamma$ finito, **LK** não deriva $\Gamma_0 \Rightarrow$, dizemos que é *consistente*.

Proposição 19.13 (Reflexividade). Se $\varphi \in \Gamma$, então $\Gamma \vdash \varphi$.

*fol:seq:ptn:
prop:reflexivity*

Prova. O sequente inicial $\varphi \Rightarrow \varphi$ é derivável, e $\{\varphi\} \subseteq \Gamma$. □

Proposição 19.14 (Monotonicidade). Se $\Gamma \subseteq \Delta$ e $\Gamma \vdash \varphi$, então $\Delta \vdash \varphi$.

*fol:seq:ptn:
prop:monotonicity*

Prova. Suponha $\Gamma \vdash \varphi$, isto é, há um $\Gamma_0 \subseteq \Gamma$ finito tal que $\Gamma_0 \Rightarrow \varphi$ é derivável. Como $\Gamma \subseteq \Delta$, então Γ_0 também é um subconjunto finito de Δ . A derivação de $\Gamma_0 \Rightarrow \varphi$ mostra, assim, também que $\Delta \vdash \varphi$. □

Proposição 19.15 (Transitividade). Se $\Gamma \vdash \varphi$ e $\{\varphi\} \cup \Delta \vdash \psi$, então $\Gamma \cup \Delta \vdash \psi$.

*fol:seq:ptn:
prop:transitivity*

Prova. Se $\Gamma \vdash \varphi$, há um $\Gamma_0 \subseteq \Gamma$ finito e a derivação π_0 de $\Gamma_0 \Rightarrow \varphi$. Se $\{\varphi\} \cup \Delta \vdash \psi$, então, para algum subconjunto finito $\Delta_0 \subseteq \Delta$, há a derivação π_1 de $\varphi, \Delta_0 \Rightarrow \psi$. Considere a seguinte derivação:

19.9. DERIVABILIDADE E CONSISTÊNCIA

$$\frac{\begin{array}{c} \vdots \pi_0 \\ \Gamma_0 \Rightarrow \varphi \end{array} \quad \begin{array}{c} \vdots \pi_1 \\ \varphi, \Delta_0 \Rightarrow \psi \end{array}}{\Gamma_0, \Delta_0 \Rightarrow \psi} \text{Cut}$$

Como $\Gamma_0 \cup \Delta_0 \subseteq \Gamma \cup \Delta$, isso mostra $\Gamma \cup \Delta \vdash \psi$. $\square$

Note que isso significa, em particular, que se $\Gamma \vdash \varphi$ e $\varphi \vdash \psi$, então $\Gamma \vdash \psi$. Segue-se também que se $\varphi_1, \dots, \varphi_n \vdash \psi$ e $\Gamma \vdash \varphi_i$ para cada i , então $\Gamma \vdash \psi$.

fol:seq:ptn: prop:incons **Proposição 19.16.** *Γ é inconsistente se, e somente se, $\Gamma \vdash \varphi$ para toda sentença φ .*

Prova. Exercício. $\square$

Problema 19.6. Prove **Proposição 19.16**

fol:seq:ptn: prop:proves-compact **Proposição 19.17 (Compacidade).**

1. Se $\Gamma \vdash \varphi$, então há um subconjunto finito $\Gamma_0 \subseteq \Gamma$ tal que $\Gamma_0 \vdash \varphi$.
2. Se todo subconjunto finito de Γ é consistente, então Γ é consistente.

Prova. 1. Se $\Gamma \vdash \varphi$, então há um subconjunto finito $\Gamma_0 \subseteq \Gamma$ tal que o sequente $\Gamma_0 \Rightarrow \varphi$ tem **a derivação**. Consequentemente, $\Gamma_0 \vdash \varphi$.

2. Se Γ é inconsistente, há um subconjunto finito $\Gamma_0 \subseteq \Gamma$ tal que **LK deriva** $\Gamma_0 \Rightarrow \perp$. Mas então Γ_0 é um subconjunto finito de Γ que é inconsistente. $\square$

content/first-order-logic/sequent-calculus/provability-consistency.tex

19.9 Derivabilidade e Consistência

fol:seq:prv: sec Estabeleceremos agora uma série de propriedades da relação de **derivabilidade**. Elas são interessantes por si mesmas, mas cada uma terá um papel na prova do teorema da completude.

fol:seq:prv: prop:provability-contr **Proposição 19.18.** *Se $\Gamma \vdash \varphi$ e $\Gamma \cup \{\varphi\}$ é inconsistente, então Γ é inconsistente.*

Prova. Há Γ_0 e $\Gamma_1 \subseteq \Gamma$ finitos tais que **LK deriva** $\Gamma_0 \Rightarrow \varphi$ e $\varphi, \Gamma_1 \Rightarrow \perp$. Seja π_0 a **LK-derivação** de $\Gamma_0 \Rightarrow \varphi$ e π_1 a **LK-derivação** de $\Gamma_1, \varphi \Rightarrow \perp$. Podemos então derivar

CAPÍTULO 19. O CÁLCULO DE SEQUENTES

$$\frac{\frac{\vdots \pi_0}{\Gamma_0 \Rightarrow \varphi} \quad \frac{\vdots \pi_1}{\varphi, \Gamma_1 \Rightarrow}}{\Gamma_0, \Gamma_1 \Rightarrow} \text{Cut}$$

Como $\Gamma_0 \subseteq \Gamma$ e $\Gamma_1 \subseteq \Gamma$, $\Gamma_0 \cup \Gamma_1 \subseteq \Gamma$; portanto, Γ é inconsistente. $\square$

Proposição 19.19. $\Gamma \vdash \varphi$ se, e somente se, $\Gamma \cup \{\neg\varphi\}$ é inconsistente.

fol:seq:prv:
prop:prov-incons

Prova. Primeiro, suponha $\Gamma \vdash \varphi$, isto é, há a derivação π_0 de $\Gamma \Rightarrow \varphi$. Acrescentando uma regra $\neg$ L, obtemos a derivação de $\neg\varphi, \Gamma \Rightarrow$, isto é, $\Gamma \cup \{\neg\varphi\}$ é inconsistente.

Se $\Gamma \cup \{\neg\varphi\}$ é inconsistente, há a derivação π_1 de $\neg\varphi, \Gamma \Rightarrow$. O seguinte é a derivação de $\Gamma \Rightarrow \varphi$:

$$\frac{\frac{\varphi \Rightarrow \varphi}{\Rightarrow \varphi, \neg\varphi} \neg R \quad \frac{\vdots \pi_1}{\neg\varphi, \Gamma \Rightarrow}}{\Gamma \Rightarrow \varphi} \text{Cut}$$

$\square$

Problema 19.7. Prove que $\Gamma \vdash \neg\varphi$ se, e somente se, $\Gamma \cup \{\varphi\}$ é inconsistente.

Proposição 19.20. Se $\Gamma \vdash \varphi$ e $\neg\varphi \in \Gamma$, então Γ é inconsistente.

fol:seq:prv:
prop:explicit-inc

Prova. Suponha $\Gamma \vdash \varphi$ e $\neg\varphi \in \Gamma$. Então há a derivação π de um sequente $\Gamma_0 \Rightarrow \varphi$. O sequente $\neg\varphi, \Gamma_0 \Rightarrow$ também é derivável:

$$\frac{\frac{\vdots \pi}{\Gamma_0 \Rightarrow \varphi} \quad \frac{\frac{\varphi \Rightarrow \varphi}{\neg\varphi, \varphi \Rightarrow} \neg L}{\frac{\varphi, \neg\varphi \Rightarrow}{\Gamma_0, \neg\varphi \Rightarrow} \text{Cut}} \text{XL}$$

Como $\neg\varphi \in \Gamma$ e $\Gamma_0 \subseteq \Gamma$, isso mostra que Γ é inconsistente. $\square$

Proposição 19.21. Se $\Gamma \cup \{\varphi\}$ e $\Gamma \cup \{\neg\varphi\}$ são ambos inconsistentes, então Γ é inconsistente.

fol:seq:prv:
prop:provability-exhaustive

Prova. Há conjuntos finitos $\Gamma_0 \subseteq \Gamma$ e $\Gamma_1 \subseteq \Gamma$ e **LK-derivações** π_0 e π_1 de $\varphi, \Gamma_0 \Rightarrow$ e $\neg\varphi, \Gamma_1 \Rightarrow$, respectivamente. Podemos então derivar

$$\frac{\frac{\frac{\vdots \pi_0}{\varphi, \Gamma_0 \Rightarrow} \neg R}{\Gamma_0 \Rightarrow \neg\varphi} \quad \frac{\vdots \pi_1}{\neg\varphi, \Gamma_1 \Rightarrow}}{\Gamma_0, \Gamma_1 \Rightarrow} \text{Cut}$$

19.10. DERIVABILIDADE E OS CONECTIVOS PROPOSICIONAIS

Como $\Gamma_0 \subseteq \Gamma$ e $\Gamma_1 \subseteq \Gamma$, $\Gamma_0 \cup \Gamma_1 \subseteq \Gamma$. Portanto, Γ é inconsistente. $\square$

content/first-order-logic/sequent-calculus/provability-propositional.tex

19.10 Derivabilidade e os Conectivos Proposicionais

fol:seq:ppr: Estabelecemos que a relação de **derivabilidade** $\vdash$ do cálculo de seqüentes é **forte** o suficiente para estabelecer alguns fatos básicos envolvendo os conectivos **sec** proposicionais, tais como $\varphi \wedge \psi \vdash \varphi$ e $\varphi, \varphi \rightarrow \psi \vdash \psi$ (modus ponens). Esses fatos são necessários para a prova do teorema da completude. **explanation**

Proposição 19.22.

fol:seq:ppr:
prop:provability-land
fol:seq:ppr:
prop:provability-land-left
fol:seq:ppr:
prop:provability-land-right

1. Tanto $\varphi \wedge \psi \vdash \varphi$ quanto $\varphi \wedge \psi \vdash \psi$.
2. $\varphi, \psi \vdash \varphi \wedge \psi$.

Prova. 1. Ambos os seqüentes $\varphi \wedge \psi \Rightarrow \varphi$ e $\varphi \wedge \psi \Rightarrow \psi$ são **deriváveis**:

$$\frac{\varphi \Rightarrow \varphi}{\varphi \wedge \psi \Rightarrow \varphi} \wedge L \quad \frac{\psi \Rightarrow \psi}{\varphi \wedge \psi \Rightarrow \psi} \wedge L$$

2. Eis a **derivação** do seqüente $\varphi, \psi \Rightarrow \varphi \wedge \psi$:

$$\frac{\varphi \Rightarrow \varphi \quad \psi \Rightarrow \psi}{\varphi, \psi \Rightarrow \varphi \wedge \psi} \wedge R$$

$\square$

Proposição 19.23.

fol:seq:ppr:
prop:provability-lor

1. $\varphi \vee \psi, \neg\varphi, \neg\psi$ é inconsistente.
2. Tanto $\varphi \vdash \varphi \vee \psi$ quanto $\psi \vdash \varphi \vee \psi$.

Prova. 1. Damos a **derivação** do seqüente $\varphi \vee \psi, \neg\varphi, \neg\psi \Rightarrow$:

$$\frac{\frac{\frac{\varphi \Rightarrow \varphi}{\neg\varphi, \varphi \Rightarrow} \neg L}{\varphi, \neg\varphi, \neg\psi \Rightarrow} \quad \frac{\frac{\frac{\psi \Rightarrow \psi}{\neg\psi, \psi \Rightarrow} \neg L}{\psi, \neg\varphi, \neg\psi \Rightarrow} \vee L}{\varphi \vee \psi, \neg\varphi, \neg\psi \Rightarrow}$$

(Lembre-se de que linhas de inferência duplas indicam várias inferências de enfraquecimento, contração e troca.)

2. Ambos os seqüentes $\varphi \Rightarrow \varphi \vee \psi$ e $\psi \Rightarrow \varphi \vee \psi$ têm **derivações**:

$$\frac{\varphi \Rightarrow \varphi}{\varphi \Rightarrow \varphi \vee \psi} \vee R \quad \frac{\psi \Rightarrow \psi}{\psi \Rightarrow \varphi \vee \psi} \vee R$$

$\square$

Proposição 19.24.

1. $\varphi, \varphi \rightarrow \psi \vdash \psi$.
2. Tanto $\neg\varphi \vdash \varphi \rightarrow \psi$ quanto $\psi \vdash \varphi \rightarrow \psi$.

*fol:seq:ppr:
prop:provability-lif
fol:seq:ppr:
prop:provability-lif-left
fol:seq:ppr:
prop:provability-lif-right*

Prova. 1. O sequente $\varphi \rightarrow \psi, \varphi \Rightarrow \psi$ é **derivável**:

$$\frac{\varphi \Rightarrow \varphi \quad \psi \Rightarrow \psi}{\varphi \rightarrow \psi, \varphi \Rightarrow \psi} \rightarrow L$$

2. Ambos os sequentes $\neg\varphi \Rightarrow \varphi \rightarrow \psi$ e $\psi \Rightarrow \varphi \rightarrow \psi$ são **deriváveis**:

$$\frac{\frac{\frac{\varphi \Rightarrow \varphi}{\neg\varphi, \varphi \Rightarrow} \neg L \quad \frac{}{\varphi, \neg\varphi \Rightarrow} XL}{\varphi, \neg\varphi \Rightarrow \psi} WR \quad \frac{\psi \Rightarrow \psi}{\varphi, \psi \Rightarrow \psi} WL \quad \frac{}{\neg\varphi \Rightarrow \varphi \rightarrow \psi} \rightarrow R \quad \frac{}{\psi \Rightarrow \varphi \rightarrow \psi} \rightarrow R \quad \square$$

`content/first-order-logic/sequent-calculus/provability-quantifiers.tex`

19.11 Derivabilidade e os Quantificadores

explanation O teorema da completude também exige que as regras do cálculo de sequentes produzam os fatos sobre $\vdash$ estabelecidos nesta seção.

*fol:seq:qpr:
sec*

Teorema 19.25. *Se c é uma constante que não ocorre em Γ ou $\varphi(x)$ e $\Gamma \vdash \varphi(c)$, então $\Gamma \vdash \forall x \varphi(x)$.*

*fol:seq:qpr:
thm:strong-generalization*

Prova. Seja π_0 uma **LK-derivação** de $\Gamma_0 \Rightarrow \varphi(c)$ para algum $\Gamma_0 \subseteq \Gamma$ finito. Acrescentando uma inferência $\forall R$, obtemos a **derivação** de $\Gamma_0 \Rightarrow \forall x \varphi(x)$, já que c não ocorre em Γ ou $\varphi(x)$ e, assim, a condição de variável própria é satisfeita. $\square$

Proposição 19.26.

1. $\varphi(t) \vdash \exists x \varphi(x)$.
2. $\forall x \varphi(x) \vdash \varphi(t)$.

*fol:seq:qpr:
prop:provability-quantifiers*

Prova. 1. O sequente $\varphi(t) \Rightarrow \exists x \varphi(x)$ é **derivável**:

$$\frac{\varphi(t) \Rightarrow \varphi(t)}{\varphi(t) \Rightarrow \exists x \varphi(x)} \exists R$$

2. O sequente $\forall x \varphi(x) \Rightarrow \varphi(t)$ é **derivável**:

$$\frac{\varphi(t) \Rightarrow \varphi(t)}{\forall x \varphi(x) \Rightarrow \varphi(t)} \forall L$$

□

content/first-order-logic/sequent-calculus/soundness.tex

19.12 Correção

fol:seq:sou:
sec

Um sistema de **derivação**, tal como o cálculo de seqüentes, é *correto* se não pode derivar coisas que de fato não valem. A correção é, assim, uma espécie de propriedade de segurança garantida para sistemas de **derivação**. Dependendo de qual propriedade da teoria da prova está em questão, gostaríamos de saber, por exemplo, que

explanation

1. todo φ **derivável** é válido;
2. se a **sentença** é **derivável** de algumas outras, ela também é consequência delas;
3. se um conjunto de **sentenças** é inconsistente, ele é insatisfatível.

Essas são propriedades importantes de um sistema de **derivação**. Se alguma delas não vale, o sistema de **derivação** é deficiente—ele **deriva** demais. Consequentemente, estabelecer a correção de um sistema de **derivação** é da máxima importância.

Como todas essas propriedades da teoria da prova são definidas via **derivabilidade** no cálculo de seqüentes de certos seqüentes, provar (1)–(3) acima requer provar algo sobre as propriedades semânticas dos seqüentes **deriváveis**. Primeiro definiremos o que significa um seqüente ser *válido*, e então mostraremos que todo seqüente **derivável** é válido. (1)–(3) seguem-se então como corolários desse resultado.

Definição 19.27. A estrutura $\mathfrak{M}$ *satisfaz* um seqüente $\Gamma \Rightarrow \Delta$ se, e somente se, ou $\mathfrak{M} \not\models \varphi$ para algum $\varphi \in \Gamma$ ou $\mathfrak{M} \models \varphi$ para algum $\varphi \in \Delta$.

Um seqüente é *válido* se, e somente se, toda **estrutura** $\mathfrak{M}$ o satisfaz.

fol:seq:sou:
thm:sequent-soundness

Teorema 19.28 (Correção). Se **LK** *deriva* $\Theta \Rightarrow \Xi$, então $\Theta \Rightarrow \Xi$ é válido.

Prova. Seja π a **derivação** de $\Theta \Rightarrow \Xi$. Procedemos por indução sobre o número de inferências n em π .

Se o número de inferências é 0, então π consiste apenas em um seqüente inicial. Todo seqüente inicial $\varphi \Rightarrow \varphi$ é obviamente válido, pois para toda $\mathfrak{M}$, ou $\mathfrak{M} \not\models \varphi$ ou $\mathfrak{M} \models \varphi$.

Se o número de inferências é maior que 0, distinguimos casos de acordo com o tipo da inferência mais inferior. Pela hipótese de indução, podemos supor que as premissas dessa inferência são válidas, já que o número de inferências na **derivação** de qualquer premissa é menor que n .

Primeiro, consideramos as inferências possíveis com apenas uma premissa.

1. A última inferência é um enfraquecimento. Então $\Theta \Rightarrow \Xi$ é $\varphi, \Gamma \Rightarrow \Delta$ (se a última inferência é WL) ou $\Gamma \Rightarrow \Delta, \varphi$ (se é WR), e a **derivação** termina em uma de

$$\frac{\begin{array}{c} \vdots \\ \Gamma \Rightarrow \Delta \end{array}}{\varphi, \Gamma \Rightarrow \Delta} \text{WL} \qquad \frac{\begin{array}{c} \vdots \\ \Gamma \Rightarrow \Delta \end{array}}{\Gamma \Rightarrow \Delta, \varphi} \text{WR}$$

Pela hipótese de indução, $\Gamma \Rightarrow \Delta$ é válido, isto é, para toda **estrutura** $\mathfrak{M}$, ou há algum $\chi \in \Gamma$ tal que $\mathfrak{M} \not\models \chi$ ou há algum $\chi \in \Delta$ tal que $\mathfrak{M} \models \chi$.

Se $\mathfrak{M} \not\models \chi$ para algum $\chi \in \Gamma$, então $\chi \in \Theta$ também, já que $\Theta = \varphi, \Gamma$, e assim $\mathfrak{M} \not\models \chi$ para algum $\chi \in \Theta$. Analogamente, se $\mathfrak{M} \models \chi$ para algum $\chi \in \Delta$, como $\chi \in \Xi$, $\mathfrak{M} \models \chi$ para algum $\chi \in \Xi$. Consequentemente, $\Theta \Rightarrow \Xi$ é válido.

2. A última inferência é $\neg$ L: Então a premissa da última inferência é $\Gamma \Rightarrow \Delta, \varphi$ e a conclusão é $\neg\varphi, \Gamma \Rightarrow \Delta$, isto é, a **derivação** termina em

$$\frac{\begin{array}{c} \vdots \\ \Gamma \Rightarrow \Delta, \varphi \end{array}}{\neg\varphi, \Gamma \Rightarrow \Delta} \neg\text{L}$$

e $\Theta = \neg\varphi, \Gamma$ enquanto $\Xi = \Delta$.

A hipótese de indução nos diz que $\Gamma \Rightarrow \Delta, \varphi$ é válido, isto é, para toda $\mathfrak{M}$, ou (a) para algum $\chi \in \Gamma$, $\mathfrak{M} \not\models \chi$, ou (b) para algum $\chi \in \Delta$, $\mathfrak{M} \models \chi$, ou (c) $\mathfrak{M} \models \varphi$. Queremos mostrar que $\Theta \Rightarrow \Xi$ também é válido. Seja $\mathfrak{M}$ a **estrutura**. Se (a) vale, então há $\chi \in \Gamma$ tal que $\mathfrak{M} \not\models \chi$, mas $\chi \in \Theta$ também. Se (b) vale, há $\chi \in \Delta$ tal que $\mathfrak{M} \models \chi$, mas $\chi \in \Xi$ também. Por fim, se $\mathfrak{M} \models \varphi$, então $\mathfrak{M} \not\models \neg\varphi$. Como $\neg\varphi \in \Theta$, há $\chi \in \Theta$ tal que $\mathfrak{M} \not\models \chi$. Consequentemente, $\Theta \Rightarrow \Xi$ é válido.

3. A última inferência é $\neg$ R: Exercício.
4. A última inferência é $\wedge$ L: Há duas variantes: $\varphi \wedge \psi$ pode ser inferido à esquerda a partir de φ ou de ψ no lado esquerdo da premissa. No primeiro caso, π termina em

$$\frac{\begin{array}{c} \vdots \\ \varphi, \Gamma \Rightarrow \Delta \end{array}}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta} \wedge\text{L}$$

19.12. CORREÇÃO

e $\Theta = \varphi \wedge \psi, \Gamma$ enquanto $\Xi = \Delta$. Considere a estrutura $\mathfrak{M}$. Como, pela hipótese de indução, $\varphi, \Gamma \Rightarrow \Delta$ é válido, (a) $\mathfrak{M} \models \varphi$, (b) $\mathfrak{M} \models \chi$ para algum $\chi \in \Gamma$, ou (c) $\mathfrak{M} \models \chi$ para algum $\chi \in \Delta$. No caso (a), $\mathfrak{M} \models \varphi \wedge \psi$, então há $\chi \in \Theta$ (a saber, $\varphi \wedge \psi$) tal que $\mathfrak{M} \models \chi$. No caso (b), há $\chi \in \Gamma$ tal que $\mathfrak{M} \models \chi$, e $\chi \in \Theta$ também. No caso (c), há $\chi \in \Delta$ tal que $\mathfrak{M} \models \chi$, e $\chi \in \Xi$ também, já que $\Xi = \Delta$. Assim, em cada caso, $\mathfrak{M}$ satisfaz $\varphi \wedge \psi, \Gamma \Rightarrow \Delta$. Como $\mathfrak{M}$ era arbitrário, $\Gamma \Rightarrow \Delta$ é válido. O caso em que $\varphi \wedge \psi$ é inferido a partir de ψ é tratado da mesma forma, mudando φ para ψ .

5. A última inferência é $\vee R$: Há duas variantes: $\varphi \vee \psi$ pode ser inferido à direita a partir de φ ou de ψ no lado direito da premissa. No primeiro caso, π termina em

$$\frac{\begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \Gamma \Rightarrow \Delta, \varphi \end{array}}{\Gamma \Rightarrow \Delta, \varphi \vee \psi} \vee R$$

Agora $\Theta = \Gamma$ e $\Xi = \Delta, \varphi \vee \psi$. Considere a estrutura $\mathfrak{M}$. Como $\Gamma \Rightarrow \Delta, \varphi$ é válido, (a) $\mathfrak{M} \models \varphi$, (b) $\mathfrak{M} \models \chi$ para algum $\chi \in \Gamma$, ou (c) $\mathfrak{M} \models \chi$ para algum $\chi \in \Delta$. No caso (a), $\mathfrak{M} \models \varphi \vee \psi$. No caso (b), há $\chi \in \Gamma$ tal que $\mathfrak{M} \models \chi$. No caso (c), há $\chi \in \Delta$ tal que $\mathfrak{M} \models \chi$. Assim, em cada caso, $\mathfrak{M}$ satisfaz $\Gamma \Rightarrow \Delta, \varphi \vee \psi$, isto é, $\Theta \Rightarrow \Xi$. Como $\mathfrak{M}$ era arbitrário, $\Theta \Rightarrow \Xi$ é válido. O caso em que $\varphi \vee \psi$ é inferido a partir de ψ é tratado da mesma forma, mudando φ para ψ .

6. A última inferência é $\rightarrow R$: Então π termina em

$$\frac{\begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \varphi, \Gamma \Rightarrow \Delta, \psi \end{array}}{\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi} \rightarrow R$$

Novamente, a hipótese de indução diz que a premissa é válida; queremos mostrar que a conclusão também é válida. Seja $\mathfrak{M}$ arbitrário. Como $\varphi, \Gamma \Rightarrow \Delta, \psi$ é válido, ao menos um dos seguintes casos ocorre: (a) $\mathfrak{M} \models \varphi$, (b) $\mathfrak{M} \models \psi$, (c) $\mathfrak{M} \models \chi$ para algum $\chi \in \Gamma$, ou (d) $\mathfrak{M} \models \chi$ para algum $\chi \in \Delta$. Nos casos (a) e (b), $\mathfrak{M} \models \varphi \rightarrow \psi$ e assim há um $\chi \in \Delta, \varphi \rightarrow \psi$ tal que $\mathfrak{M} \models \chi$. No caso (c), para algum $\chi \in \Gamma$, $\mathfrak{M} \models \chi$. No caso (d), para algum $\chi \in \Delta$, $\mathfrak{M} \models \chi$. Em cada caso, $\mathfrak{M}$ satisfaz $\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi$. Como $\mathfrak{M}$ era arbitrário, $\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi$ é válido.

7. A última inferência é $\forall L$: Então há a fórmula $\varphi(x)$ e um termo fechado t tais que π termina em

$$\frac{\begin{array}{c} \vdots \\ \varphi(t), \Gamma \Rightarrow \Delta \end{array}}{\forall x \varphi(x), \Gamma \Rightarrow \Delta} \forall L$$

Queremos mostrar que a conclusão $\forall x \varphi(x), \Gamma \Rightarrow \Delta$ é válida. Considere a estrutura $\mathfrak{M}$. Como a premissa $\varphi(t), \Gamma \Rightarrow \Delta$ é válida, (a) $\mathfrak{M} \not\models \varphi(t)$, (b) $\mathfrak{M} \not\models \chi$ para algum $\chi \in \Gamma$, ou (c) $\mathfrak{M} \models \chi$ para algum $\chi \in \Delta$. No caso (a), por **Proposição 16.31**, se $\mathfrak{M} \models \forall x \varphi(x)$, então $\mathfrak{M} \models \varphi(t)$. Como $\mathfrak{M} \not\models \varphi(t)$, $\mathfrak{M} \not\models \forall x \varphi(x)$. Nos casos (b) e (c), $\mathfrak{M}$ também satisfaz $\forall x \varphi(x), \Gamma \Rightarrow \Delta$. Como $\mathfrak{M}$ era arbitrário, $\forall x \varphi(x), \Gamma \Rightarrow \Delta$ é válido.

8. A última inferência é $\exists R$: Exercício.
9. A última inferência é $\forall R$: Então há a fórmula $\varphi(x)$ e a símbolo de constante a tais que π termina em

$$\frac{\begin{array}{c} \vdots \\ \Gamma \Rightarrow \Delta, \varphi(a) \end{array}}{\Gamma \Rightarrow \Delta, \forall x \varphi(x)} \forall R$$

em que a condição de variável própria é satisfeita, isto é, a não ocorre em $\varphi(x)$, Γ ou Δ . Pela hipótese de indução, a premissa da última inferência é válida. Temos de mostrar que a conclusão também é válida, isto é, que para qualquer estrutura $\mathfrak{M}$, (a) $\mathfrak{M} \models \forall x \varphi(x)$, (b) $\mathfrak{M} \not\models \chi$ para algum $\chi \in \Gamma$, ou (c) $\mathfrak{M} \models \chi$ para algum $\chi \in \Delta$.

Suponha que $\mathfrak{M}$ seja a estrutura arbitrária. Se (b) ou (c) vale, terminamos, então suponha que nenhuma vale: para todo $\chi \in \Gamma$, $\mathfrak{M} \models \chi$, e para todo $\chi \in \Delta$, $\mathfrak{M} \not\models \chi$. Temos de mostrar que (a) vale, isto é, $\mathfrak{M} \models \forall x \varphi(x)$. Por **Proposição 16.19**, basta mostrar que $\mathfrak{M}, s \models \varphi(x)$ para toda atribuição de variáveis s . Então seja s uma atribuição de variáveis arbitrária. Considere a estrutura $\mathfrak{M}'$ que é exatamente como $\mathfrak{M}$ exceto que $a^{\mathfrak{M}'} = s(x)$. Por **Corolário 16.21**, para qualquer $\chi \in \Gamma$, $\mathfrak{M}' \models \chi$ já que a não ocorre em Γ , e para qualquer $\chi \in \Delta$, $\mathfrak{M}' \not\models \chi$. Mas a premissa é válida, então $\mathfrak{M}' \models \varphi(a)$. Por **Proposição 16.18**, $\mathfrak{M}', s \models \varphi(a)$, já que $\varphi(a)$ é uma sentença. Agora $s \sim_x s$ com $s(x) = \text{Val}_s^{\mathfrak{M}'}(a)$, já que definimos $\mathfrak{M}'$ exatamente desse modo. Assim **Proposição 16.23** se aplica, e obtemos $\mathfrak{M}', s \models \varphi(x)$. Como a não ocorre em $\varphi(x)$, por **Proposição 16.20**, $\mathfrak{M}, s \models \varphi(x)$. Como s era arbitrário, completamos a prova de que $\mathfrak{M}, s \models \varphi(x)$ para toda atribuição de variáveis.

10. A última inferência é $\exists L$: Exercício.

Agora consideremos as inferências possíveis com duas premissas.

19.12. CORREÇÃO

1. A última inferência é um corte: então π termina em

$$\frac{\begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \Gamma \Rightarrow \Delta, \varphi \end{array} \quad \begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \varphi, \Pi \Rightarrow \Lambda \end{array}}{\Gamma, \Pi \Rightarrow \Delta, \Lambda} \text{Cut}$$

Seja $\mathfrak{M}$ a estrutura. Pela hipótese de indução, as premissas são válidas, então $\mathfrak{M}$ satisfaz ambas as premissas. Distinguímos dois casos: (a) $\mathfrak{M} \not\models \varphi$ e (b) $\mathfrak{M} \models \varphi$. No caso (a), para que $\mathfrak{M}$ satisfaça a premissa esquerda, ela deve satisfazer $\Gamma \Rightarrow \Delta$. Mas então também satisfaz a conclusão. No caso (b), para que $\mathfrak{M}$ satisfaça a premissa direita, ela deve satisfazer $\Pi \setminus \Lambda$. Novamente, $\mathfrak{M}$ satisfaz a conclusão.

2. A última inferência é $\wedge R$. Então π termina em

$$\frac{\begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \Gamma \Rightarrow \Delta, \varphi \end{array} \quad \begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \Gamma \Rightarrow \Delta, \psi \end{array}}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi} \wedge R$$

Considere a estrutura $\mathfrak{M}$. Se $\mathfrak{M}$ satisfaz $\Gamma \Rightarrow \Delta$, terminamos. Então suponha que não. Como $\Gamma \Rightarrow \Delta, \varphi$ é válido pela hipótese de indução, $\mathfrak{M} \models \varphi$. Analogamente, como $\Gamma \Rightarrow \Delta, \psi$ é válido, $\mathfrak{M} \models \psi$. Mas então $\mathfrak{M} \models \varphi \wedge \psi$.

3. A última inferência é $\vee L$: Exercício.
4. A última inferência é $\rightarrow L$. Então π termina em

$$\frac{\begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \Gamma \Rightarrow \Delta, \varphi \end{array} \quad \begin{array}{c} \vdots \\ \vdots \\ \vdots \\ \psi, \Pi \Rightarrow \Lambda \end{array}}{\varphi \rightarrow \psi, \Gamma, \Pi \Rightarrow \Delta, \Lambda} \rightarrow L$$

Novamente, considere a estrutura $\mathfrak{M}$ e suponha que $\mathfrak{M}$ não satisfaça $\Gamma, \Pi \Rightarrow \Delta, \Lambda$. Temos de mostrar que $\mathfrak{M} \not\models \varphi \rightarrow \psi$. Se $\mathfrak{M}$ não satisfaz $\Gamma, \Pi \Rightarrow \Delta, \Lambda$, não satisfaz nem $\Gamma \Rightarrow \Delta$ nem $\Pi \Rightarrow \Lambda$. Como $\Gamma \Rightarrow \Delta, \varphi$ é válido, temos $\mathfrak{M} \models \varphi$. Como $\psi, \Pi \Rightarrow \Lambda$ é válido, temos $\mathfrak{M} \not\models \psi$. Mas então $\mathfrak{M} \not\models \varphi \rightarrow \psi$, que é o que queríamos mostrar. $\square$

Problema 19.8. Complete a prova de Teorema 19.28.

Corolário 19.29. Se $\vdash \varphi$, então φ é válido.

*fol:seq:sou:
cor:weak-soundness*

Corolário 19.30. Se $\Gamma \vdash \varphi$, então $\Gamma \models \varphi$.

fol:seq:sou:
cor:entailment-soundness

Prova. Se $\Gamma \vdash \varphi$, então, para algum subconjunto finito $\Gamma_0 \subseteq \Gamma$, há a derivação de $\Gamma_0 \Rightarrow \varphi$. Por Teorema 19.28, toda estrutura $\mathfrak{M}$ ou torna algum $\psi \in \Gamma_0$ falso ou torna φ verdadeiro. Logo, se $\mathfrak{M} \models \Gamma$, então também $\mathfrak{M} \models \varphi$. $\square$

Corolário 19.31. Se Γ é satisfatível, então é consistente.

fol:seq:sou:
cor:consistency-soundness

Prova. Provamos a contrapositiva. Suponha que Γ não seja consistente. Então há um $\Gamma_0 \subseteq \Gamma$ finito e a derivação de $\Gamma_0 \Rightarrow \perp$. Por Teorema 19.28, $\Gamma_0 \Rightarrow \perp$ é válido. Em outras palavras, para toda estrutura $\mathfrak{M}$, há $\chi \in \Gamma_0$ tal que $\mathfrak{M} \not\models \chi$, e como $\Gamma_0 \subseteq \Gamma$, esse χ também está em Γ . Assim, nenhuma $\mathfrak{M}$ satisfaz Γ , e Γ não é satisfatível. $\square$

content/first-order-logic/sequent-calculus/identity.tex

19.13 Derivações com Predicado de identidade

Derivações com predicado de identidade requerem sequentes iniciais e regras de inferência adicionais.

fol:seq:ide:
sec

Definição 19.32 (Sequentes iniciais para =). Se t é um termo fechado, então $\Rightarrow t = t$ é um sequente inicial.

As regras para = são (t_1 e t_2 são termos fechados):

$$\frac{t_1 = t_2, \Gamma \Rightarrow \Delta, \varphi(t_1)}{t_1 = t_2, \Gamma \Rightarrow \Delta, \varphi(t_2)} = \qquad \frac{t_1 = t_2, \Gamma \Rightarrow \Delta, \varphi(t_2)}{t_1 = t_2, \Gamma \Rightarrow \Delta, \varphi(t_1)} =$$

Exemplo 19.33. Se s e t são termos fechados, então $s = t, \varphi(s) \vdash \varphi(t)$:

$$\frac{\varphi(s) \Rightarrow \varphi(s)}{s = t, \varphi(s) \Rightarrow \varphi(s)} \text{WL} \quad \frac{s = t, \varphi(s) \Rightarrow \varphi(s)}{s = t, \varphi(s) \Rightarrow \varphi(t)} =$$

Isso pode ser familiar como o princípio da substitutibilidade de idênticos, ou Lei de Leibniz.

LK prova que = é simétrica e transitiva:

$$\frac{\Rightarrow t_1 = t_1}{t_1 = t_2 \Rightarrow t_1 = t_1} \text{WL} \quad \frac{t_1 = t_2 \Rightarrow t_1 = t_2}{t_2 = t_3, t_1 = t_2 \Rightarrow t_1 = t_2} \text{WL} \quad \frac{t_2 = t_3, t_1 = t_2 \Rightarrow t_1 = t_2}{t_2 = t_3, t_1 = t_2 \Rightarrow t_1 = t_3} = \quad \frac{t_2 = t_3, t_1 = t_2 \Rightarrow t_1 = t_3}{t_1 = t_2, t_2 = t_3 \Rightarrow t_1 = t_3} \text{XL}$$

Na derivação à esquerda, a fórmula $x = t_1$ é o nosso $\varphi(x)$. À direita, tomamos $\varphi(x)$ como $t_1 = x$.

Problema 19.9. Dê **derivações** dos seguintes sequentes:

1. $\Rightarrow \forall x \forall y ((x = y \wedge \varphi(x)) \rightarrow \varphi(y))$
2. $\exists x \varphi(x) \wedge \forall y \forall z ((\varphi(y) \wedge \varphi(z)) \rightarrow y = z) \Rightarrow \exists x (\varphi(x) \wedge \forall y (\varphi(y) \rightarrow y = x))$

`content/first-order-logic/sequent-calculus/soundness-identity.tex`

19.14 Correção com Predicado de identidade

fol:seq:sid:
sec

Proposição 19.34. **LK** com sequentes iniciais e regras para a identidade é correto.

Prova. Sequentes iniciais da forma $\Rightarrow t = t$ são válidos, pois para toda **estrutura** $\mathfrak{M}$, $\mathfrak{M} \models t = t$. (Note que supomos que o termo t seja fechado, isto é, que não contenha variáveis, de modo que atribuições de variáveis são irrelevantes).

Suponha que a última inferência em **a derivação** seja $=$. Então a premissa é $t_1 = t_2, \Gamma \Rightarrow \Delta, \varphi(t_1)$ e a conclusão é $t_1 = t_2, \Gamma \Rightarrow \Delta, \varphi(t_2)$. Considere **a estrutura** $\mathfrak{M}$. Precisamos mostrar que a conclusão é válida, isto é, se $\mathfrak{M} \models t_1 = t_2$ e $\mathfrak{M} \models \Gamma$, então ou $\mathfrak{M} \models \chi$ para algum $\chi \in \Delta$ ou $\mathfrak{M} \models \varphi(t_2)$.

Pela hipótese de indução, a premissa é válida. Isso significa que, se $\mathfrak{M} \models t_1 = t_2$ e $\mathfrak{M} \models \Gamma$, então ou (a) para algum $\chi \in \Delta$, $\mathfrak{M} \models \chi$, ou (b) $\mathfrak{M} \models \varphi(t_1)$. No caso (a), terminamos. Considere o caso (b). Seja s uma atribuição de variáveis com $s(x) = \text{Val}^{\mathfrak{M}}(t_1)$. Por **Proposição 16.18**, $\mathfrak{M}, s \models \varphi(t_1)$. Como $s \sim_x s$, por **Proposição 16.23**, $\mathfrak{M}, s \models \varphi(x)$. Como $\mathfrak{M} \models t_1 = t_2$, temos $\text{Val}^{\mathfrak{M}}(t_1) = \text{Val}^{\mathfrak{M}}(t_2)$, e portanto $s(x) = \text{Val}^{\mathfrak{M}}(t_2)$. Aplicando **Proposição 16.23** novamente, temos também $\mathfrak{M}, s \models \varphi(t_2)$. Por **Proposição 16.18**, $\mathfrak{M} \models \varphi(t_2)$. $\square$

Capítulo 20

Dedução Natural

Este capítulo apresenta um sistema de dedução natural no estilo de Gentzen/Prawitz.

Para incluir ou excluir material relevante à dedução natural como sistema de prova, use a tag “prfND”.

`content/first-order-logic/natural-deduction/rules-and-proofs.tex`

20.1 Regras e Derivações

explanation Sistemas de dedução natural pretendem ser um paralelo próximo do raciocínio informal usado em provas matemáticas (daí serem, de certo modo, “naturais”). Provas em dedução natural começam com hipóteses. Em seguida, aplicam-se regras de inferência. As hipóteses são “*descarregadas*” pelas regras de inferência $\neg$ -Intro, $\rightarrow$ -Intro, $\vee$ -Elim e $\exists$ -Elim, e o rótulo da hipótese *descarregada* é colocado ao lado da inferência para maior clareza. fol:ntd:rul:sec

Definição 20.1 (Hipótese). Uma *hipótese* é qualquer *sentença* na posição mais alta de qualquer ramo.

Derivações em dedução natural são certas árvores de *sentenças*, em que as *sentenças* no topo são hipóteses e, se a *sentença* está abaixo de um, dois ou três outros sequentes, ela deve seguir corretamente por uma regra de inferência. As *sentenças* no topo da inferência são chamadas de *premissas* e a *sentença* abaixo, de *conclusão* da inferência. As regras vêm em pares, uma regra de introdução e uma de eliminação para cada *operador lógico*. Elas introduzem a *operador lógico* na conclusão ou removem a *operador lógico* de uma premissa da regra. Algumas das regras permitem que uma hipótese de certo tipo seja *descarregada*. Para indicar qual hipótese é *descarregada* por qual inferência, também atribuímos rótulos tanto à hipótese quanto à inferência. Isso é indicado escrevendo-se a hipótese como “[φ]ⁿ.”

É costume considerar regras para todos os *operadores lógicos* $\wedge$, $\vee$, $\rightarrow$, $\neg$ e $\perp$, mesmo que alguns deles sejam definidos.

`content/first-order-logic/natural-deduction/propositional-rules.tex`

20.2 Regras Proposicionais

Regras para $\wedge$

fol:ntd:prl:sec

20.2. REGRAS PROPOSICIONAIS

$$\begin{array}{c}
 \frac{\varphi \quad \psi}{\varphi \wedge \psi} \wedge\text{Intro}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{\varphi \wedge \psi}{\varphi} \wedge\text{Elim} \\
 \frac{\varphi \wedge \psi}{\psi} \wedge\text{Elim}
 \end{array}$$

Regras para $\vee$

$$\begin{array}{c}
 \frac{\varphi}{\varphi \vee \psi} \vee\text{Intro} \\
 \frac{\psi}{\varphi \vee \psi} \vee\text{Intro}
 \end{array}
 \qquad
 \begin{array}{c}
 [\varphi]^n \qquad [\psi]^n \\
 \vdots \qquad \vdots \\
 \vdots \qquad \vdots \\
 \vdots \qquad \vdots \\
 n \quad \varphi \vee \psi \quad \chi \qquad \chi \quad \vee\text{Elim} \\
 \chi
 \end{array}$$

Regras para $\rightarrow$

$$\begin{array}{c}
 [\varphi]^n \\
 \vdots \\
 \vdots \\
 \vdots \\
 n \quad \psi \\
 \varphi \rightarrow \psi \quad \rightarrow\text{Intro}
 \end{array}
 \qquad
 \frac{\varphi \rightarrow \psi \quad \varphi}{\psi} \rightarrow\text{Elim}$$

Regras para $\neg$

$$\begin{array}{c}
 [\varphi]^n \\
 \vdots \\
 \vdots \\
 \vdots \\
 n \quad \perp \\
 \neg\varphi \quad \neg\text{Intro}
 \end{array}
 \qquad
 \frac{\neg\varphi \quad \varphi}{\perp} \neg\text{Elim}$$

Regras para $\perp$

$$\frac{\perp}{\varphi} \perp_I
 \qquad
 \begin{array}{c}
 [\neg\varphi]^n \\
 \vdots \\
 \vdots \\
 \vdots \\
 n \quad \perp \\
 \varphi \quad \perp_C
 \end{array}$$

Note que $\neg$ Intro e $\perp_C$ são muito semelhantes: a diferença é que $\neg$ Intro deriva uma **sentença** negada $\neg\varphi$, mas $\perp_C$, uma **sentença** positiva φ .

Sempre que uma regra indica que alguma hipótese pode ser descarregada, nós tomamos isso como uma permissão, mas não uma exigência. Por exemplo, na regra $\rightarrow$ Intro, podemos descarregar qualquer número de hipóteses da forma φ na **derivação** da premissa ψ , inclusive zero.

`content/first-order-logic/natural-deduction/quantifier-rules.tex`

20.3 Regras de Quantificadores

Regras para $\forall$

fol:ntd:qrl:
sec

$$\frac{\varphi(a)}{\forall x \varphi(x)} \forall\text{Intro} \qquad \frac{\forall x \varphi(x)}{\varphi(t)} \forall\text{Elim}$$

Nas regras para $\forall$, t é um termo fechado (um termo que não contém nenhuma variável), e a é **a símbolo de constante** que não ocorre na conclusão $\forall x \varphi(x)$, nem em qualquer hipótese que esteja **não descarregada** na **derivação** que termina com a premissa $\varphi(a)$. Chamamos a de *variável própria* da inferência $\forall$ Intro.¹

Regras para $\exists$

$$\frac{\varphi(t)}{\exists x \varphi(x)} \exists\text{Intro} \qquad \frac{n \quad \frac{\exists x \varphi(x)}{\chi} \quad \begin{array}{c} [\varphi(a)]^n \\ \vdots \\ \chi \end{array}}{\chi} \exists\text{Elim}$$

Novamente, t é um termo fechado, e a é **a símbolo de constante** que não ocorre na premissa $\exists x \varphi(x)$, na conclusão χ , nem em qualquer hipótese que esteja **não descarregada** nas **derivações** que terminam com as duas premissas (além das hipóteses $\varphi(a)$). Chamamos a de *variável própria* da inferência $\exists$ Elim.

A condição de que uma variável própria não ocorra nem nas premissas nem em qualquer hipótese que esteja **não descarregada** nas **derivações** que levam às premissas para a inferência $\forall$ Intro ou $\exists$ Elim é chamada de *condição de variável própria*.

explanation

¹Usamos o termo “variável própria” embora a na regra acima seja uma constante. Isso tem razões históricas.

20.4. DERIVAÇÕES

Lembre-se da convenção de que, quando φ é a fórmula com a variável x livre, indicamos isso escrevendo $\varphi(x)$. No mesmo contexto, $\varphi(t)$ é então uma abreviação de $\varphi[t/x]$. Assim, poderíamos também escrever a regra $\exists$ Intro como:

$$\frac{\varphi[t/x]}{\exists x \varphi} \exists\text{Intro}$$

Note que t pode já ocorrer em φ , por exemplo, φ poderia ser $P(t, x)$. Assim, inferir $\exists x P(t, x)$ de $P(t, t)$ é uma aplicação correta de $\exists$ Intro—pode-se “substituir” uma ou mais, e não necessariamente todas, as ocorrências de t na premissa pela variável x ligada. Contudo, as condições de variável própria em $\forall$ Intro e $\exists$ Elim exigem que a símbolo de constante a não ocorra em φ . Portanto, não se pode inferir corretamente $\forall x P(a, x)$ de $P(a, a)$ usando $\forall$ Intro.

Em $\exists$ Intro e $\forall$ Elim não há restrições, e o termo t pode ser qualquer coisa, de modo que não temos de nos preocupar com nenhuma condição. Por outro lado, nas regras $\exists$ Elim e $\forall$ Intro, a condição de variável própria exige que a símbolo de constante a não ocorra em lugar algum na conclusão nem em uma hipótese não descarregada. A condição é necessária para garantir que o sistema seja correto, isto é, que só deriva sentenças de hipóteses não descarregadas das quais elas se seguem. Sem essa condição, o seguinte seria permitido:

$$\frac{\frac{\exists x \varphi(x)}{\forall x \varphi(x)} \exists\text{Elim} \quad \frac{[\varphi(a)]^1}{\forall x \varphi(x)} * \forall\text{Intro}}{\forall x \varphi(x)} \exists\text{Elim}$$

Contudo, $\exists x \varphi(x) \not\models \forall x \varphi(x)$.

Como as regras de eliminação para quantificadores só permitem substituir variáveis por termos fechados, segue-se que qualquer fórmula que possa ser derivada de um conjunto de sentenças é ela própria a sentença.

content/first-order-logic/natural-deduction/derivations.tex

20.4 Derivações

Já dissemos o que é uma hipótese e demos as regras de inferência. Derivações em dedução natural são geradas indutivamente a partir disso: cada derivação ou é uma hipótese por si só, ou consiste em uma, duas ou três derivações seguidas de uma inferência correta.

Definição 20.2 (Derivação). A derivação de a sentença φ a partir de hipóteses Γ é uma árvore finita de sentenças que satisfaz as seguintes condições:

1. As sentenças no topo da árvore ou estão em Γ ou são descarregadas por uma inferência na árvore.
2. A sentença na base da árvore é φ .

3. Toda *sentença* na árvore exceto a sentença φ na base é uma premissa de uma aplicação correta de uma regra de inferência cuja conclusão está diretamente abaixo daquela *sentença* na árvore.

Dizemos então que φ é a *conclusão* da *derivação* e Γ , suas hipóteses *não descarregadas*.

Se existe a *derivação* de φ a partir de Γ , dizemos que φ é *derivável* a partir de Γ , ou, em símbolos: $\Gamma \vdash \varphi$. Se há a *derivação* de φ em que toda hipótese é *descarregada*, escrevemos $\vdash \varphi$.

Exemplo 20.3. Toda hipótese por si só é a *derivação*. Assim, por exemplo, φ por si só é a *derivação*, e ψ por si só também. Podemos obter uma nova *derivação* a partir delas aplicando, digamos, a regra $\wedge$ Intro,

$$\frac{\varphi \quad \psi}{\varphi \wedge \psi} \wedge \text{Intro}$$

Essas regras devem ser gerais: podemos substituir o φ e ψ nela por quaisquer *sentenças*, por exemplo, por χ e θ . Então a conclusão seria $\chi \wedge \theta$, e assim

$$\frac{\chi \quad \theta}{\chi \wedge \theta} \wedge \text{Intro}$$

é uma *derivação* correta. É claro que também podemos trocar as hipóteses, de modo que θ desempenhe o papel de φ e χ o de ψ . Assim,

$$\frac{\theta \quad \chi}{\theta \wedge \chi} \wedge \text{Intro}$$

também é uma *derivação* correta.

Podemos agora aplicar outra regra, digamos, $\rightarrow$ Intro, que nos permite concluir um condicional e nos permite descarregar qualquer hipótese que seja idêntica ao antecedente desse condicional. Assim, ambas as seguintes seriam *derivações* corretas:

$$\begin{array}{c} \frac{[\chi]^1 \quad \theta}{\chi \wedge \theta} \wedge \text{Intro} \\ 1 \frac{\quad}{\chi \rightarrow (\chi \wedge \theta)} \rightarrow \text{Intro} \end{array} \quad \begin{array}{c} \frac{\chi \quad [\theta]^1}{\chi \wedge \theta} \wedge \text{Intro} \\ 1 \frac{\quad}{\theta \rightarrow (\chi \wedge \theta)} \rightarrow \text{Intro} \end{array}$$

Elas mostram, respectivamente, que $\theta \vdash \chi \rightarrow (\chi \wedge \theta)$ e $\chi \vdash \theta \rightarrow (\chi \wedge \theta)$.

Lembre-se de que descarregar hipóteses é uma permissão, não uma exigência: não temos de descarregar as hipóteses. Em particular, podemos aplicar uma regra mesmo que as hipóteses não estejam presentes na *derivação*. Por exemplo, o seguinte é válido, embora não haja hipótese φ a ser *descarregada*:

$$1 \frac{\psi}{\varphi \rightarrow \psi} \rightarrow \text{Intro}$$

20.5. EXEMPLOS DE DERIVAÇÕES

20.5 Exemplos de Derivações

fol:ntd:pro:
sec

Exemplo 20.4. Vamos dar a derivação da sentença $(\varphi \wedge \psi) \rightarrow \varphi$.

Começamos escrevendo a conclusão desejada na parte inferior da derivação.

$$\frac{}{(\varphi \wedge \psi) \rightarrow \varphi}$$

Em seguida, precisamos descobrir que tipo de inferência poderia resultar em a sentença desta forma. O operador principal da conclusão é $\rightarrow$, então tentaremos chegar à conclusão usando a regra $\rightarrow$ Intro. O melhor é anotar as suposições envolvidas e rotular as regras de inferência conforme avançamos, para que seja fácil ver se todas as suposições foram descarregadas ao final da prova.

$$\begin{array}{c} [\varphi \wedge \psi]^1 \\ \vdots \\ \vdots \\ \varphi \\ 1 \frac{}{(\varphi \wedge \psi) \rightarrow \varphi} \rightarrow \text{Intro} \end{array}$$

Agora precisamos preencher os passos da suposição $\varphi \wedge \psi$ até φ . Como só temos um conectivo com o qual lidar, $\wedge$, devemos usar a regra $\wedge$ elim. Isso nos dá a seguinte prova:

$$\begin{array}{c} \frac{[\varphi \wedge \psi]^1}{\varphi} \wedge \text{Elim} \\ 1 \frac{}{(\varphi \wedge \psi) \rightarrow \varphi} \rightarrow \text{Intro} \end{array}$$

Agora temos uma derivação correta de $(\varphi \wedge \psi) \rightarrow \varphi$.

Exemplo 20.5. Agora vamos dar a derivação de $(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)$.

Começamos escrevendo a conclusão desejada na parte inferior da derivação.

$$\frac{}{(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)}$$

Para encontrar uma regra lógica que possa nos dar esta conclusão, observamos os conectivos lógicos na conclusão: $\neg$, $\vee$ e $\rightarrow$. Por enquanto, só nos interessa a primeira ocorrência de $\rightarrow$, porque é o operador principal da sentença na conclusão, enquanto $\neg$, $\vee$ e a segunda ocorrência de $\rightarrow$ estão dentro do escopo de outro conectivo, então cuidaremos deles depois. Começamos, portanto, com a regra $\rightarrow$ Intro. Uma aplicação correta deve ser assim:

$$\begin{array}{c} [\neg\varphi \vee \psi]^1 \\ \vdots \\ \vdots \\ \varphi \rightarrow \psi \\ 1 \frac{}{(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)} \rightarrow \text{Intro} \end{array}$$

Isso nos deixa com duas possibilidades para continuar. Podemos continuar trabalhando de baixo para cima e procurar outra aplicação da regra $\rightarrow$ Intro, ou podemos trabalhar de cima para baixo e aplicar a regra $\vee$ Elim. Vamos adotar a segunda opção. Usaremos a suposição $\neg\varphi \vee \psi$ como a premissa mais à esquerda de $\vee$ Elim. Para uma aplicação válida de $\vee$ Elim, as outras duas premissas devem ser idênticas à conclusão $\varphi \rightarrow \psi$, mas cada uma pode ser derivada, por sua vez, de outra suposição, a saber, um dos dois disjuntos de $\neg\varphi \vee \psi$. Nossa **derivação** ficará, então, assim:

$$\begin{array}{c}
 \begin{array}{c} [\neg\varphi]^2 \\ \vdots \\ \varphi \rightarrow \psi \end{array} \quad \begin{array}{c} [\psi]^2 \\ \vdots \\ \varphi \rightarrow \psi \end{array} \\
 2 \frac{[\neg\varphi \vee \psi]^1 \quad \varphi \rightarrow \psi \quad \varphi \rightarrow \psi}{\varphi \rightarrow \psi} \vee\text{Elim} \\
 1 \frac{\varphi \rightarrow \psi}{(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)} \rightarrow\text{Intro}
 \end{array}$$

Em cada um dos dois ramos à direita, queremos derivar $\varphi \rightarrow \psi$, o que é melhor feito usando $\rightarrow$ Intro.

$$\begin{array}{c}
 \begin{array}{c} [\neg\varphi]^2, [\varphi]^3 \\ \vdots \\ \psi \end{array} \quad \begin{array}{c} [\psi]^2, [\varphi]^4 \\ \vdots \\ \psi \end{array} \\
 2 \frac{[\neg\varphi \vee \psi]^1 \quad 3 \frac{\varphi \rightarrow \psi}{\varphi \rightarrow \psi} \rightarrow\text{Intro} \quad 4 \frac{\varphi \rightarrow \psi}{\varphi \rightarrow \psi} \rightarrow\text{Intro}}{\varphi \rightarrow \psi} \vee\text{Elim} \\
 1 \frac{\varphi \rightarrow \psi}{(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)} \rightarrow\text{Intro}
 \end{array}$$

Nas duas partes que faltam na **derivação**, precisamos de **derivações** de ψ a partir de $\neg\varphi$ e φ no meio, e a partir de φ e ψ à esquerda. Começemos pelo primeiro caso. $\neg\varphi$ e φ são as duas premissas de $\neg$ Elim:

$$\begin{array}{c}
 \frac{[\neg\varphi]^2 \quad [\varphi]^3}{\perp} \neg\text{Elim} \\
 \vdots \\
 \psi
 \end{array}$$

Usando $\perp_I$, podemos obter ψ como conclusão e completar o ramo.

$$\begin{array}{c}
 \begin{array}{c} [\neg\varphi]^2 \quad [\varphi]^3 \\ \vdots \\ \perp \end{array} \quad \begin{array}{c} [\psi]^2, [\varphi]^4 \\ \vdots \\ \psi \end{array} \\
 \frac{\perp}{\psi} \perp_I \quad \frac{\psi}{\varphi \rightarrow \psi} \rightarrow\text{Intro} \quad 4 \frac{\varphi \rightarrow \psi}{\varphi \rightarrow \psi} \rightarrow\text{Intro} \\
 2 \frac{[\neg\varphi \vee \psi]^1 \quad 3 \frac{\varphi \rightarrow \psi}{\varphi \rightarrow \psi} \rightarrow\text{Intro} \quad \varphi \rightarrow \psi}{\varphi \rightarrow \psi} \vee\text{Elim} \\
 1 \frac{\varphi \rightarrow \psi}{(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)} \rightarrow\text{Intro}
 \end{array}$$

20.5. EXEMPLOS DE DERIVAÇÕES

Vejam agora o ramo mais à direita. Aqui é importante perceber que a definição de **derivação** permite que hipóteses sejam descarregadas mas não exige que o sejam. Em outras palavras, se pudermos derivar ψ a partir de uma das suposições φ e ψ sem usar a outra, tudo bem. E para derivar ψ a partir de ψ é trivial: ψ por si só é tal **a derivação**, e nenhuma inferência é necessária. Portanto, podemos simplesmente eliminar a suposição φ .

$$\begin{array}{c}
 \frac{[\neg\varphi]^2 \quad [\varphi]^3}{\frac{\perp}{\psi} \perp_I} \neg\text{Elim} \\
 \frac{2 \quad \frac{[\neg\varphi \vee \psi]^1}{\varphi \rightarrow \psi} \rightarrow\text{Intro} \quad \frac{[\psi]^2}{\varphi \rightarrow \psi} \rightarrow\text{Intro}}{\varphi \rightarrow \psi} \vee\text{Elim} \\
 1 \quad \frac{\varphi \rightarrow \psi}{(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)} \rightarrow\text{Intro}
 \end{array}$$

Observe que, na **derivação** concluída, a inferência $\rightarrow\text{Intro}$ mais à direita na verdade não descarrega nenhuma hipótese.

Exemplo 20.6. Até agora não precisamos da regra $\perp_C$. Ela é especial porque nos permite descarregar uma hipótese que não é uma sub-fórmula da conclusão da regra. Está intimamente relacionada à regra $\perp_I$. Na verdade, a regra $\perp_I$ é um caso especial da regra $\perp_C$ —existe uma lógica chamada “lógica intuicionista” na qual apenas $\perp_I$ é permitida. A regra $\perp_C$ é um último recurso quando nada mais funciona. Por exemplo, suponha que queiramos derivar $\varphi \vee \neg\varphi$. Nossa estratégia usual seria tentar derivar $\varphi \vee \neg\varphi$ usando $\vee\text{Intro}$. Mas isso exigiria derivar φ ou $\neg\varphi$ a partir de nenhuma suposição, e isso não pode ser feito. $\perp_C$ ao resgate!

$$\begin{array}{c}
 [\neg(\varphi \vee \neg\varphi)]^1 \\
 \vdots \\
 \perp \\
 1 \quad \frac{\perp}{\varphi \vee \neg\varphi} \perp_C
 \end{array}$$

Agora procuramos **a derivação** de $\perp$ a partir de $\neg(\varphi \vee \neg\varphi)$. Como $\perp$ é a conclusão de $\neg\text{Elim}$, podemos tentar isso:

$$\begin{array}{c}
 \frac{[\neg(\varphi \vee \neg\varphi)]^1 \quad [\neg(\varphi \vee \neg\varphi)]^1}{\frac{\neg\varphi \quad \varphi}{\perp} \neg\text{Elim}} \\
 1 \quad \frac{\perp}{\varphi \vee \neg\varphi} \perp_C
 \end{array}$$

Nossa estratégia para encontrar **a derivação** de $\neg\varphi$ exige uma aplicação de $\neg\text{Intro}$:

$$\begin{array}{c}
[\neg(\varphi \vee \neg\varphi)]^1, [\varphi]^2 \\
\vdots \\
2 \frac{\perp}{\neg\varphi} \neg\text{Intro} \quad \quad \quad [\neg(\varphi \vee \neg\varphi)]^1 \\
\vdots \\
\varphi \neg\text{Elim} \\
\hline
1 \frac{\perp}{\varphi \vee \neg\varphi} \perp_C
\end{array}$$

Aqui, podemos obter $\perp$ facilmente aplicando $\neg\text{Elim}$ à suposição $\neg(\varphi \vee \neg\varphi)$ e a $\varphi \vee \neg\varphi$, que segue da nossa nova suposição φ por $\vee\text{Intro}$:

$$\begin{array}{c}
[\neg(\varphi \vee \neg\varphi)]^1 \quad \quad \quad \frac{[\varphi]^2}{\varphi \vee \neg\varphi} \vee\text{Intro} \quad \quad \quad [\neg(\varphi \vee \neg\varphi)]^1 \\
\hline
2 \frac{\perp}{\neg\varphi} \neg\text{Intro} \quad \quad \quad \neg\text{Elim} \quad \quad \quad \vdots \\
\varphi \neg\text{Elim} \\
\hline
1 \frac{\perp}{\varphi \vee \neg\varphi} \perp_C
\end{array}$$

No lado direito usamos a mesma estratégia, exceto que obtemos φ por $\perp_C$:

$$\begin{array}{c}
[\neg(\varphi \vee \neg\varphi)]^1 \quad \quad \quad \frac{[\varphi]^2}{\varphi \vee \neg\varphi} \vee\text{Intro} \quad \quad \quad [\neg(\varphi \vee \neg\varphi)]^1 \quad \quad \quad \frac{[\neg\varphi]^3}{\varphi \vee \neg\varphi} \vee\text{Intro} \\
\hline
2 \frac{\perp}{\neg\varphi} \neg\text{Intro} \quad \quad \quad \neg\text{Elim} \quad \quad \quad \frac{\perp}{\varphi} \perp_C \quad \quad \quad \neg\text{Elim} \\
\hline
1 \frac{\perp}{\varphi \vee \neg\varphi} \perp_C
\end{array}$$

Problema 20.1. Dê *derivações* que mostrem o seguinte:

1. $\varphi \wedge (\psi \wedge \chi) \vdash (\varphi \wedge \psi) \wedge \chi$.
2. $\varphi \vee (\psi \vee \chi) \vdash (\varphi \vee \psi) \vee \chi$.
3. $\varphi \rightarrow (\psi \rightarrow \chi) \vdash \psi \rightarrow (\varphi \rightarrow \chi)$.
4. $\varphi \vdash \neg\neg\varphi$.

Problema 20.2. Dê *derivações* que mostrem o seguinte:

1. $(\varphi \vee \psi) \rightarrow \chi \vdash \varphi \rightarrow \chi$.
2. $(\varphi \rightarrow \chi) \wedge (\psi \rightarrow \chi) \vdash (\varphi \vee \psi) \rightarrow \chi$.
3. $\vdash \neg(\varphi \wedge \neg\varphi)$.
4. $\psi \rightarrow \varphi \vdash \neg\varphi \rightarrow \neg\psi$.
5. $\vdash (\varphi \rightarrow \neg\varphi) \rightarrow \neg\varphi$.
6. $\vdash \neg(\varphi \rightarrow \psi) \rightarrow \neg\psi$.

20.6. DERIVAÇÕES COM QUANTIFICADORES

7. $\varphi \rightarrow \chi \vdash \neg(\varphi \wedge \neg\chi)$.
8. $\varphi \wedge \neg\chi \vdash \neg(\varphi \rightarrow \chi)$.
9. $\varphi \vee \psi, \neg\psi \vdash \varphi$.
10. $\neg\varphi \vee \neg\psi \vdash \neg(\varphi \wedge \psi)$.
11. $\vdash (\neg\varphi \wedge \neg\psi) \rightarrow \neg(\varphi \vee \psi)$.
12. $\vdash \neg(\varphi \vee \psi) \rightarrow (\neg\varphi \wedge \neg\psi)$.

Problema 20.3. Dê **derivações** que mostrem o seguinte:

1. $\neg(\varphi \rightarrow \psi) \vdash \varphi$.
2. $\neg(\varphi \wedge \psi) \vdash \neg\varphi \vee \neg\psi$.
3. $\varphi \rightarrow \psi \vdash \neg\varphi \vee \psi$.
4. $\vdash \neg\neg\varphi \rightarrow \varphi$.
5. $\varphi \rightarrow \psi, \neg\varphi \rightarrow \psi \vdash \psi$.
6. $(\varphi \wedge \psi) \rightarrow \chi \vdash (\varphi \rightarrow \chi) \vee (\psi \rightarrow \chi)$.
7. $(\varphi \rightarrow \psi) \rightarrow \varphi \vdash \varphi$.
8. $\vdash (\varphi \rightarrow \psi) \vee (\psi \rightarrow \chi)$.

(Todos exigem a regra $\perp_C$.)

`content/first-order-logic/natural-deduction/proving-things-quant.tex`

20.6 Derivações com Quantificadores

fol:ntd:prq:
sec

Exemplo 20.7. Ao lidar com quantificadores, temos de cuidar para não violar a condição de variável própria, e às vezes isso exige que brinquemos com a ordem de execução de certas inferências. Em geral, ajuda tentar cuidar primeiro das regras sujeitas à condição de variável própria (elas ficarão mais abaixo na prova finalizada).

Vamos ver como daríamos a **derivação** de **fórmula** $\exists x \neg\varphi(x) \rightarrow \neg\forall x \varphi(x)$. Começando como de costume, escrevemos

$$\overline{\exists x \neg\varphi(x) \rightarrow \neg\forall x \varphi(x)}$$

Começamos escrevendo o que seria necessário para justificar esse último passo usando a regra $\rightarrow$ Intro.

$$\begin{array}{c}
[\exists x \neg \varphi(x)]^1 \\
\vdots \\
\neg \forall x \varphi(x) \\
1 \frac{}{\exists x \neg \varphi(x) \rightarrow \neg \forall x \varphi(x)} \rightarrow \text{Intro}
\end{array}$$

Como não há uma regra óbvia para aplicar a $\neg \forall x \varphi(x)$, prosseguiremos montando a **derivação** de modo que possamos usar a regra $\exists$ Elim. Aqui devemos prestar atenção à condição de variável própria e escolher uma constante que não apareça em $\exists x \varphi(x)$ nem em quaisquer suposições das quais ela dependa. (Como, no entanto, nenhum **símbolo de constante** aparece, qualquer escolha servirá.)

$$\begin{array}{c}
[\neg \varphi(a)]^2 \\
\vdots \\
2 \frac{[\exists x \neg \varphi(x)]^1 \quad \neg \forall x \varphi(x)}{\neg \forall x \varphi(x)} \exists \text{Elim} \\
1 \frac{}{\exists x \neg \varphi(x) \rightarrow \neg \forall x \varphi(x)} \rightarrow \text{Intro}
\end{array}$$

Para derivar $\neg \forall x \varphi(x)$, tentaremos usar a regra $\neg$ Intro: isso exige que derive-mos uma contradição, possivelmente usando $\forall x \varphi(x)$ como suposição adicional. É claro que essa contradição pode envolver a suposição $\neg \varphi(a)$, que será descarregada pela inferência $\exists$ Elim. Podemos montá-la da seguinte forma:

$$\begin{array}{c}
[\neg \varphi(a)]^2, [\forall x \varphi(x)]^3 \\
\vdots \\
3 \frac{}{\perp} \neg \text{Intro} \\
2 \frac{[\exists x \neg \varphi(x)]^1 \quad \neg \forall x \varphi(x)}{\neg \forall x \varphi(x)} \exists \text{Elim} \\
1 \frac{}{\exists x \neg \varphi(x) \rightarrow \neg \forall x \varphi(x)} \rightarrow \text{Intro}
\end{array}$$

Parece que estamos perto de obter uma contradição. A regra mais fácil de aplicar é a $\forall$ Elim, que não possui condição de variável própria. Como podemos usar qualquer termo que queiramos para substituir o x universalmente quantificado, faz mais sentido continuar usando a para chegarmos a uma contradição.

$$\begin{array}{c}
\frac{[\forall x \varphi(x)]^3}{\varphi(a)} \forall \text{Elim} \\
\frac{[\neg \varphi(a)]^2 \quad \varphi(a)}{} \neg \text{Elim} \\
3 \frac{}{\perp} \neg \text{Intro} \\
2 \frac{[\exists x \neg \varphi(x)]^1 \quad \neg \forall x \varphi(x)}{\neg \forall x \varphi(x)} \exists \text{Elim} \\
1 \frac{}{\exists x \neg \varphi(x) \rightarrow \neg \forall x \varphi(x)} \rightarrow \text{Intro}
\end{array}$$

20.6. DERIVAÇÕES COM QUANTIFICADORES

É importante, especialmente ao lidar com quantificadores, verificar com cuidado neste ponto que a condição de variável própria não foi violada. Como a única regra que aplicamos sujeita à condição de variável própria foi $\exists\text{Elim}$, e a variável própria a não ocorre em nenhuma suposição da qual dependa, esta é uma **derivação** correta.

Exemplo 20.8. Às vezes podemos derivar **a fórmula** a partir de outras **fórmulas**. Nesses casos, podemos ter suposições não descarregadas. É importante acompanhar nossas suposições, bem como o objetivo final.

Vamos ver como daríamos **a derivação** de **fórmula** $\exists x \chi(x, b)$ a partir das suposições $\exists x (\varphi(x) \wedge \psi(x))$ e $\forall x (\psi(x) \rightarrow \chi(x, b))$. Começando como de costume, escrevemos a conclusão na parte inferior.

$$\overline{\exists x \chi(x, b)}$$

Temos duas premissas com as quais trabalhar. Para usar a primeira, isto é, tentar encontrar **a derivação** de $\exists x \chi(x, b)$ a partir de $\exists x (\varphi(x) \wedge \psi(x))$, usariamos a regra $\exists\text{Elim}$. Como ela possui condição de variável própria, aplicaremos essa regra primeiro. Obtemos o seguinte:

$$\begin{array}{c} [\varphi(a) \wedge \psi(a)]^1 \\ \vdots \\ \exists x (\varphi(x) \wedge \psi(x)) \quad \exists x \chi(x, b) \\ \hline 1 \quad \exists x \chi(x, b) \quad \exists\text{Elim} \end{array}$$

As duas suposições com as quais estamos trabalhando compartilham ψ . Pode ser útil neste ponto aplicar $\wedge\text{Elim}$ para separar $\psi(a)$.

$$\begin{array}{c} [\varphi(a) \wedge \psi(a)]^1 \\ \hline \psi(a) \quad \wedge\text{Elim} \\ \vdots \\ \exists x (\varphi(x) \wedge \psi(x)) \quad \exists x \chi(x, b) \\ \hline 1 \quad \exists x \chi(x, b) \quad \exists\text{Elim} \end{array}$$

A segunda suposição com a qual temos de trabalhar é $\forall x (\psi(x) \rightarrow \chi(x, b))$. Como não há condição de variável própria, podemos instanciar x com o **símbolo de constante** a usando $\forall\text{Elim}$ para obter $\psi(a) \rightarrow \chi(a, b)$. Agora temos tanto $\psi(a) \rightarrow \chi(a, b)$ quanto $\psi(a)$. Nosso próximo passo deve ser uma aplicação direta da regra $\rightarrow\text{Elim}$.

$$\begin{array}{c}
\frac{\frac{\forall x (\psi(x) \rightarrow \chi(x, b))}{\psi(a) \rightarrow \chi(a, b)} \forall\text{Elim} \quad \frac{[\varphi(a) \wedge \psi(a)]^1}{\psi(a)} \wedge\text{Elim}}{\chi(a, b)} \rightarrow\text{Elim} \\
\vdots \\
{}_1 \frac{\frac{\exists x (\varphi(x) \wedge \psi(x))}{\exists x \chi(x, b)} \exists\text{Elim} \quad \exists x \chi(x, b)}{\exists x \chi(x, b)} \exists\text{Elim}
\end{array}$$

Estamos quase lá! Uma aplicação de $\exists\text{Intro}$ e atingimos nosso objetivo.

$$\begin{array}{c}
\frac{\frac{\forall x (\psi(x) \rightarrow \chi(x, b))}{\psi(a) \rightarrow \chi(a, b)} \forall\text{Elim} \quad \frac{[\varphi(a) \wedge \psi(a)]^1}{\psi(a)} \wedge\text{Elim}}{\chi(a, b)} \rightarrow\text{Elim} \\
{}_1 \frac{\frac{\exists x (\varphi(x) \wedge \psi(x))}{\exists x \chi(x, b)} \exists\text{Elim} \quad \frac{\chi(a, b)}{\exists x \chi(x, b)} \exists\text{Intro}}{\exists x \chi(x, b)} \exists\text{Elim}
\end{array}$$

Como garantimos em cada etapa que as condições de variável própria não foram violadas, podemos estar confiantes de que esta é uma **derivação** correta.

Exemplo 20.9. Dê a **derivação** de **fórmula** $\neg\forall x \varphi(x)$ a partir das suposições $\forall x \varphi(x) \rightarrow \exists y \psi(y)$ e $\neg\exists y \psi(y)$. Começando como de costume, escrevemos a **fórmula** alvo na parte inferior.

$$\overline{\neg\forall x \varphi(x)}$$

A última linha da **derivação** é uma negação, então vamos tentar usar $\neg\text{Intro}$. Isso exigirá que descubramos como derivar uma contradição.

$$\begin{array}{c}
[\forall x \varphi(x)]^1 \\
\vdots \\
\vdots \\
{}_1 \frac{\perp}{\neg\forall x \varphi(x)} \neg\text{Intro}
\end{array}$$

Até aqui, tudo bem. Podemos usar $\forall\text{Elim}$, mas não é óbvio se isso nos ajudará a alcançar nosso objetivo. Em vez disso, vamos usar uma de nossas suposições. $\forall x \varphi(x) \rightarrow \exists y \psi(y)$ junto com $\forall x \varphi(x)$ nos permitirá usar a regra $\rightarrow\text{Elim}$.

$$\begin{array}{c}
\frac{\forall x \varphi(x) \rightarrow \exists y \psi(y) \quad [\forall x \varphi(x)]^1}{\exists y \psi(y)} \rightarrow\text{Elim} \\
\vdots \\
{}_1 \frac{\perp}{\neg\forall x \varphi(x)} \neg\text{Intro}
\end{array}$$

20.7. NOÇÕES DA TEORIA DA PROVA

Agora temos uma suposição final com a qual trabalhar, e parece que isso nos ajudará a chegar a uma contradição usando $\neg$ Elim.

$$\frac{\frac{\neg \exists y \psi(y) \quad \frac{\forall x \varphi(x) \rightarrow \exists y \psi(y) \quad [\forall x \varphi(x)]^1}{\exists y \psi(y)} \rightarrow \text{Elim}}{1 \quad \frac{\perp}{\neg \forall x \varphi(x)} \neg \text{Intro}} \neg \text{Elim}$$

Problema 20.4. Dê **derivações** que mostrem o seguinte:

1. $\vdash (\forall x \varphi(x) \wedge \forall y \psi(y)) \rightarrow \forall z (\varphi(z) \wedge \psi(z))$.
2. $\vdash (\exists x \varphi(x) \vee \exists y \psi(y)) \rightarrow \exists z (\varphi(z) \vee \psi(z))$.
3. $\forall x (\varphi(x) \rightarrow \psi) \vdash \exists y \varphi(y) \rightarrow \psi$.
4. $\forall x \neg \varphi(x) \vdash \neg \exists x \varphi(x)$.
5. $\vdash \neg \exists x \varphi(x) \rightarrow \forall x \neg \varphi(x)$.
6. $\vdash \neg \exists x \forall y ((\varphi(x, y) \rightarrow \neg \varphi(y, y)) \wedge (\neg \varphi(y, y) \rightarrow \varphi(x, y)))$.

Problema 20.5. Dê **derivações** que mostrem o seguinte:

1. $\vdash \neg \forall x \varphi(x) \rightarrow \exists x \neg \varphi(x)$.
2. $(\forall x \varphi(x) \rightarrow \psi) \vdash \exists y (\varphi(y) \rightarrow \psi)$.
3. $\vdash \exists x (\varphi(x) \rightarrow \forall y \varphi(y))$.

(Todas exigem a regra $\perp_C$.)

`content/first-order-logic/natural-deduction/proof-theoretic-notions.tex`

20.7 Noções da Teoria da Prova

fol:ntd:ptn:
sec

Esta seção reúne as definições da relação de demonstrabilidade e de consistência para a dedução natural.

Assim como definimos várias noções semânticas importantes (validade, consequência, satisfatibilidade), agora definimos noções da *Teoria da Prova* correspondentes. Elas não são definidas por apelo à satisfação de **sentenças** em **estruturas**, mas por apelo à **derivabilidade** ou **não derivabilidade** de certas **sentenças** a partir de outras. Foi uma descoberta importante que essas noções coincidem. Que coincidem é o conteúdo dos teoremas de *correção* e *completez*.

[explanation](#)

Definição 20.10 (Teoremas). Uma *sentença* φ é um *teorema* se há uma *derivação* de φ em dedução natural na qual todas as hipóteses são *descarregadas*. Escrevemos $\vdash \varphi$ se φ é um teorema e $\nvdash \varphi$ se não o é.

Definição 20.11 (Derivabilidade). A *sentença* φ é *derivável* de um conjunto de *sentenças* Γ , $\Gamma \vdash \varphi$, se há uma *derivação* com conclusão φ e na qual toda hipótese é ou *descarregada* ou está em Γ . Se φ não é *derivável* de Γ , escrevemos $\Gamma \nvdash \varphi$.

Definição 20.12 (Consistência). Um conjunto de *sentenças* Γ é *inconsistente* se, e somente se, $\Gamma \vdash \perp$. Se Γ não é inconsistente, isto é, se $\Gamma \nvdash \perp$, dizemos que é *consistente*.

Proposição 20.13 (Reflexividade). Se $\varphi \in \Gamma$, então $\Gamma \vdash \varphi$.

*fol:ntd:ptn:
prop:reflexivity*

Prova. A hipótese φ por si só é uma *derivação* de φ em que toda hipótese *não descarregada* (isto é, φ) está em Γ . $\square$

Proposição 20.14 (Monotonicidade). Se $\Gamma \subseteq \Delta$ e $\Gamma \vdash \varphi$, então $\Delta \vdash \varphi$.

*fol:ntd:ptn:
prop:monotonicity*

Prova. Qualquer *derivação* de φ a partir de Γ é também uma *derivação* de φ a partir de Δ . $\square$

Proposição 20.15 (Transitividade). Se $\Gamma \vdash \varphi$ e $\{\varphi\} \cup \Delta \vdash \psi$, então $\Gamma \cup \Delta \vdash \psi$.

*fol:ntd:ptn:
prop:transitivity*

Prova. Se $\Gamma \vdash \varphi$, há uma *derivação* δ_0 de φ com todas as hipóteses *não descarregadas* em Γ . Se $\{\varphi\} \cup \Delta \vdash \psi$, então há uma *derivação* δ_1 de ψ com todas as hipóteses *não descarregadas* em $\{\varphi\} \cup \Delta$. Considere agora:

$$\begin{array}{c}
 \Delta, [\varphi]^1 \\
 \vdots \\
 \delta_1 \\
 \vdots \\
 \psi \\
 \hline
 \varphi \rightarrow \psi \quad \rightarrow\text{Intro} \\
 \hline
 \psi \\
 \hline
 \varphi \\
 \vdots \\
 \delta_0 \\
 \hline
 \varphi \\
 \hline
 \psi \quad \rightarrow\text{Elim}
 \end{array}$$

As hipóteses *não descarregadas* estão agora todas entre $\Gamma \cup \Delta$, de modo que isto mostra $\Gamma \cup \Delta \vdash \psi$. $\square$

Quando $\Gamma = \{\varphi_1, \varphi_2, \dots, \varphi_k\}$ é um conjunto finito, podemos usar a notação simplificada $\varphi_1, \varphi_2, \dots, \varphi_k \vdash \psi$ para $\Gamma \vdash \psi$; em particular, $\varphi \vdash \psi$ significa que $\{\varphi\} \vdash \psi$.

Note que se $\Gamma \vdash \varphi$ e $\varphi \vdash \psi$, então $\Gamma \vdash \psi$. Segue também que se $\varphi_1, \dots, \varphi_n \vdash \psi$ e $\Gamma \vdash \varphi_i$ para cada i , então $\Gamma \vdash \psi$.

Proposição 20.16. As seguintes afirmações são equivalentes.

*fol:ntd:ptn:
prop:incons*

20.8. DERIVABILIDADE E CONSISTÊNCIA

1. Γ é inconsistente.
2. $\Gamma \vdash \varphi$ para toda *sentença* φ .
3. $\Gamma \vdash \varphi$ e $\Gamma \vdash \neg\varphi$ para alguma *sentença* φ .

Prova. Exercício. □

Problema 20.6. Prove **Proposição 20.16**

*fol:ntd:ptn:
prop:proves-compact*

Proposição 20.17 (Compacidade).

1. Se $\Gamma \vdash \varphi$, então há um subconjunto finito $\Gamma_0 \subseteq \Gamma$ tal que $\Gamma_0 \vdash \varphi$.
2. Se todo subconjunto finito de Γ é consistente, então Γ é consistente.

Prova. 1. Se $\Gamma \vdash \varphi$, então há uma *derivação* δ de φ a partir de Γ . Seja Γ_0 o conjunto de hipóteses *não descarregadas* de δ . Como toda *derivação* é finita, Γ_0 só pode conter finitamente muitas *sentenças*. Assim, δ é uma *derivação* de φ a partir de um Γ_0 finito $\subseteq \Gamma$.

2. Isto é a contrapositiva de (1) no caso especial $\varphi \equiv \perp$. □

content/first-order-logic/natural-deduction/provability-consistency.tex

20.8 Derivabilidade e Consistência

*fol:ntd:prv:
sec*

Estabeleceremos agora várias propriedades da relação de *derivabilidade*. Elas são independentemente interessantes, mas cada uma desempenhará um papel na prova do teorema da completude.

*fol:ntd:prv:
prop:provability-contr*

Proposição 20.18. Se $\Gamma \vdash \varphi$ e $\Gamma \cup \{\varphi\}$ é inconsistente, então Γ é inconsistente.

Prova. Seja a *derivação* de φ a partir de Γ igual a δ_1 e a *derivação* de $\perp$ a partir de $\Gamma \cup \{\varphi\}$ igual a δ_2 . Podemos então derivar:

$$\begin{array}{c}
 \Gamma, [\varphi]^1 \\
 \vdots \\
 \delta_2 \\
 \vdots \\
 \perp \\
 \hline
 1 \frac{}{\neg\varphi} \neg\text{Intro} \quad \begin{array}{c} \Gamma \\ \vdots \\ \delta_1 \\ \vdots \\ \varphi \end{array} \\
 \hline
 \perp \quad \neg\text{Elim}
 \end{array}$$

Na nova *derivação*, a hipótese φ é *descarregada*, de modo que é uma *derivação* a partir de Γ . □

*fol:ntd:prv:
prop:prov-incons*

Proposição 20.19. $\Gamma \vdash \varphi$ se, e somente se, $\Gamma \cup \{\neg\varphi\}$ é inconsistente.

Prova. Primeiro suponha que $\Gamma \vdash \varphi$, isto é, há uma **derivação** δ_0 de φ a partir de hipóteses **não descarregadas** Γ . Obtemos uma **derivação** de $\perp$ a partir de $\Gamma \cup \{\neg\varphi\}$ da seguinte forma:

$$\frac{\begin{array}{c} \Gamma \\ \vdots \\ \delta_0 \\ \varphi \end{array} \quad \neg\varphi}{\perp} \neg\text{Elim}$$

Agora suponha que $\Gamma \cup \{\neg\varphi\}$ é inconsistente, e seja δ_1 a **derivação** correspondente de $\perp$ a partir de hipóteses **não descarregadas** em $\Gamma \cup \{\neg\varphi\}$. Obtemos uma **derivação** de φ apenas a partir de Γ usando $\perp_C$:

$$\frac{\begin{array}{c} \Gamma, [\neg\varphi]^1 \\ \vdots \\ \delta_1 \\ \perp \end{array}}{\varphi} \perp_C$$

□

Problema 20.7. Prove que $\Gamma \vdash \neg\varphi$ se, e somente se, $\Gamma \cup \{\varphi\}$ é inconsistente.

Proposição 20.20. Se $\Gamma \vdash \varphi$ e $\neg\varphi \in \Gamma$, então Γ é inconsistente.

fol:ntd:prv:
prop:explicit-inc

Prova. Suponha que $\Gamma \vdash \varphi$ e $\neg\varphi \in \Gamma$. Então há uma **derivação** δ de φ a partir de Γ . Considere esta aplicação simples da regra $\neg\text{Elim}$:

$$\frac{\begin{array}{c} \Gamma \\ \vdots \\ \delta \\ \varphi \end{array} \quad \neg\varphi}{\perp} \neg\text{Elim}$$

Como $\neg\varphi \in \Gamma$, todas as hipóteses **não descarregadas** estão em Γ ; isto mostra que $\Gamma \vdash \perp$. □

Proposição 20.21. Se $\Gamma \cup \{\varphi\}$ e $\Gamma \cup \{\neg\varphi\}$ são ambos inconsistentes, então Γ é inconsistente.

fol:ntd:prv:
prop:provability-exhaustive

Prova. Há **derivações** δ_1 e δ_2 de $\perp$ a partir de $\Gamma \cup \{\varphi\}$ e de $\perp$ a partir de $\Gamma \cup \{\neg\varphi\}$, respectivamente. Podemos então derivar

$$\frac{\begin{array}{c} \Gamma, [\neg\varphi]^2 \\ \vdots \\ \delta_2 \\ \perp \end{array} \quad \neg\neg\varphi \quad \begin{array}{c} \Gamma, [\varphi]^1 \\ \vdots \\ \delta_1 \\ \perp \end{array}}{\perp} \neg\text{Intro} \quad \neg\text{Intro} \quad \neg\text{Elim}$$

20.9. DERIVABILIDADE E OS CONECTIVOS PROPOSICIONAIS

Como as hipóteses φ e $\neg\varphi$ são **descarregadas**, isto é uma **derivação** de $\perp$ apenas a partir de Γ . Logo Γ é inconsistente. $\square$

`content/first-order-logic/natural-deduction/provability-propositional.tex`

20.9 Derivabilidade e os Conectivos Proposicionais

fol:ntd:ppr:sec Estabelecemos que a relação de **derivabilidade** $\vdash$ da dedução natural é forte o explanation suficiente para estabelecer alguns fatos básicos envolvendo os conectivos proposicionais, tais como $\varphi \wedge \psi \vdash \varphi$ e $\varphi, \varphi \rightarrow \psi \vdash \psi$ (modus ponens). Esses fatos são necessários para a prova do teorema da completude.

Proposição 20.22.

fol:ntd:ppr:prop:provability-land
fol:ntd:ppr:prop:provability-land-left
fol:ntd:ppr:prop:provability-land-right

1. Tanto $\varphi \wedge \psi \vdash \varphi$ quanto $\varphi \wedge \psi \vdash \psi$.
2. $\varphi, \psi \vdash \varphi \wedge \psi$.

Prova. 1. Podemos derivar ambos:

$$\frac{\varphi \wedge \psi}{\varphi} \wedge\text{Elim} \qquad \frac{\varphi \wedge \psi}{\psi} \wedge\text{Elim}$$

2. Podemos derivar:

$$\frac{\varphi \quad \psi}{\varphi \wedge \psi} \wedge\text{Intro}$$

$\square$

Proposição 20.23.

fol:ntd:ppr:prop:provability-lor

1. $\varphi \vee \psi, \neg\varphi, \neg\psi$ é inconsistente.
2. Tanto $\varphi \vdash \varphi \vee \psi$ quanto $\psi \vdash \varphi \vee \psi$.

Prova. 1. Considere a seguinte **derivação**:

$$\frac{1 \quad \varphi \vee \psi \quad \frac{\neg\varphi \quad [\varphi]^1}{\perp} \neg\text{Elim} \quad \frac{\neg\psi \quad [\psi]^1}{\perp} \neg\text{Elim}}{\perp} \vee\text{Elim}$$

Esta é a **derivação** de $\perp$ a partir das hipóteses não descarregadas $\varphi \vee \psi$, $\neg\varphi$ e $\neg\psi$.

2. Podemos derivar ambos:

$$\frac{\varphi}{\varphi \vee \psi} \vee\text{Intro} \qquad \frac{\psi}{\varphi \vee \psi} \vee\text{Intro}$$

$\square$

Proposição 20.24.

fol:ntd:prr:
prop:provability-lif
fol:ntd:prr:
prop:provability-lif-left
fol:ntd:prr:
prop:provability-lif-right

1. $\varphi, \varphi \rightarrow \psi \vdash \psi$.
2. Tanto $\neg\varphi \vdash \varphi \rightarrow \psi$ quanto $\psi \vdash \varphi \rightarrow \psi$.

Prova. 1. Podemos derivar:

$$\frac{\varphi \rightarrow \psi \quad \varphi}{\psi} \rightarrow\text{Elim}$$

2. Isso é mostrado pelas duas **derivações** seguintes:

$$\begin{array}{c} \frac{\neg\varphi \quad [\varphi]^1}{\perp} \neg\text{Elim} \\ \frac{\perp}{\psi} \perp_I \\ 1 \frac{\psi}{\varphi \rightarrow \psi} \rightarrow\text{Intro} \end{array} \quad \frac{\psi}{\varphi \rightarrow \psi} \rightarrow\text{Intro}$$

Note que $\rightarrow\text{Intro}$ pode, mas não precisa, descarregar a hipótese φ . $\square$

content/first-order-logic/natural-deduction/provability-quantifiers.tex

20.10 Derivabilidade e os Quantificadores

explanation O teorema da completude também requer que as regras de dedução natural produzam os fatos sobre $\vdash$ estabelecidos nesta seção.

fol:ntd:qpr:
sec

Teorema 20.25. *Se c é uma constante que não ocorre em Γ ou $\varphi(x)$ e $\Gamma \vdash \varphi(c)$, então $\Gamma \vdash \forall x \varphi(x)$.*

fol:ntd:qpr:
thm:strong-generalization

Prova. Seja δ uma **derivação** de $\varphi(c)$ a partir de Γ . Adicionando uma inferência $\forall\text{Intro}$, obtemos uma **derivação** de $\forall x \varphi(x)$. Como c não ocorre em Γ ou $\varphi(x)$, a condição de variável própria é satisfeita. $\square$

Proposição 20.26.

fol:ntd:qpr:
prop:provability-quantifiers

1. $\varphi(t) \vdash \exists x \varphi(x)$.
2. $\forall x \varphi(x) \vdash \varphi(t)$.

Prova. 1. A seguinte é uma **derivação** de $\exists x \varphi(x)$ a partir de $\varphi(t)$:

$$\frac{\varphi(t)}{\exists x \varphi(x)} \exists\text{Intro}$$

2. A seguinte é uma **derivação** de $\varphi(t)$ a partir de $\forall x \varphi(x)$:

20.11. CORREÇÃO

$$\frac{\forall x \varphi(x)}{\varphi(t)} \forall\text{Elim}$$

□

content/first-order-logic/natural-deduction/soundness.tex

20.11 Correção

fol:ntd:sou:
sec

Um sistema de **derivação**, como a dedução natural, é *correto* se não pode derivar coisas que de fato não seguem. A correção é assim uma espécie de propriedade de segurança garantida para sistemas de **derivação**. Dependendo de qual propriedade da Teoria da Prova está em questão, gostaríamos de saber, por exemplo, que

explanation

1. toda **sentença derivável** é válida;
2. se uma **sentença** é **derivável** a partir de outras, ela também é uma consequência delas;
3. se um conjunto de **sentenças** é inconsistente, ele é insatisfável.

Estas são propriedades importantes de um sistema de **derivação**. Se alguma delas não vale, o sistema de **derivação** é deficiente—ele **deriva** demais. Consequentemente, estabelecer a correção de um sistema de **derivação** é da maior importância.

fol:ntd:sou:
thm:soundness

Teorema 20.27 (Correção). *Se φ é **derivável** a partir das hipóteses **não descarregadas** Γ , então $\Gamma \models \varphi$.*

Prova. Seja δ uma **derivação** de φ . Procedemos por indução no número de inferências em δ .

Para a base da indução, mostramos a afirmação quando o número de inferências é 0. Neste caso, δ consiste apenas de uma única **sentença** φ , isto é, uma hipótese. Essa hipótese é **não descarregada**, pois hipóteses só podem ser **descarregadas** por inferências, e não há inferências. Assim, qualquer **estrutura** $\mathfrak{M}$ que satisfaz todas as hipóteses **não descarregadas** da prova também satisfaz φ .

Agora o passo indutivo. Suponha que δ contém n inferências. A(s) premissa(s) da inferência mais baixa são **derivadas** usando sub-**derivações**, cada uma das quais contém menos de n inferências. Assumimos a hipótese de indução: as premissas da inferência mais baixa seguem das hipóteses **não descarregadas** das sub-**derivações** que terminam nessas premissas. Temos que mostrar que a conclusão φ segue das hipóteses **não descarregadas** de toda a prova.

Distinguimos casos de acordo com o tipo da inferência mais baixa. Primeiro, consideramos as possíveis inferências com apenas uma premissa.

1. Suponha que a última inferência é $\neg$ -Intro: a **derivação** tem a forma

$$\begin{array}{c}
\Gamma, [\varphi]^n \\
\vdots \\
\vdots \delta_1 \\
\vdots \\
\perp \\
\hline
n \frac{}{\neg\varphi} \neg\text{Intro}
\end{array}$$

Pela hipótese de indução, $\perp$ segue das hipóteses **não descarregadas** $\Gamma \cup \{\varphi\}$ de δ_1 . Considere **a estrutura** $\mathfrak{M}$. Precisamos mostrar que, se $\mathfrak{M} \models \Gamma$, então $\mathfrak{M} \models \neg\varphi$. Suponha por redução ao absurdo que $\mathfrak{M} \models \Gamma$, mas $\mathfrak{M} \not\models \neg\varphi$, isto é, $\mathfrak{M} \models \varphi$. Isto significaria que $\mathfrak{M} \models \Gamma \cup \{\varphi\}$. Isto é contrário à nossa hipótese de indução. Logo, $\mathfrak{M} \models \neg\varphi$.

2. A última inferência é $\wedge\text{Elim}$: há duas variantes: φ ou ψ pode ser inferido da premissa $\varphi \wedge \psi$. Considere o primeiro caso. A **derivação** δ tem este aspecto:

$$\begin{array}{c}
\Gamma \\
\vdots \\
\vdots \delta_1 \\
\vdots \\
\varphi \wedge \psi \\
\hline
\varphi \wedge\text{Elim}
\end{array}$$

Pela hipótese de indução, $\varphi \wedge \psi$ segue das hipóteses **não descarregadas** Γ de δ_1 . Considere **a estrutura** $\mathfrak{M}$. Precisamos mostrar que, se $\mathfrak{M} \models \Gamma$, então $\mathfrak{M} \models \varphi$. Suponha $\mathfrak{M} \models \Gamma$. Pela nossa hipótese de indução ($\Gamma \models \varphi \wedge \psi$), sabemos que $\mathfrak{M} \models \varphi \wedge \psi$. Por definição, $\mathfrak{M} \models \varphi \wedge \psi$ se, e somente se, $\mathfrak{M} \models \varphi$ e $\mathfrak{M} \models \psi$. (O caso em que ψ é inferido de $\varphi \wedge \psi$ é tratado de forma similar.)

3. A última inferência é $\vee\text{Intro}$: há duas variantes: $\varphi \vee \psi$ pode ser inferido da premissa φ ou da premissa ψ . Considere o primeiro caso. A **derivação** tem a forma

$$\begin{array}{c}
\Gamma \\
\vdots \\
\vdots \delta_1 \\
\vdots \\
\varphi \\
\hline
\varphi \vee \psi \vee\text{Intro}
\end{array}$$

Pela hipótese de indução, φ segue das hipóteses **não descarregadas** Γ de δ_1 . Considere **a estrutura** $\mathfrak{M}$. Precisamos mostrar que, se $\mathfrak{M} \models \Gamma$, então $\mathfrak{M} \models \varphi \vee \psi$. Suponha $\mathfrak{M} \models \Gamma$; então $\mathfrak{M} \models \varphi$, pois $\Gamma \models \varphi$ (a hipótese de indução). Logo também deve ser o caso que $\mathfrak{M} \models \varphi \vee \psi$. (O caso em que $\varphi \vee \psi$ é inferido de ψ é tratado de forma similar.)

4. A última inferência é $\rightarrow\text{Intro}$: $\varphi \rightarrow \psi$ é inferido de uma subprova com hipótese φ e conclusão ψ , isto é,

20.11. CORREÇÃO

$$\frac{\begin{array}{c} \Gamma, [\varphi]^n \\ \vdots \\ \delta_1 \\ \vdots \\ \psi \end{array}}{\varphi \rightarrow \psi} \rightarrow \text{Intro}$$

Pela hipótese de indução, ψ segue das hipóteses **não descarregadas** de δ_1 , isto é, $\Gamma \cup \{\varphi\} \models \psi$. Considere a estrutura $\mathfrak{M}$. As hipóteses **não descarregadas** de δ são apenas Γ , pois φ é descartada na última inferência. Precisamos mostrar que $\Gamma \models \varphi \rightarrow \psi$. Por redução ao absurdo, suponha que para alguma estrutura $\mathfrak{M}$, $\mathfrak{M} \models \Gamma$ mas $\mathfrak{M} \not\models \varphi \rightarrow \psi$. Assim, $\mathfrak{M} \models \varphi$ e $\mathfrak{M} \not\models \psi$. Mas por hipótese, ψ é uma consequência de $\Gamma \cup \{\varphi\}$, isto é, $\mathfrak{M} \models \psi$, o que é uma contradição. Logo, $\Gamma \models \varphi \rightarrow \psi$.

5. A última inferência é $\perp_I$: aqui, δ termina em

$$\frac{\begin{array}{c} \Gamma \\ \vdots \\ \delta_1 \\ \vdots \\ \perp \end{array}}{\varphi} \perp_I$$

Pela hipótese de indução, $\Gamma \models \perp$. Temos que mostrar que $\Gamma \models \varphi$. Suponha o contrário; então para algum $\mathfrak{M}$ temos $\mathfrak{M} \models \Gamma$ e $\mathfrak{M} \not\models \varphi$. Mas sempre temos $\mathfrak{M} \models \perp$, de modo que isto significaria que $\Gamma \not\models \perp$, contrariando a hipótese de indução.

6. A última inferência é $\perp_C$: Exercício.
 7. A última inferência é $\forall \text{Intro}$: então δ tem a forma

$$\frac{\begin{array}{c} \Gamma \\ \vdots \\ \delta_1 \\ \vdots \\ \varphi(a) \end{array}}{\forall x \varphi(x)} \forall \text{Intro}$$

A premissa $\varphi(a)$ é consequência das hipóteses **não descarregadas** Γ pela hipótese de indução. Considere alguma estrutura $\mathfrak{M}$ tal que $\mathfrak{M} \models \Gamma$. Precisamos mostrar que $\mathfrak{M} \models \forall x \varphi(x)$. Como $\forall x \varphi(x)$ é a **sentença**, isto significa que temos que mostrar que para toda atribuição de variáveis s , $\mathfrak{M}, s \models \varphi(x)$ (**Proposição 16.19**). Como Γ consiste inteiramente em sentenças, $\mathfrak{M}, s \models \psi$ para todo $\psi \in \Gamma$ por **Definição 16.11**. Seja $\mathfrak{M}'$ como $\mathfrak{M}$, exceto que $a^{\mathfrak{M}'} = s(x)$. Como a não ocorre em Γ , $\mathfrak{M}' \models \Gamma$ por **Corolário 16.21**. Como $\Gamma \models \varphi(a)$, $\mathfrak{M}' \models \varphi(a)$. Como $\varphi(a)$ é a **sentença**, $\mathfrak{M}', s \models \varphi(a)$ por **Proposição 16.18**. $\mathfrak{M}', s \models \varphi(x)$ se, e somente se,

$\mathfrak{M}' \models \varphi(a)$ por **Proposição 16.23** (lembre que $\varphi(a)$ é apenas $\varphi(x)[a/x]$). Assim, $\mathfrak{M}', s \models \varphi(x)$. Como a não ocorre em $\varphi(x)$, por **Proposição 16.20**, $\mathfrak{M}, s \models \varphi(x)$. Mas s foi uma atribuição de variáveis arbitrária, de modo que $\mathfrak{M} \models \forall x \varphi(x)$.

8. A última inferência é $\exists$ Intro: Exercício.

9. A última inferência é $\forall$ Elim: Exercício.

Agora consideremos as possíveis inferências com várias premissas: $\forall$ Elim, $\wedge$ Intro, $\rightarrow$ Elim, e $\exists$ Elim.

1. A última inferência é $\wedge$ Intro. $\varphi \wedge \psi$ é inferido das premissas φ e ψ e δ tem a forma

$$\frac{\begin{array}{c} \Gamma_1 \\ \vdots \\ \delta_1 \\ \vdots \\ \varphi \end{array} \quad \begin{array}{c} \Gamma_2 \\ \vdots \\ \delta_2 \\ \vdots \\ \psi \end{array}}{\varphi \wedge \psi} \wedge \text{Intro}$$

Pela hipótese de indução, φ segue das hipóteses **não descarregadas** Γ_1 de δ_1 e ψ segue das hipóteses **não descarregadas** Γ_2 de δ_2 . As hipóteses **não descarregadas** de δ são $\Gamma_1 \cup \Gamma_2$, de modo que temos que mostrar que $\Gamma_1 \cup \Gamma_2 \models \varphi \wedge \psi$. Considere a **estrutura** $\mathfrak{M}$ com $\mathfrak{M} \models \Gamma_1 \cup \Gamma_2$. Como $\mathfrak{M} \models \Gamma_1$, deve ser o caso que $\mathfrak{M} \models \varphi$, pois $\Gamma_1 \models \varphi$, e como $\mathfrak{M} \models \Gamma_2$, $\mathfrak{M} \models \psi$, pois $\Gamma_2 \models \psi$. Juntas, $\mathfrak{M} \models \varphi \wedge \psi$.

2. A última inferência é $\forall$ Elim: Exercício.

3. A última inferência é $\rightarrow$ Elim. ψ é inferido das premissas $\varphi \rightarrow \psi$ e φ . A **derivação** δ tem este aspecto:

$$\frac{\begin{array}{c} \Gamma_1 \\ \vdots \\ \delta_1 \\ \vdots \\ \varphi \rightarrow \psi \end{array} \quad \begin{array}{c} \Gamma_2 \\ \vdots \\ \delta_2 \\ \vdots \\ \varphi \end{array}}{\psi} \rightarrow \text{Elim}$$

Pela hipótese de indução, $\varphi \rightarrow \psi$ segue das hipóteses **não descarregadas** Γ_1 de δ_1 e φ segue das hipóteses **não descarregadas** Γ_2 de δ_2 . Considere a **estrutura** $\mathfrak{M}$. Precisamos mostrar que, se $\mathfrak{M} \models \Gamma_1 \cup \Gamma_2$, então $\mathfrak{M} \models \psi$. Suponha $\mathfrak{M} \models \Gamma_1 \cup \Gamma_2$. Como $\Gamma_1 \models \varphi \rightarrow \psi$, $\mathfrak{M} \models \varphi \rightarrow \psi$. Como $\Gamma_2 \models \varphi$, temos $\mathfrak{M} \models \varphi$. Isto significa que $\mathfrak{M} \models \psi$ (pois se $\mathfrak{M} \not\models \psi$, como $\mathfrak{M} \models \varphi$, teríamos $\mathfrak{M} \not\models \varphi \rightarrow \psi$, contradizendo $\mathfrak{M} \models \varphi \rightarrow \psi$).

4. A última inferência é $\neg$ Elim: Exercício.

20.12. DERIVAÇÕES COM PREDICADO DE IDENTIDADE

5. A última inferência é $\exists$ Elim: Exercício. $\square$

Problema 20.8. Complete a prova de [Teorema 20.27](#).

*fol:ntd:sou:
cor:weak-soundness*

Corolário 20.28. *Se $\vdash \varphi$, então φ é válida.*

*fol:ntd:sou:
cor:consistency-soundness*

Corolário 20.29. *Se Γ é satisfatível, então Γ é consistente.*

Prova. Provamos a contrapositiva. Suponha que Γ não é consistente. Então $\Gamma \vdash \perp$, isto é, há uma [derivação](#) de $\perp$ a partir de hipóteses [não descarregadas](#) em Γ . Por [Teorema 20.27](#), qualquer [estrutura](#) $\mathfrak{M}$ que satisfaz Γ deve satisfazer $\perp$. Como $\mathfrak{M} \not\models \perp$ para toda [estrutura](#) $\mathfrak{M}$, nenhuma $\mathfrak{M}$ pode satisfazer Γ , isto é, Γ não é satisfatível. $\square$

[content/first-order-logic/natural-deduction/identity.tex](#)

20.12 Derivações com Predicado de identidade

*fol:ntd:ide:
sec*

As [Derivações com predicado de identidade](#) requerem regras de inferência adicionais.

$\frac{}{t = t} =\text{Intro}$	$\frac{t_1 = t_2 \quad \varphi(t_1)}{\varphi(t_2)} =\text{Elim}$ $\frac{t_1 = t_2 \quad \varphi(t_2)}{\varphi(t_1)} =\text{Elim}$

Nas regras acima, t , t_1 e t_2 são termos fechados. A regra =Intro nos permite derivar qualquer afirmação de identidade da forma $t = t$ diretamente, sem hipóteses.

Exemplo 20.30. Se s e t são termos fechados, então $\varphi(s), s = t \vdash \varphi(t)$:

$$\frac{s = t \quad \varphi(s)}{\varphi(t)} =\text{Elim}$$

Isto pode ser familiar como o “princípio da substituíbilidade dos idênticos”, ou Lei de Leibniz.

Problema 20.9. Prove que $=$ é simétrica e transitiva, isto é, dê [derivações](#) de $\forall x \forall y (x = y \rightarrow y = x)$ e $\forall x \forall y \forall z ((x = y \wedge y = z) \rightarrow x = z)$

Exemplo 20.31. Derivamos a **sentença**

$$\forall x \forall y ((\varphi(x) \wedge \varphi(y)) \rightarrow x = y)$$

a partir da **sentença**

$$\exists x \forall y (\varphi(y) \rightarrow y = x)$$

Desenvolvemos a **derivação** de trás para frente:

$$\frac{\frac{\frac{\vdots}{\vdots}{\vdots} \quad \frac{1}{a = b} \rightarrow \text{Intro}}{((\varphi(a) \wedge \varphi(b)) \rightarrow a = b)} \rightarrow \text{Intro} \quad \frac{\vdots}{\forall y ((\varphi(a) \wedge \varphi(y)) \rightarrow a = y)} \forall \text{Intro}}{\forall x \forall y ((\varphi(x) \wedge \varphi(y)) \rightarrow x = y)} \forall \text{Intro}$$

Agora teremos que usar a hipótese principal: como é uma **fórmula** existencial, usamos $\exists$ Elim para derivar a conclusão intermediária $a = b$.

$$\frac{\frac{\frac{\frac{[\forall y (\varphi(y) \rightarrow y = c)]^2}{[\varphi(a) \wedge \varphi(b)]^1} \quad \vdots}{a = b} \quad \frac{2}{\frac{\frac{1}{a = b} \rightarrow \text{Intro}}{((\varphi(a) \wedge \varphi(b)) \rightarrow a = b)} \rightarrow \text{Intro} \quad \frac{\vdots}{\forall y ((\varphi(a) \wedge \varphi(y)) \rightarrow a = y)} \forall \text{Intro}}{\forall x \forall y ((\varphi(x) \wedge \varphi(y)) \rightarrow x = y)} \forall \text{Intro} \quad \frac{\exists x \forall y (\varphi(y) \rightarrow y = x)}{a = b} \exists \text{Elim}}{\vdots}$$

A sub-**derivação** no canto superior direito é completada usando suas hipóteses para mostrar que $a = c$ e $b = c$. Isso requer duas **derivações** separadas. A **derivação** para $a = c$ é a seguinte:

$$\frac{\frac{[\forall y (\varphi(y) \rightarrow y = c)]^2}{\varphi(a) \rightarrow a = c} \forall \text{Elim} \quad \frac{[\varphi(a) \wedge \varphi(b)]^1}{\varphi(a)} \wedge \text{Elim}}{a = c} \rightarrow \text{Elim}$$

De $a = c$ e $b = c$ **derivamos** $a = b$ por $=$ Elim.

Problema 20.10. Dê **derivações** das seguintes **fórmulas**:

1. $\forall x \forall y ((x = y \wedge \varphi(x)) \rightarrow \varphi(y))$
2. $\exists x \varphi(x) \wedge \forall y \forall z ((\varphi(y) \wedge \varphi(z)) \rightarrow y = z) \rightarrow \exists x (\varphi(x) \wedge \forall y (\varphi(y) \rightarrow y = x))$

20.13 Correção com Predicado de identidade

fol:ntd:sid:
sec

Proposição 20.32. *A dedução natural com regras para = é correta.*

Prova. Qualquer fórmula da forma $t = t$ é válida, pois para toda estrutura $\mathfrak{M}$, $\mathfrak{M} \models t = t$. (Note que assumimos que o termo t é fechado, isto é, não contém variáveis, de modo que as atribuições de variáveis são irrelevantes).

Suponha que a última inferência em uma derivação é =Elim, isto é, a derivação tem a seguinte forma:

$$\frac{\begin{array}{c} \Gamma_1 \\ \vdots \\ \delta_1 \\ \vdots \\ t_1 = t_2 \end{array} \quad \begin{array}{c} \Gamma_2 \\ \vdots \\ \delta_2 \\ \vdots \\ \varphi(t_1) \end{array}}{\varphi(t_2)} =\text{Elim}$$

As premissas $t_1 = t_2$ e $\varphi(t_1)$ são derivadas a partir de hipóteses não descarregadas Γ_1 e Γ_2 , respectivamente. Queremos mostrar que $\varphi(t_2)$ segue de $\Gamma_1 \cup \Gamma_2$. Considere uma estrutura $\mathfrak{M}$ com $\mathfrak{M} \models \Gamma_1 \cup \Gamma_2$. Pela hipótese de indução, $\mathfrak{M} \models \varphi(t_1)$ e $\mathfrak{M} \models t_1 = t_2$. Portanto, $\text{Val}^{\mathfrak{M}}(t_1) = \text{Val}^{\mathfrak{M}}(t_2)$. Seja s qualquer atribuição de variáveis e $m = \text{Val}^{\mathfrak{M}}(t_1) = \text{Val}^{\mathfrak{M}}(t_2)$. Por **Proposição 16.23**, $\mathfrak{M}, s \models \varphi(t_1)$ se, e somente se, $\mathfrak{M}, s[m/x] \models \varphi(x)$ se, e somente se, $\mathfrak{M}, s \models \varphi(t_2)$. Como $\mathfrak{M} \models \varphi(t_1)$, temos $\mathfrak{M} \models \varphi(t_2)$. $\square$

Capítulo 21

Tableaux

This chapter presents a signed analytic tableaux system.
To include or exclude material relevant to natural deduction as a proof system, use the “prfTab” tag.

-NoValue-

content/first-order-logic/tableaux/rules-and-proofs.tex

21.1 Rules and Tableaux

A **tableau** is a systematic survey of the possible ways a **sentença** can be true or false in a **estrutura**. The building blocks of a tableau are **fórmulas assinadas**: **sentenças** plus a truth value “sign,” either $\mathbb{T}$ or $\mathbb{F}$. These signed **fórmulas** are arranged in a (downward growing) tree. fol:tab:rul:
sec

Definição 21.1. A **fórmula assinada** is a pair consisting of a truth value and a **sentença**, O.s., either:

$$\mathbb{T}\varphi \text{ or } \mathbb{F}\varphi.$$

Intuitively, we might read $\mathbb{T}\varphi$ as “ φ might be true” and $\mathbb{F}\varphi$ as “ φ might be false” (in some **estrutura**).

Each **fórmula assinada** in the tree is either an *assumption* (which are listed at the very top of the tree), or it is obtained from a **fórmula assinada** above it by one of a number of rules of inference. There are two rules for each possible **operador principal** of the preceding **fórmula**, one for the case where the sign is $\mathbb{T}$, and one for the case where the sign is $\mathbb{F}$. Some rules allow the tree to branch, and some only add **fórmulas assinadas** to the branch. A rule may be (and often must be) applied not to the immediately preceding **fórmula assinada**, but to any **fórmula assinada** in the branch from the root to the place the rule is applied.

A branch is *closed* when it contains both $\mathbb{T}\varphi$ and $\mathbb{F}\varphi$. A closed **tableau** is one where every branch is closed. Under the intuitive interpretation, any branch describes a joint possibility, but $\mathbb{T}\varphi$ and $\mathbb{F}\varphi$ are not jointly possible. In other words, if a branch is closed, the possibility it describes has been ruled out. In particular, that means that a closed **tableau** rules out all possibilities of simultaneously making every assumption of the form $\mathbb{T}\varphi$ true and every assumption of the form $\mathbb{F}\varphi$ false.

A closed **tableau for** φ is a closed **tableau** with root $\mathbb{F}\varphi$. If such a closed **tableau** exists, all possibilities for φ being false have been ruled out; O.s., φ must be true in every **estrutura**.

-NoValue-

content/first-order-logic/tableaux/propositional-rules.tex

21.2 Propositional Rules

Rules for $\neg$

fol:tab:prl:
sec

$\frac{\mathbb{T}\neg\varphi}{\mathbb{F}\varphi} \neg\mathbb{T}$	$\frac{\mathbb{F}\neg\varphi}{\mathbb{T}\varphi} \neg\mathbb{F}$

21.3. QUANTIFIER RULES

Rules for $\wedge$

$$\frac{\mathbb{T} \varphi \wedge \psi}{\mathbb{T} \varphi} \wedge \mathbb{T} \quad \frac{\mathbb{F} \varphi \wedge \psi}{\mathbb{F} \varphi \mid \mathbb{F} \psi} \wedge \mathbb{F}$$

Rules for $\vee$

$$\frac{\mathbb{T} \varphi \vee \psi}{\mathbb{T} \varphi \mid \mathbb{T} \psi} \vee \mathbb{T} \quad \frac{\mathbb{F} \varphi \vee \psi}{\mathbb{F} \varphi \mid \mathbb{F} \psi} \vee \mathbb{F}$$

Rules for $\rightarrow$

$$\frac{\mathbb{T} \varphi \rightarrow \psi}{\mathbb{F} \varphi \mid \mathbb{T} \psi} \rightarrow \mathbb{T} \quad \frac{\mathbb{F} \varphi \rightarrow \psi}{\mathbb{T} \varphi \mid \mathbb{F} \psi} \rightarrow \mathbb{F}$$

The Cut Rule

$$\frac{\mathbb{T} \varphi \mid \mathbb{F} \varphi}{\text{Cut}}$$

The Cut rule is not applied “to” a previous *fórmula assinada*; rather, it allows every branch in a *tableau* to be split in two, one branch containing $\mathbb{T} \varphi$, the other $\mathbb{F} \varphi$. It is not necessary—any set of *fórmulas assinadas* with a closed *tableau* has one not using Cut—but it allows us to combine *tableaux* in a convenient way.

-NoValue-

`content/first-order-logic/tableaux/quantifier-rules.tex`

21.3 Quantifier Rules

fol:tab:qrl:
sec

Rules for $\forall$

$\frac{\mathbb{T} \forall x \varphi(x)}{\mathbb{T} \varphi(t)} \forall \mathbb{T}$	$\frac{\mathbb{F} \forall x \varphi(x)}{\mathbb{F} \varphi(a)} \forall \mathbb{F}$

In $\forall \mathbb{T}$, t is a closed term (O.s., one without variables). In $\forall \mathbb{F}$, a is a **símbolo de constante** which must not occur anywhere in the branch above $\forall \mathbb{F}$ rule. We call a the *eigenvariable* of the $\forall \mathbb{F}$ inference.¹

Rules for $\exists$

$\frac{\mathbb{T} \exists x \varphi(x)}{\mathbb{T} \varphi(a)} \exists \mathbb{T}$	$\frac{\mathbb{F} \exists x \varphi(x)}{\mathbb{F} \varphi(t)} \exists \mathbb{F}$

Again, t is a closed term, and a is a **símbolo de constante** which does not occur in the branch above the $\exists \mathbb{T}$ rule. We call a the *eigenvariable* of the $\exists \mathbb{T}$ inference.

The condition that an eigenvariable not occur in the branch above the $\forall \mathbb{F}$ or $\exists \mathbb{T}$ inference is called the *eigenvariable condition*.

explanation Recall the convention that when φ is a **fórmula** with the **variável** x free, we indicate this by writing $\varphi(x)$. In the same context, $\varphi(t)$ then is short for $\varphi[t/x]$. So we could also write the $\exists \mathbb{F}$ rule as:

$$\frac{\mathbb{F} \exists x \varphi}{\mathbb{F} \varphi[t/x]} \exists \mathbb{F}$$

Note that t may already occur in φ , p.ex., φ might be $P(t, x)$. Thus, inferring $\mathbb{F} P(t, t)$ from $\mathbb{F} \exists x P(t, x)$ is a correct application of $\exists \mathbb{F}$. However, the eigenvariable conditions in $\forall \mathbb{F}$ and $\exists \mathbb{T}$ require that the **símbolo de constante** a does not occur in φ . So, you cannot correctly infer $\mathbb{F} P(a, a)$ from $\mathbb{F} \forall x P(a, x)$ using $\forall \mathbb{F}$.

explanation In $\forall \mathbb{T}$ and $\exists \mathbb{F}$ there are no restrictions on the term t . On the other hand, in the $\exists \mathbb{T}$ and $\forall \mathbb{F}$ rules, the eigenvariable condition requires that the **símbolo de constante** a does not occur anywhere in the branches above the respective inference. It is necessary to ensure that the system is sound. Without this condition, the following would be a closed **tableau** for $\exists x \varphi(x) \rightarrow \forall x \varphi(x)$:

1.	$\mathbb{F} \exists x \varphi(x) \rightarrow \forall x \varphi(x)$	Assumption
2.	$\mathbb{T} \exists x \varphi(x)$	$\rightarrow \mathbb{F} 1$
3.	$\mathbb{F} \forall x \varphi(x)$	$\rightarrow \mathbb{F} 1$
4.	$\mathbb{T} \varphi(a)$	$\exists \mathbb{T} 2$
5.	$\mathbb{F} \varphi(a)$	$\forall \mathbb{F} 3$
	$\otimes$	

¹We use the term “eigenvariable” even though a in the above rule is a **símbolo de constante**. This has historical reasons.

21.4. TABLEAUX

However, $\exists x \varphi(x) \rightarrow \forall x \varphi(x)$ is not valid.

-NoValue-

content/first-order-logic/tableaux/derivations.tex

21.4 Tableaux

fol:tab:der: sec We've said what an assumption is, and we've given the rules of inference. explanation **Tableaux** are inductively generated from these: each **tableau** either is a single branch consisting of one or more assumptions, or it results from a **tableau** by applying one of the rules of inference on a branch.

Definição 21.2 (Tableau). A **tableau** for assumptions $S_1\varphi_1, \dots, S_n\varphi_n$ (where each S_i is either $\mathbb{T}$ or $\mathbb{F}$) is a finite tree of **fórmulas assinadas** satisfying the following conditions:

1. The n topmost **fórmulas assinadas** of the tree are $S_i\varphi_i$, one below the other.
2. Every **fórmula assinada** in the tree that is not one of the assumptions results from a correct application of an inference rule to a **fórmula assinada** in the branch above it.

A branch of a **tableau** is *closed* iff it contains both $\mathbb{T}\varphi$ and $\mathbb{F}\varphi$, and *open* otherwise. A **tableau** in which every branch is closed is a *closed tableau* (for its set of assumptions). If a **tableau** is not closed, O.s., if it contains at least one open branch, it is *open*.

Exemplo 21.3. Every set of assumptions on its own is a **tableau**, but it will generally not be closed. (Obviously, it is closed only if the assumptions already contain a pair of **fórmulas assinadas** $\mathbb{T}\varphi$ and $\mathbb{F}\varphi$.)

From a **tableau** (open or closed) we can obtain a new, larger one by applying one of the rules of inference to a **fórmula assinada** φ in it. The rule will append one or more **fórmulas assinadas** to the end of any branch containing the occurrence of φ to which we apply the rule.

For instance, consider the assumption $\mathbb{T}\varphi \wedge \neg\varphi$. Here is the (open) **tableau** consisting of just that assumption:

1. $\mathbb{T}\varphi \wedge \neg\varphi$ Assumption

We obtain a new **tableau** from it by applying the $\wedge\mathbb{T}$ rule to the assumption. That rule allows us to add two new lines to the **tableau**, $\mathbb{T}\varphi$ and $\mathbb{T}\neg\varphi$:

1. $\mathbb{T}\varphi \wedge \neg\varphi$ Assumption
2. $\mathbb{T}\varphi$ $\wedge\mathbb{T} 1$
3. $\mathbb{T}\neg\varphi$ $\wedge\mathbb{T} 1$

When we write down **tableaux**, we record the rules we've applied on the right (p.ex., $\wedge\mathbb{T}1$ means that the **fórmula assinada** on that line is the result of applying the $\wedge\mathbb{T}$ rule to the **fórmula assinada** on line 1). This new **tableau** now contains additional **fórmulas assinadas**, but to only one ($\mathbb{T}\neg\varphi$) can we apply a rule (in this case, the $\neg\mathbb{T}$ rule). This results in the closed **tableau**

1.	$\mathbb{T}\varphi \wedge \neg\varphi$	Assumption
2.	$\mathbb{T}\varphi$	$\wedge\mathbb{T}1$
3.	$\mathbb{T}\neg\varphi$	$\wedge\mathbb{T}1$
4.	$\mathbb{F}\varphi$	$\neg\mathbb{T}3$
	$\otimes$	

-NoValue-

content/first-order-logic/tableaux/proving-things.tex

21.5 Examples of Tableaux

Exemplo 21.4. Let's find a closed **tableau** for the **sentença** $(\varphi \wedge \psi) \rightarrow \varphi$.

fol:tab:pro:
sec

We begin by writing the corresponding assumption at the top of the **tableau**.

1.	$\mathbb{F}(\varphi \wedge \psi) \rightarrow \varphi$	Assumption

There is only one assumption, so only one **fórmula assinada** to which we can apply a rule. (For every **fórmula assinada**, there is always at most one rule that can be applied: it's the rule for the corresponding sign and **operador principal** of the **sentença**.) In this case, this means, we must apply $\rightarrow\mathbb{F}$.

1.	$\mathbb{F}(\varphi \wedge \psi) \rightarrow \varphi \checkmark$	Assumption
2.	$\mathbb{T}\varphi \wedge \psi$	$\rightarrow\mathbb{F}1$
3.	$\mathbb{F}\varphi$	$\rightarrow\mathbb{F}1$

To keep track of which **fórmulas assinadas** we have applied their corresponding rules to, we write a checkmark next to the sentence. However, *only* write a checkmark if the rule has been applied to all open branches. Once a **fórmula assinada** has had the corresponding rule applied in every open branch, we will not have to return to it and apply the rule again. In this case, there is only one branch, so the rule only has to be applied once. (Note that checkmarks are only a convenience for constructing tableaux and are not officially part of the syntax of tableaux.)

There is one new **fórmula assinada** to which we can apply a rule: the $\mathbb{T}\varphi \wedge \psi$ on line 2. Applying the $\wedge\mathbb{T}$ rule results in:

21.5. EXAMPLES OF TABLEAUX

1.	$\mathbb{F}(\varphi \wedge \psi) \rightarrow \varphi \checkmark$	Assumption
2.	$\mathbb{T}\varphi \wedge \psi \checkmark$	$\rightarrow\mathbb{F} 1$
3.	$\mathbb{F}\varphi$	$\rightarrow\mathbb{F} 1$
4.	$\mathbb{T}\varphi$	$\wedge\mathbb{T} 2$
5.	$\mathbb{T}\psi$	$\wedge\mathbb{T} 2$
	$\otimes$	

Since the branch now contains both $\mathbb{T}\varphi$ (on line 4) and $\mathbb{F}\varphi$ (on line 3), the branch is closed. Since it is the only branch, the **tableau** is closed. We have found a closed **tableau** for $(\varphi \wedge \psi) \rightarrow \varphi$.

Exemplo 21.5. Now let's find a closed **tableau** for $(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)$.

We begin with the corresponding assumption:

1.	$\mathbb{F}(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)$	Assumption

The one **fórmula assinada** in this **tableau** has **operador principal** $\rightarrow$ and sign $\mathbb{F}$, so we apply the $\rightarrow\mathbb{F}$ rule to it to obtain:

1.	$\mathbb{F}(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi) \checkmark$	Assumption
2.	$\mathbb{T}\neg\varphi \vee \psi$	$\rightarrow\mathbb{F} 1$
3.	$\mathbb{F}(\varphi \rightarrow \psi)$	$\rightarrow\mathbb{F} 1$

We now have a choice as to whether to apply $\vee\mathbb{T}$ to line 2 or $\rightarrow\mathbb{F}$ to line 3. It actually doesn't matter which order we pick, as long as each **fórmula assinada** has its corresponding rule applied in every branch. So let's pick the first one. The $\vee\mathbb{T}$ rule allows the **tableau** to branch, and the two conclusions of the rule will be the new **fórmulas assinadas** added to the two new branches. This results in:

1.	$\mathbb{F}(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi) \checkmark$	Assumption
2.	$\mathbb{T}\neg\varphi \vee \psi \checkmark$	$\rightarrow\mathbb{F} 1$
3.	$\mathbb{F}(\varphi \rightarrow \psi)$	$\rightarrow\mathbb{F} 1$
4.	$\begin{array}{c} \diagup \quad \diagdown \\ \mathbb{T}\neg\varphi \quad \mathbb{T}\psi \end{array}$	$\vee\mathbb{T} 2$

We have not applied the $\rightarrow\mathbb{F}$ rule to line 3 yet: let's do that now. To save time, we apply it to both branches. Recall that we write a checkmark next to a **fórmula assinada** only if we have applied the corresponding rule in every open branch. So it's a good idea to apply a rule at the end of every branch that contains the **fórmula assinada** the rule applies to. That way we won't have to return to that **fórmula assinada** lower down in the various branches.

1.	$\mathbb{F}(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi) \checkmark$	Assumption
2.	$\mathbb{T}\neg\varphi \vee \psi \checkmark$	$\rightarrow\mathbb{F} 1$
3.	$\mathbb{F}(\varphi \rightarrow \psi) \checkmark$	$\rightarrow\mathbb{F} 1$
$\swarrow \quad \searrow$		
4.	$\mathbb{T}\neg\varphi \quad \mathbb{T}\psi$	$\vee\mathbb{T} 2$
5.	$\mathbb{T}\varphi \quad \mathbb{T}\varphi$	$\rightarrow\mathbb{F} 3$
6.	$\mathbb{F}\psi \quad \mathbb{F}\psi$	$\rightarrow\mathbb{F} 3$
	$\otimes$	

The right branch is now closed. On the left branch, we can still apply the $\neg\mathbb{T}$ rule to line 4. This results in $\mathbb{F}\varphi$ and closes the left branch:

1.	$\mathbb{F}(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi) \checkmark$	Assumption
2.	$\mathbb{T}\neg\varphi \vee \psi \checkmark$	$\rightarrow\mathbb{F} 1$
3.	$\mathbb{F}(\varphi \rightarrow \psi) \checkmark$	$\rightarrow\mathbb{F} 1$
$\swarrow \quad \searrow$		
4.	$\mathbb{T}\neg\varphi \quad \mathbb{T}\psi$	$\vee\mathbb{T} 2$
5.	$\mathbb{T}\varphi \quad \mathbb{T}\varphi$	$\rightarrow\mathbb{F} 3$
6.	$\mathbb{F}\psi \quad \mathbb{F}\psi$	$\rightarrow\mathbb{F} 3$
7.	$\mathbb{F}\varphi \quad \otimes$	$\neg\mathbb{T} 4$
	$\otimes$	

Exemplo 21.6. We can give **tableaux** for any number of **fórmulas assinadas** as assumptions. Often it is also necessary to apply more than one rule that allows branching; and in general a **tableau** can have any number of branches. For instance, consider a **tableau** for $\{\mathbb{T}\varphi \vee (\psi \wedge \chi), \mathbb{F}(\varphi \vee \psi) \wedge (\varphi \vee \chi)\}$. We start by applying the $\vee\mathbb{T}$ to the first assumption:

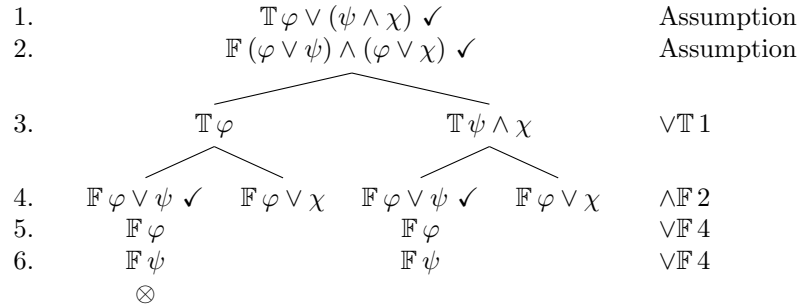
1.	$\mathbb{T}\varphi \vee (\psi \wedge \chi) \checkmark$	Assumption
2.	$\mathbb{F}(\varphi \vee \psi) \wedge (\varphi \vee \chi) \checkmark$	Assumption
$\swarrow \quad \searrow$		
3.	$\mathbb{T}\varphi \quad \mathbb{T}\psi \wedge \chi$	$\vee\mathbb{T} 1$

Now we can apply the $\wedge\mathbb{F}$ rule to line 2. We do this on both branches simultaneously, and can therefore check off line 2:

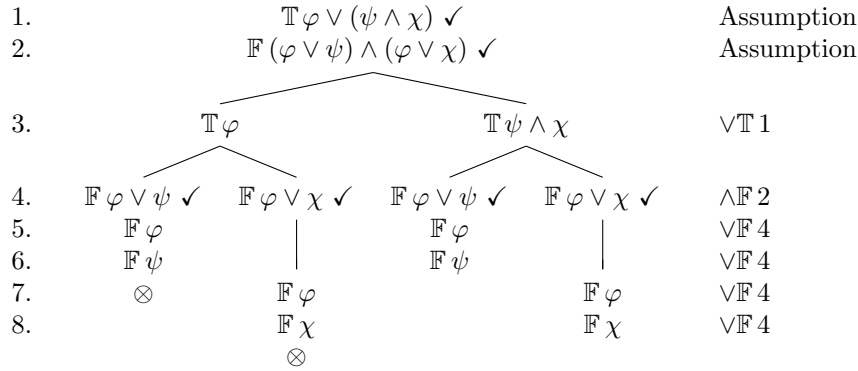
1.	$\mathbb{T}\varphi \vee (\psi \wedge \chi) \checkmark$	Assumption
2.	$\mathbb{F}(\varphi \vee \psi) \wedge (\varphi \vee \chi) \checkmark$	Assumption
$\swarrow \quad \searrow$		
3.	$\mathbb{T}\varphi \quad \mathbb{T}\psi \wedge \chi$	$\vee\mathbb{T} 1$
$\swarrow \quad \searrow \quad \swarrow \quad \searrow$		
4.	$\mathbb{F}\varphi \vee \psi \quad \mathbb{F}\varphi \vee \chi \quad \mathbb{F}\varphi \vee \psi \quad \mathbb{F}\varphi \vee \chi$	$\wedge\mathbb{F} 2$

Now we can apply $\vee\mathbb{F}$ to all the branches containing $\varphi \vee \psi$:

21.5. EXAMPLES OF TABLEAUX

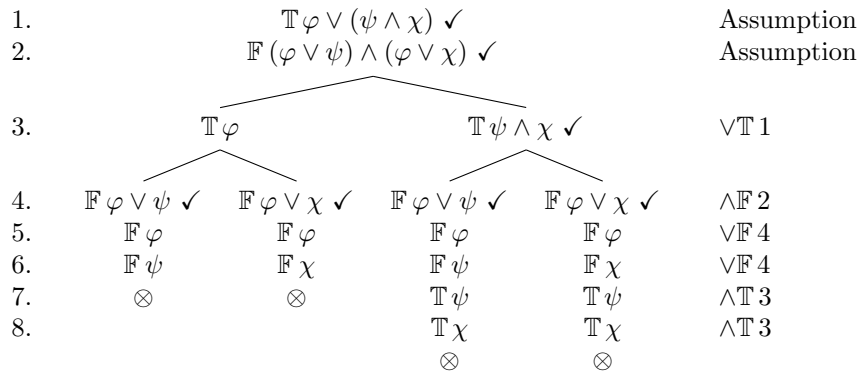


The leftmost branch is now closed. Let's now apply $\vee\mathsf{F}$ to $\varphi \vee \chi$:



Note that we moved the result of applying $\vee\mathsf{F}$ a second time below for clarity. In this instance it would not have been needed, since the justifications would have been the same.

Two branches remain open, and $\mathsf{T}\psi \wedge \chi$ on line 3 remains unchecked. We apply $\wedge\mathsf{T}$ to it to obtain a closed **tableau**:



For comparison, here's a closed **tableau** for the same set of assumptions in which the rules are applied in a different order:

1.	$\mathbb{T} \varphi \vee (\psi \wedge \chi) \checkmark$	Assumption
2.	$\mathbb{F} (\varphi \vee \psi) \wedge (\varphi \vee \chi) \checkmark$	Assumption
		
3.	$\mathbb{F} \varphi \vee \psi \checkmark$	$\wedge \mathbb{F} 2$
4.	$\mathbb{F} \varphi$	$\vee \mathbb{F} 3$
5.	$\mathbb{F} \psi$	$\vee \mathbb{F} 3$
		
6.	$\mathbb{T} \varphi$	$\vee \mathbb{T} 1$
7.	$\otimes$	$\wedge \mathbb{T} 6$
8.	$\otimes$	$\wedge \mathbb{T} 6$

Problema 21.1. Give closed **tableaux** of the following:

1. $\mathbb{T} \varphi \wedge (\psi \wedge \chi), \mathbb{F} (\varphi \wedge \psi) \wedge \chi$.
2. $\mathbb{T} \varphi \vee (\psi \vee \chi), \mathbb{F} (\varphi \vee \psi) \vee \chi$.
3. $\mathbb{T} \varphi \rightarrow (\psi \rightarrow \chi), \mathbb{F} \psi \rightarrow (\varphi \rightarrow \chi)$.
4. $\mathbb{T} \varphi, \mathbb{F} \neg \neg \varphi$.

Problema 21.2. Give closed **tableaux** of the following:

1. $\mathbb{T} (\varphi \vee \psi) \rightarrow \chi, \mathbb{F} \varphi \rightarrow \chi$.
2. $\mathbb{T} (\varphi \rightarrow \chi) \wedge (\psi \rightarrow \chi), \mathbb{F} (\varphi \vee \psi) \rightarrow \chi$.
3. $\mathbb{F} \neg(\varphi \wedge \neg \varphi)$.
4. $\mathbb{T} \psi \rightarrow \varphi, \mathbb{F} \neg \varphi \rightarrow \neg \psi$.
5. $\mathbb{F} (\varphi \rightarrow \neg \varphi) \rightarrow \neg \varphi$.
6. $\mathbb{F} \neg(\varphi \rightarrow \psi) \rightarrow \neg \psi$.
7. $\mathbb{T} \varphi \rightarrow \chi, \mathbb{F} \neg(\varphi \wedge \neg \chi)$.
8. $\mathbb{T} \varphi \wedge \neg \chi, \mathbb{F} \neg(\varphi \rightarrow \chi)$.
9. $\mathbb{T} \varphi \vee \psi, \neg \psi, \mathbb{F} \varphi$.
10. $\mathbb{T} \neg \varphi \vee \neg \psi, \mathbb{F} \neg(\varphi \wedge \psi)$.
11. $\mathbb{F} (\neg \varphi \wedge \neg \psi) \rightarrow \neg(\varphi \vee \psi)$.
12. $\mathbb{F} \neg(\varphi \vee \psi) \rightarrow (\neg \varphi \wedge \neg \psi)$.

Problema 21.3. Give closed **tableaux** of the following:

1. $\mathbb{T} \neg(\varphi \rightarrow \psi), \mathbb{F} \varphi$.

21.6. TABLEAUX WITH QUANTIFIERS

2. $\mathbb{T} \neg(\varphi \wedge \psi), \mathbb{F} \neg\varphi \vee \neg\psi.$
3. $\mathbb{T} \varphi \rightarrow \psi, \mathbb{F} \neg\varphi \vee \psi.$
4. $\mathbb{F} \neg\neg\varphi \rightarrow \varphi.$
5. $\mathbb{T} \varphi \rightarrow \psi, \mathbb{T} \neg\varphi \rightarrow \psi, \mathbb{F} \psi.$
6. $\mathbb{T} (\varphi \wedge \psi) \rightarrow \chi, \mathbb{F} (\varphi \rightarrow \chi) \vee (\psi \rightarrow \chi).$
7. $\mathbb{T} (\varphi \rightarrow \psi) \rightarrow \varphi, \mathbb{F} \varphi.$
8. $\mathbb{F} (\varphi \rightarrow \psi) \vee (\psi \rightarrow \chi).$

-NoValue-

content/first-order-logic/tableaux/proving-things-quant.tex

21.6 Tableaux with Quantifiers

fol:tab:prq:
sec

Exemplo 21.7. When dealing with quantifiers, we have to make sure not to violate the eigenvariable condition, and sometimes this requires us to play around with the order of carrying out certain inferences. In general, it helps to try and take care of rules subject to the eigenvariable condition first (they will be higher up in the finished **tableau**).

Let's see how we'd give a **tableau** for the **sentença** $\exists x \neg\varphi(x) \rightarrow \neg\forall x \varphi(x)$. Starting as usual, we start by recording the assumption,

1. $\mathbb{F} \exists x \neg\varphi(x) \rightarrow \neg\forall x \varphi(x)$ Assumption

Since the **operador principal** is $\rightarrow$, we apply the $\rightarrow\mathbb{F}$:

1. $\mathbb{F} \exists x \neg\varphi(x) \rightarrow \neg\forall x \varphi(x) \checkmark$ Assumption
2. $\mathbb{T} \exists x \neg\varphi(x)$ $\rightarrow\mathbb{F} 1$
3. $\mathbb{F} \neg\forall x \varphi(x)$ $\rightarrow\mathbb{F} 1$

The next line to deal with is 2. We use $\exists\mathbb{T}$. This requires a new **símbolo de constante**; since no **símbolos de constante** yet occur, we can pick any one, say, a .

1. $\mathbb{F} \exists x \neg\varphi(x) \rightarrow \neg\forall x \varphi(x) \checkmark$ Assumption
2. $\mathbb{T} \exists x \neg\varphi(x) \checkmark$ $\rightarrow\mathbb{F} 1$
3. $\mathbb{F} \neg\forall x \varphi(x)$ $\rightarrow\mathbb{F} 1$
4. $\mathbb{T} \neg\varphi(a)$ $\exists\mathbb{T} 2$

Now we apply $\neg\mathbb{F}$ to line 3:

1.	$\mathbb{F} \exists x \neg \varphi(x) \rightarrow \neg \forall x \varphi(x) \checkmark$	Assumption
2.	$\mathbb{T} \exists x \neg \varphi(x) \checkmark$	$\rightarrow \mathbb{F} 1$
3.	$\mathbb{F} \neg \forall x \varphi(x) \checkmark$	$\rightarrow \mathbb{F} 1$
4.	$\mathbb{T} \neg \varphi(a)$	$\exists \mathbb{T} 2$
5.	$\mathbb{T} \forall x \varphi(x)$	$\neg \mathbb{F} 3$

We obtain a closed **tableau** by applying $\neg \mathbb{T}$ to line 4, followed by $\forall \mathbb{T}$ to line 5.

1.	$\mathbb{F} \exists x \neg \varphi(x) \rightarrow \neg \forall x \varphi(x) \checkmark$	Assumption
2.	$\mathbb{T} \exists x \neg \varphi(x) \checkmark$	$\rightarrow \mathbb{F} 1$
3.	$\mathbb{F} \neg \forall x \varphi(x) \checkmark$	$\rightarrow \mathbb{F} 1$
4.	$\mathbb{T} \neg \varphi(a)$	$\exists \mathbb{T} 2$
5.	$\mathbb{T} \forall x \varphi(x)$	$\neg \mathbb{F} 3$
6.	$\mathbb{F} \varphi(a)$	$\neg \mathbb{T} 4$
7.	$\mathbb{T} \varphi(a)$	$\forall \mathbb{T} 5$
	$\otimes$	

Exemplo 21.8. Let's see how we'd give a **tableau** for the set

$$\mathbb{F} \exists x \chi(x, b), \mathbb{T} \exists x (\varphi(x) \wedge \psi(x)), \mathbb{T} \forall x (\psi(x) \rightarrow \chi(x, b)).$$

Starting as usual, we start with the assumptions:

1.	$\mathbb{F} \exists x \chi(x, b)$	Assumption
2.	$\mathbb{T} \exists x (\varphi(x) \wedge \psi(x))$	Assumption
3.	$\mathbb{T} \forall x (\psi(x) \rightarrow \chi(x, b))$	Assumption

We should always apply a rule with the eigenvariable condition first; in this case that would be $\exists \mathbb{T}$ to line 2. Since the assumptions contain the **simbolo de constante** b , we have to use a different one; let's pick a again.

1.	$\mathbb{F} \exists x \chi(x, b)$	Assumption
2.	$\mathbb{T} \exists x (\varphi(x) \wedge \psi(x)) \checkmark$	Assumption
3.	$\mathbb{T} \forall x (\psi(x) \rightarrow \chi(x, b))$	Assumption
4.	$\mathbb{T} \varphi(a) \wedge \psi(a)$	$\exists \mathbb{T} 2$

If we now apply $\exists \mathbb{F}$ to line 1 or $\forall \mathbb{T}$ to line 3, we have to decide which term t to substitute for x . Since there is no eigenvariable condition for these rules, we can pick any term we like. In some cases we may even have to apply the rule several times with different ts . But as a general rule, it pays to pick one of the terms already occurring in the **tableau**—in this case, a and b —and in this case we can guess that a will be more likely to result in a closed branch.

1.	$\mathbb{F} \exists x \chi(x, b)$	Assumption
2.	$\mathbb{T} \exists x (\varphi(x) \wedge \psi(x)) \checkmark$	Assumption
3.	$\mathbb{T} \forall x (\psi(x) \rightarrow \chi(x, b))$	Assumption
4.	$\mathbb{T} \varphi(a) \wedge \psi(a)$	$\exists \mathbb{T} 2$
5.	$\mathbb{F} \chi(a, b)$	$\exists \mathbb{F} 1$
6.	$\mathbb{T} \psi(a) \rightarrow \chi(a, b)$	$\forall \mathbb{T} 3$

21.6. TABLEAUX WITH QUANTIFIERS

We don't check the **fórmulas assinadas** in lines 1 and 3, since we may have to use them again. Now apply $\wedge\mathbb{T}$ to line 4:

1.	$\mathbb{F} \exists x \chi(x, b)$	Assumption
2.	$\mathbb{T} \exists x (\varphi(x) \wedge \psi(x)) \checkmark$	Assumption
3.	$\mathbb{T} \forall x (\psi(x) \rightarrow \chi(x, b))$	Assumption
4.	$\mathbb{T} \varphi(a) \wedge \psi(a) \checkmark$	$\exists\mathbb{T} 2$
5.	$\mathbb{F} \chi(a, b)$	$\exists\mathbb{F} 1$
6.	$\mathbb{T} \psi(a) \rightarrow \chi(a, b)$	$\forall\mathbb{T} 3$
7.	$\mathbb{T} \varphi(a)$	$\wedge\mathbb{T} 4$
8.	$\mathbb{T} \psi(a)$	$\wedge\mathbb{T} 4$

If we now apply $\rightarrow\mathbb{T}$ to line 6, the **tableau** closes:

1.	$\mathbb{F} \exists x \chi(x, b)$	Assumption
2.	$\mathbb{T} \exists x (\varphi(x) \wedge \psi(x)) \checkmark$	Assumption
3.	$\mathbb{T} \forall x (\psi(x) \rightarrow \chi(x, b))$	Assumption
4.	$\mathbb{T} \varphi(a) \wedge \psi(a) \checkmark$	$\exists\mathbb{T} 2$
5.	$\mathbb{F} \chi(a, b)$	$\exists\mathbb{F} 1$
6.	$\mathbb{T} \psi(a) \rightarrow \chi(a, b) \checkmark$	$\forall\mathbb{T} 3$
7.	$\mathbb{T} \varphi(a)$	$\wedge\mathbb{T} 4$
8.	$\mathbb{T} \psi(a)$	$\wedge\mathbb{T} 4$
$\swarrow \quad \searrow$ $\otimes \quad \otimes$		
9.	$\mathbb{F} \psi(a) \quad \mathbb{T} \chi(a, b)$	$\rightarrow\mathbb{T} 6$

Exemplo 21.9. We construct a **tableau** for the set

$$\mathbb{T} \forall x \varphi(x), \mathbb{T} \forall x \varphi(x) \rightarrow \exists y \psi(y), \mathbb{T} \neg \exists y \psi(y).$$

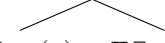
Starting as usual, we write down the assumptions:

1.	$\mathbb{T} \forall x \varphi(x)$	Assumption
2.	$\mathbb{T} \forall x \varphi(x) \rightarrow \exists y \psi(y)$	Assumption
3.	$\mathbb{T} \neg \exists y \psi(y)$	Assumption

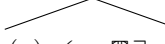
We begin by applying the $\neg\mathbb{T}$ rule to line 3. A corollary to the rule “always apply rules with eigenvariable conditions first” is “defer applying quantifier rules without eigenvariable conditions until needed.” Also, defer rules that result in a split.

1.	$\mathbb{T} \forall x \varphi(x)$	Assumption
2.	$\mathbb{T} \forall x \varphi(x) \rightarrow \exists y \psi(y)$	Assumption
3.	$\mathbb{T} \neg \exists y \psi(y) \checkmark$	Assumption
4.	$\mathbb{F} \exists y \psi(y)$	$\neg\mathbb{T} 3$

The new line 4 requires $\exists\mathbb{F}$, a quantifier rule without the eigenvariable condition. So we defer this in favor of using $\rightarrow\mathbb{T}$ on line 2.

1.	$\mathbb{T}\forall x \varphi(x)$	Assumption
2.	$\mathbb{T}\forall x \varphi(x) \rightarrow \exists y \psi(y) \checkmark$	Assumption
3.	$\mathbb{T}\neg\exists y \psi(y) \checkmark$	Assumption
4.	$\mathbb{F}\exists y \psi(y)$	$\neg\mathbb{T} 3$
		
5.	$\mathbb{F}\forall x \varphi(x) \quad \mathbb{T}\exists y \psi(y)$	$\rightarrow\mathbb{T} 2$

Both new **fórmulas assinadas** require rules with eigenvariable conditions, so these should be next:

1.	$\mathbb{T}\forall x \varphi(x)$	Assumption
2.	$\mathbb{T}\forall x \varphi(x) \rightarrow \exists y \psi(y) \checkmark$	Assumption
3.	$\mathbb{T}\neg\exists y \psi(y) \checkmark$	Assumption
4.	$\mathbb{F}\exists y \psi(y)$	$\neg\mathbb{T} 3$
		
5.	$\mathbb{F}\forall x \varphi(x) \checkmark \quad \mathbb{T}\exists y \psi(y) \checkmark$	$\rightarrow\mathbb{T} 2$
6.	$\mathbb{F}\varphi(b) \quad \mathbb{T}\psi(c)$	$\forall\mathbb{F} 5; \exists\mathbb{T} 5$

To close the branches, we have to use the **fórmulas assinadas** on lines 1 and 3. The corresponding rules ($\forall\mathbb{T}$ and $\exists\mathbb{F}$) don't have eigenvariable conditions, so we are free to pick whichever terms are suitable. In this case, that's b and c , respectively.

1.	$\mathbb{T}\forall x \varphi(x)$	Assumption
2.	$\mathbb{T}\forall x \varphi(x) \rightarrow \exists y \psi(y) \checkmark$	Assumption
3.	$\mathbb{T}\neg\exists y \psi(y) \checkmark$	Assumption
4.	$\mathbb{F}\exists y \psi(y)$	$\neg\mathbb{T} 3$
		
5.	$\mathbb{F}\forall x \varphi(x) \checkmark \quad \mathbb{T}\exists y \psi(y) \checkmark$	$\rightarrow\mathbb{T} 2$
6.	$\mathbb{F}\varphi(b) \quad \mathbb{T}\psi(c)$	$\forall\mathbb{F} 5; \exists\mathbb{T} 5$
7.	$\mathbb{T}\varphi(b) \quad \mathbb{F}\psi(c)$	$\forall\mathbb{T} 1; \exists\mathbb{F} 4$
	$\otimes \quad \otimes$	

Problema 21.4. Give closed **tableaux** of the following:

1. $\mathbb{F}(\forall x \varphi(x) \wedge \forall y \psi(y)) \rightarrow \forall z (\varphi(z) \wedge \psi(z)).$
2. $\mathbb{F}(\exists x \varphi(x) \vee \exists y \psi(y)) \rightarrow \exists z (\varphi(z) \vee \psi(z)).$
3. $\mathbb{T}\forall x (\varphi(x) \rightarrow \psi), \mathbb{F}\exists y \varphi(y) \rightarrow \psi.$
4. $\mathbb{T}\forall x \neg\varphi(x), \mathbb{F}\neg\exists x \varphi(x).$
5. $\mathbb{F}\neg\exists x \varphi(x) \rightarrow \forall x \neg\varphi(x).$

21.7. PROOF-THEORETIC NOTIONS

6. $\mathbb{F} \neg \exists x \forall y ((\varphi(x, y) \rightarrow \neg \varphi(y, y)) \wedge (\neg \varphi(y, y) \rightarrow \varphi(x, y)))$.

Problema 21.5. Give closed **tableaux** of the following:

1. $\mathbb{F} \neg \forall x \varphi(x) \rightarrow \exists x \neg \varphi(x)$.
2. $\mathbb{T} (\forall x \varphi(x) \rightarrow \psi), \mathbb{F} \exists y (\varphi(y) \rightarrow \psi)$.
3. $\mathbb{F} \exists x (\varphi(x) \rightarrow \forall y \varphi(y))$.

-NoValue-

`content/first-order-logic/tableaux/proof-theoretic-notions.tex`

21.7 Proof-Theoretic Notions

fol:tab:ptn:
sec

This section collects the definitions of the provability relation and consistency for tableaux.

Just as we've defined a number of important semantic notions (validity, entailment, satisfiability), we now define corresponding *proof-theoretic notions*. These are not defined by appeal to satisfaction of **sentenças** in **estruturas**, but by appeal to the existence of certain closed **tableaux**. It was an important discovery that these notions coincide. That they do is the content of the *soundness* and *completeness theorems*.

explanation

Definição 21.10 (Theorems). A **sentença** φ is a *theorem* if there is a closed **tableau** for $\mathbb{F} \varphi$. We write $\vdash \varphi$ if φ is a theorem and $\not\vdash \varphi$ if it is not.

Definição 21.11 (Derivabilidade). A **sentença** φ is *derivável* from a set of **sentenças** Γ , $\Gamma \vdash \varphi$ iff there is a finite set $\{\psi_1, \dots, \psi_n\} \subseteq \Gamma$ and a closed **tableau** for the set

$$\{\mathbb{F} \varphi, \mathbb{T} \psi_1, \dots, \mathbb{T} \psi_n\}.$$

If φ is not *derivável* from Γ we write $\Gamma \not\vdash \varphi$.

Definição 21.12 (Consistency). A set of **sentenças** Γ is *inconsistent* iff there is a finite set $\{\psi_1, \dots, \psi_n\} \subseteq \Gamma$ and a closed **tableau** for the set

$$\{\mathbb{T} \psi_1, \dots, \mathbb{T} \psi_n\}.$$

If Γ is not inconsistent, we say it is *consistent*.

fol:tab:ptn:
prop:reflexivity

Proposição 21.13 (Reflexivity). If $\varphi \in \Gamma$, then $\Gamma \vdash \varphi$.

Prova. If $\varphi \in \Gamma$, $\{\varphi\}$ is a finite subset of Γ and the **tableau**

1. $\mathbb{F}\varphi$ Assumption
 2. $\mathbb{T}\varphi$ Assumption
- $$\otimes$$

is closed. □

Proposição 21.14 (Monotonicity). *If $\Gamma \subseteq \Delta$ and $\Gamma \vdash \varphi$, then $\Delta \vdash \varphi$.*

*fol:tab:ptn:
prop:monotonicity*

Prova. Any finite subset of Γ is also a finite subset of Δ . □

Proposição 21.15 (Transitivity). *If $\Gamma \vdash \varphi$ and $\{\varphi\} \cup \Delta \vdash \psi$, then $\Gamma \cup \Delta \vdash \psi$.*

*fol:tab:ptn:
prop:transitivity*

Prova. If $\{\varphi\} \cup \Delta \vdash \psi$, then there is a finite subset $\Delta_0 = \{\chi_1, \dots, \chi_n\} \subseteq \Delta$ such that

$$\{\mathbb{F}\psi, \mathbb{T}\varphi, \mathbb{T}\chi_1, \dots, \mathbb{T}\chi_n\}$$

has a closed **tableau**. If $\Gamma \vdash \varphi$ then there are $\theta_1, \dots, \theta_m$ such that

$$\{\mathbb{F}\varphi, \mathbb{T}\theta_1, \dots, \mathbb{T}\theta_m\}$$

has a closed **tableau**.

Now consider the **tableau** with assumptions

$$\mathbb{F}\psi, \mathbb{T}\chi_1, \dots, \mathbb{T}\chi_n, \mathbb{T}\theta_1, \dots, \mathbb{T}\theta_m.$$

Apply the Cut rule on φ . This generates two branches, one has $\mathbb{T}\varphi$ in it, the other $\mathbb{F}\varphi$. Thus, on the one branch, all of

$$\{\mathbb{F}\psi, \mathbb{T}\varphi, \mathbb{T}\chi_1, \dots, \mathbb{T}\chi_n\}$$

are available. Since there is a closed **tableau** for these assumptions, we can attach it to that branch; every branch through $\mathbb{T}\varphi$ closes. On the other branch, all of

$$\{\mathbb{F}\varphi, \mathbb{T}\theta_1, \dots, \mathbb{T}\theta_m\}$$

are available, so we can also complete the other side to obtain a closed **tableau**. This shows $\Gamma \cup \Delta \vdash \psi$. □

Note that this means that in particular if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. It follows also that if $\varphi_1, \dots, \varphi_n \vdash \psi$ and $\Gamma \vdash \varphi_i$ for each i , then $\Gamma \vdash \psi$.

Proposição 21.16. *Γ is inconsistent iff $\Gamma \vdash \varphi$ for every **sentença** φ .*

*fol:tab:ptn:
prop:incons*

Prova. Exercise. □

Problema 21.6. Prove **Proposição 21.16**

21.8. DERIVABILIDADE AND CONSISTENCY

fol:tab:ptn:
prop:proves-compact

Proposição 21.17 (Compactness).

1. If $\Gamma \vdash \varphi$ then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vdash \varphi$.
2. If every finite subset of Γ is consistent, then Γ is consistent.

Prova. 1. If $\Gamma \vdash \varphi$, then there is a finite subset $\Gamma_0 = \{\psi_1, \dots, \psi_n\}$ and a closed **tableau** for

$$\{\mathbb{F} \varphi, \mathbb{T} \psi_1, \dots, \mathbb{T} \psi_n\}$$

This **tableau** also shows $\Gamma_0 \vdash \varphi$.

2. If Γ is inconsistent, then for some finite subset $\Gamma_0 = \{\psi_1, \dots, \psi_n\}$ there is a closed **tableau** for

$$\{\mathbb{T} \psi_1, \dots, \mathbb{T} \psi_n\}$$

This closed **tableau** shows that Γ_0 is inconsistent. □

-NoValue-

content/first-order-logic/tableaux/provability-consistency.tex

21.8 Derivabilidade and Consistency

fol:tab:prv:
sec

We will now establish a number of properties of the **derivabilidade** relation. They are independently interesting, but each will play a role in the proof of the completeness theorem.

fol:tab:prv:
prop:provability-contr

Proposição 21.18. If $\Gamma \vdash \varphi$ and $\Gamma \cup \{\varphi\}$ is inconsistent, then Γ is inconsistent.

Prova. There are finite $\Gamma_0 = \{\psi_1, \dots, \psi_n\}$ and $\Gamma_1 = \{\chi_1, \dots, \chi_m\} \subseteq \Gamma$ such that

$$\begin{aligned} &\{\mathbb{F} \varphi, \mathbb{T} \psi_1, \dots, \mathbb{T} \psi_n\} \\ &\{\mathbb{T} \varphi, \mathbb{T} \chi_1, \dots, \mathbb{T} \chi_m\} \end{aligned}$$

have closed **tableaux**. Using the Cut rule on φ we can combine these into a single closed **tableau** that shows $\Gamma_0 \cup \Gamma_1$ is inconsistent. Since $\Gamma_0 \subseteq \Gamma$ and $\Gamma_1 \subseteq \Gamma$, $\Gamma_0 \cup \Gamma_1 \subseteq \Gamma$, hence Γ is inconsistent. □

fol:tab:prv:
prop:prov-incons

Proposição 21.19. $\Gamma \vdash \varphi$ iff $\Gamma \cup \{\neg \varphi\}$ is inconsistent.

Prova. First suppose $\Gamma \vdash \varphi$, O.s., there is a closed **tableau** for

$$\{\mathbb{F} \varphi, \mathbb{T} \psi_1, \dots, \mathbb{T} \psi_n\}$$

Using the $\neg \mathbb{T}$ rule, this can be turned into a closed **tableau** for

$$\{\mathbb{T} \neg \varphi, \mathbb{T} \psi_1, \dots, \mathbb{T} \psi_n\}.$$

On the other hand, if there is a closed **tableau** for the latter, we can turn it into a closed **tableau** of the former by removing every formula that results from $\neg\mathbb{T}$ applied to the first assumption $\mathbb{T}\neg\varphi$ as well as that assumption, and adding the assumption $\mathbb{F}\varphi$. For if a branch was closed before because it contained the conclusion of $\neg\mathbb{T}$ applied to $\mathbb{T}\neg\varphi$, O.s., $\mathbb{F}\varphi$, the corresponding branch in the new **tableau** is also closed. If a branch in the old tableau was closed because it contained the assumption $\mathbb{T}\neg\varphi$ as well as $\mathbb{F}\neg\varphi$ we can turn it into a closed branch by applying $\neg\mathbb{F}$ to $\mathbb{F}\neg\varphi$ to obtain $\mathbb{T}\varphi$. This closes the branch since we added $\mathbb{F}\varphi$ as an assumption. $\square$

Problema 21.7. Prove that $\Gamma \vdash \neg\varphi$ iff $\Gamma \cup \{\varphi\}$ is inconsistent.

Proposição 21.20. *If $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$, then Γ is inconsistent.*

*fol:tab:prv:
prop:explicit-inc*

Prova. Suppose $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$. Then there are $\psi_1, \dots, \psi_n \in \Gamma$ such that

$$\{\mathbb{F}\varphi, \mathbb{T}\psi_1, \dots, \mathbb{T}\psi_n\}$$

has a closed tableau. Replace the assumption $\mathbb{F}\varphi$ by $\mathbb{T}\neg\varphi$, and insert the conclusion of $\neg\mathbb{T}$ applied to $\mathbb{F}\varphi$ after the assumptions. Any **sentença** in the **tableau** justified by appeal to line 1 in the old **tableau** is now justified by appeal to line $n+1$. So if the old **tableau** was closed, the new one is. It shows that Γ is inconsistent, since all assumptions are in Γ . $\square$

Proposição 21.21. *If $\Gamma \cup \{\varphi\}$ and $\Gamma \cup \{\neg\varphi\}$ are both inconsistent, then Γ is inconsistent.*

*fol:tab:prv:
prop:provability-exhaustive*

Prova. If there are $\psi_1, \dots, \psi_n \in \Gamma$ and $\chi_1, \dots, \chi_m \in \Gamma$ such that

$$\{\mathbb{T}\varphi, \mathbb{T}\psi_1, \dots, \mathbb{T}\psi_n\} \text{ and } \{\mathbb{T}\neg\varphi, \mathbb{T}\chi_1, \dots, \mathbb{T}\chi_m\}$$

both have closed **tableaux**, we can construct a single, combined **tableau** that shows that Γ is inconsistent by using as assumptions $\mathbb{T}\psi_1, \dots, \mathbb{T}\psi_n$ together with $\mathbb{T}\chi_1, \dots, \mathbb{T}\chi_m$, followed by an application of the Cut rule. This yields two branches, one starting with $\mathbb{T}\varphi$, the other with $\mathbb{F}\varphi$.

On the left side, add the part of the first **tableau** below its assumptions. Here, every rule application is still correct, since each of the assumptions of the first **tableau**, including $\mathbb{T}\varphi$, is available. Thus, every branch below $\mathbb{T}\varphi$ closes.

On the right side, add the part of the second **tableau** below its assumption, with the results of any applications of $\neg\mathbb{T}$ to $\mathbb{T}\neg\varphi$ removed. The conclusion of $\neg\mathbb{T}$ to $\mathbb{T}\neg\varphi$ is $\mathbb{F}\varphi$, which is nevertheless available, as it is the conclusion of the Cut rule on the right side of the combined **tableau**.

If a branch in the second tableau was closed because it contained the assumption $\mathbb{T}\neg\varphi$ (which no longer appears as an assumption in the combined **tableau**) as well as $\mathbb{F}\neg\varphi$, we can applying $\neg\mathbb{F}$ to $\mathbb{F}\neg\varphi$ to obtain $\mathbb{T}\varphi$. Now the corresponding branch in the combined **tableau** also closes, because it contains

21.9. DERIVABILIDADE AND THE PROPOSITIONAL CONNECTIVES

the right-hand conclusion of the Cut rule, $\mathbb{F}\varphi$. If a branch in the second **tableau** closed for any other reason, the corresponding branch in the combined **tableau** also closes, since any **fórmulas assinadas** other than $\mathbb{T}\neg\varphi$ occurring on the branch in the old, second **tableau** also occur on the corresponding branch in the combined **tableau**. $\square$

-NoValue-

content/first-order-logic/tableaux/provability-propositional.tex

21.9 Derivabilidade and the Propositional Connectives

We establish that the **derivabilidade** relation $\vdash$ of tableaux is strong enough to establish some basic facts involving the propositional connectives, such as that $\varphi \wedge \psi \vdash \varphi$ and $\varphi, \varphi \rightarrow \psi \vdash \psi$ (modus ponens). These facts are needed for the proof of the completeness theorem. explanation

Proposição 21.22.

1. Both $\varphi \wedge \psi \vdash \varphi$ and $\varphi \wedge \psi \vdash \psi$.
2. $\varphi, \psi \vdash \varphi \wedge \psi$.

Prova. 1. Both $\{\mathbb{F}\varphi, \mathbb{T}\varphi \wedge \psi\}$ and $\{\mathbb{F}\psi, \mathbb{T}\varphi \wedge \psi\}$ have closed **tableaux**

1.	$\mathbb{F}\varphi$	Assumption
2.	$\mathbb{T}\varphi \wedge \psi$	Assumption
3.	$\mathbb{T}\varphi$	$\wedge\mathbb{T}2$
4.	$\mathbb{T}\psi$	$\wedge\mathbb{T}2$
	$\otimes$	

1.	$\mathbb{F}\psi$	Assumption
2.	$\mathbb{T}\varphi \wedge \psi$	Assumption
3.	$\mathbb{T}\varphi$	$\wedge\mathbb{T}2$
4.	$\mathbb{T}\psi$	$\wedge\mathbb{T}2$
	$\otimes$	

2. Here is a closed **tableau** for $\{\mathbb{T}\varphi, \mathbb{T}\psi, \mathbb{F}\varphi \wedge \psi\}$:

1.	$\mathbb{F}\varphi \wedge \psi$	Assumption
2.	$\mathbb{T}\varphi$	Assumption
3.	$\mathbb{T}\psi$	Assumption
4.	$\begin{array}{c} \diagup \quad \diagdown \\ \mathbb{F}\varphi \quad \mathbb{F}\psi \end{array}$	$\wedge\mathbb{F}1$
	$\otimes \quad \otimes$	

Proposição 21.23.

fol:tab:ppr:
prop:provability-lor

1. $\{\varphi \vee \psi, \neg\varphi, \neg\psi\}$ is inconsistent.
2. Both $\varphi \vdash \varphi \vee \psi$ and $\psi \vdash \varphi \vee \psi$.

Prova. 1. We give a closed **tableau** of $\{\mathbb{T}\varphi \vee \psi, \mathbb{T}\neg\varphi, \mathbb{T}\neg\psi\}$:

1.	$\mathbb{T}\varphi \vee \psi$	Assumption
2.	$\mathbb{T}\neg\varphi$	Assumption
3.	$\mathbb{T}\neg\psi$	Assumption
4.	$\mathbb{F}\varphi$	$\neg\mathbb{T} 2$
5.	$\mathbb{F}\psi$	$\neg\mathbb{T} 3$
$\swarrow \quad \searrow$		
6.	$\mathbb{T}\varphi \quad \mathbb{T}\psi$	$\vee\mathbb{T} 1$
	$\otimes \quad \otimes$	

2. Both $\{\mathbb{F}\varphi \vee \psi, \mathbb{T}\varphi\}$ and $\{\mathbb{F}\varphi \vee \psi, \mathbb{T}\psi\}$ have closed **tableaux**:

1.	$\mathbb{F}\varphi \vee \psi$	Assumption
2.	$\mathbb{T}\varphi$	Assumption
3.	$\mathbb{F}\varphi$	$\vee\mathbb{F} 1$
4.	$\mathbb{F}\psi$	$\vee\mathbb{F} 1$
	$\otimes$	

1.	$\mathbb{F}\varphi \vee \psi$	Assumption
2.	$\mathbb{T}\psi$	Assumption
3.	$\mathbb{F}\varphi$	$\vee\mathbb{F} 1$
4.	$\mathbb{F}\psi$	$\vee\mathbb{F} 1$
	$\otimes$	

Proposição 21.24.

fol:tab:ppr:
prop:provability-lif
fol:tab:ppr:
prop:provability-lif-left
fol:tab:ppr:
prop:provability-lif-right

1. $\varphi, \varphi \rightarrow \psi \vdash \psi$.
2. Both $\neg\varphi \vdash \varphi \rightarrow \psi$ and $\psi \vdash \varphi \rightarrow \psi$.

Prova. 1. $\{\mathbb{F}\psi, \mathbb{T}\varphi \rightarrow \psi, \mathbb{T}\varphi\}$ has a closed **tableau**:

1.	$\mathbb{F}\psi$	Assumption
2.	$\mathbb{T}\varphi \rightarrow \psi$	Assumption
3.	$\mathbb{T}\varphi$	Assumption
$\swarrow \quad \searrow$		
4.	$\mathbb{F}\varphi \quad \mathbb{T}\psi$	$\rightarrow\mathbb{T} 2$
	$\otimes \quad \otimes$	

21.10. DERIVABILIDADE AND THE QUANTIFIERS

2. Both $\{\mathbb{F} \varphi \rightarrow \psi, \mathbb{T} \neg \varphi\}$ and $\{\mathbb{F} \varphi \rightarrow \psi, \mathbb{T} \psi\}$ have closed **tableaux**:

1.	$\mathbb{F} \varphi \rightarrow \psi$	Assumption
2.	$\mathbb{T} \neg \varphi$	Assumption
3.	$\mathbb{T} \varphi$	$\rightarrow \mathbb{F} 1$
4.	$\mathbb{F} \psi$	$\rightarrow \mathbb{F} 1$
5.	$\mathbb{F} \varphi$	$\neg \mathbb{T} 2$
	$\otimes$	

1.	$\mathbb{F} \varphi \rightarrow \psi$	Assumption
2.	$\mathbb{T} \psi$	Assumption
3.	$\mathbb{T} \varphi$	$\rightarrow \mathbb{F} 1$
4.	$\mathbb{F} \psi$	$\rightarrow \mathbb{F} 1$
	$\otimes$	

-NoValue-

[content/first-order-logic/tableaux/provability-quantifiers.tex](#)

21.10 Derivabilidade and the Quantifiers

[fol:tab:qpr:sec](#) The completeness theorem also requires that the tableaux rules yield the facts [explanation](#) about $\vdash$ established in this section.

[fol:tab:qpr:thm:strong-generalization](#) **Teorema 21.25.** *If c is a constant not occurring in Γ or $\varphi(x)$ and $\Gamma \vdash \varphi(c)$, then $\Gamma \vdash \forall x \varphi(x)$.*

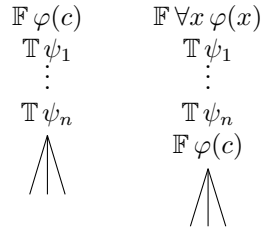
Prova. Suppose $\Gamma \vdash \varphi(c)$, O.s., there are $\psi_1, \dots, \psi_n \in \Gamma$ and a closed **tableau** for

$$\{\mathbb{F} \varphi(c), \mathbb{T} \psi_1, \dots, \mathbb{T} \psi_n\}.$$

We have to show that there is also a closed **tableau** for

$$\{\mathbb{F} \forall x \varphi(x), \mathbb{T} \psi_1, \dots, \mathbb{T} \psi_n\}.$$

Take the closed **tableau** and replace the first assumption with $\mathbb{F} \forall x \varphi(x)$, and insert $\mathbb{F} \varphi(c)$ after the assumptions.



The tableau is still closed, since all **sentenças** available as assumptions before are still available at the top of the **tableau**. The inserted line is the result of a correct application of $\forall\mathbb{F}$, since the **símbolo de constante** c does not occur in $\psi_1, \dots, \psi_n$ or $\forall x \varphi(x)$, O.s., it does not occur above the inserted line in the new **tableau**. $\square$

Proposição 21.26.

fol:tab:qpr:
prop:provability-quantifiers

1. $\varphi(t) \vdash \exists x \varphi(x)$.
2. $\forall x \varphi(x) \vdash \varphi(t)$.

Prova. 1. A closed **tableau** for $\mathbb{F} \exists x \varphi(x), \mathbb{T} \varphi(t)$ is:

1.	$\mathbb{F} \exists x \varphi(x)$	Assumption
2.	$\mathbb{T} \varphi(t)$	Assumption
3.	$\mathbb{F} \varphi(t)$	$\exists\mathbb{F} 1$
	$\otimes$	

2. A closed **tableau** for $\mathbb{F} \varphi(t), \mathbb{T} \forall x \varphi(x)$, is:

1.	$\mathbb{F} \varphi(t)$	Assumption
2.	$\mathbb{T} \forall x \varphi(x)$	Assumption
3.	$\mathbb{T} \varphi(t)$	$\forall\mathbb{T} 2$
	$\otimes$	

-NoValue-

`content/first-order-logic/tableaux/soundness.tex`

21.11 Soundness

explanation A **derivação** system, such as tableaux, is *sound* if it cannot **deriva** things that do not actually hold. Soundness is thus a kind of guaranteed safety property for **derivação** systems. Depending on which proof theoretic property is in question, we would like to know for instance, that *fol:tab:sou:sec*

1. every **derivável** φ is valid;
2. if a **sentença** is **derivável** from some others, it is also a consequence of them;
3. if a set of **sentenças** is inconsistent, it is unsatisfiable.

21.11. SOUNDNESS

These are important properties of a *derivação* system. If any of them do not hold, the *derivação* system is deficient—it would *deriva* too much. Consequently, establishing the soundness of a *derivação* system is of the utmost importance.

Because all these proof-theoretic properties are defined via closed *tableaux* of some kind or other, proving (1)–(3) above requires proving something about the semantic properties of closed *tableaux*. We will first define what it means for a *fórmula assinada* to be satisfied in a structure, and then show that if a *tableau* is closed, no structure satisfies all its assumptions. (1)–(3) then follow as corollaries from this result.

Definição 21.27. A *estrutura* $\mathfrak{M}$ *satisfies* a *fórmula assinada* $\mathbb{T}\varphi$ iff $\mathfrak{M} \models \varphi$, and it satisfies $\mathbb{F}\varphi$ iff $\mathfrak{M} \not\models \varphi$. $\mathfrak{M}$ satisfies a set of *fórmulas assinadas* Γ iff it satisfies every $S\varphi \in \Gamma$. Γ is *satisfiable* if there is a *estrutura* that satisfies it, and *unsatisfiable* otherwise.

fol:tab:sou:

thm:tableau-soundness

Teorema 21.28 (Soundness). *If Γ has a closed *tableau*, Γ is unsatisfiable.*

Prova. Let's call a branch of a *tableau* *satisfiable* iff the set of *fórmulas assinadas* on it is satisfiable, and let's call a *tableau* *satisfiable* if it contains at least one satisfiable branch.

We show the following: Extending a satisfiable *tableau* by one of the rules of inference always results in a satisfiable *tableau*. This will prove the theorem: any closed *tableau* results by applying rules of inference to the *tableau* consisting only of assumptions from Γ . So if Γ were satisfiable, any *tableau* for it would be satisfiable. A closed *tableau*, however, is clearly not satisfiable: every branch contains both $\mathbb{T}\varphi$ and $\mathbb{F}\varphi$, and no structure can both satisfy and not satisfy φ .

Suppose we have a satisfiable *tableau*, O.s., a *tableau* with at least one satisfiable branch. Applying a rule of inference either adds *fórmulas assinadas* to a branch, or splits a branch in two. If the *tableau* has a satisfiable branch which is not extended by the rule application in question, it remains a satisfiable branch in the extended *tableau*, so the extended *tableau* is satisfiable. So we only have to consider the case where a rule is applied to a satisfiable branch.

Let Γ be the set of *fórmulas assinadas* on that branch, and let $S\varphi \in \Gamma$ be the *fórmula assinada* to which the rule is applied. If the rule does not result in a split branch, we have to show that the extended branch, O.s., Γ together with the conclusions of the rule, is still satisfiable. If the rule results in a split branch, we have to show that at least one of the two resulting branches is satisfiable.

First, we consider the possible inferences that do not result in a split branch.

1. The branch is expanded by applying $\neg\mathbb{T}$ to $\mathbb{T}\neg\psi \in \Gamma$. Then the extended branch contains the *fórmulas assinadas* $\Gamma \cup \{\mathbb{F}\psi\}$. Suppose $\mathfrak{M} \models \Gamma$. In particular, $\mathfrak{M} \models \neg\psi$. Thus, $\mathfrak{M} \not\models \psi$, O.s., $\mathfrak{M}$ satisfies $\mathbb{F}\psi$.
2. The branch is expanded by applying $\neg\mathbb{F}$ to $\mathbb{F}\neg\psi \in \Gamma$: Exercise.

3. The branch is expanded by applying $\wedge\mathbb{T}$ to $\mathbb{T}\psi \wedge \chi \in \Gamma$, which results in two new **fórmulas assinadas** on the branch: $\mathbb{T}\psi$ and $\mathbb{T}\chi$. Suppose $\mathfrak{M} \models \Gamma$, in particular $\mathfrak{M} \models \psi \wedge \chi$. Then $\mathfrak{M} \models \psi$ and $\mathfrak{M} \models \chi$. This means that $\mathfrak{M}$ satisfies both $\mathbb{T}\psi$ and $\mathbb{T}\chi$.
4. The branch is expanded by applying $\vee\mathbb{F}$ to $\mathbb{F}\psi \vee \chi \in \Gamma$: Exercise.
5. The branch is expanded by applying $\rightarrow\mathbb{F}$ to $\mathbb{F}\psi \rightarrow \chi \in \Gamma$: This results in two new **fórmulas assinadas** on the branch: $\mathbb{T}\psi$ and $\mathbb{F}\chi$. Suppose $\mathfrak{M} \models \Gamma$, in particular $\mathfrak{M} \models \psi \rightarrow \chi$. Then $\mathfrak{M} \models \psi$ and $\mathfrak{M} \models \chi$. This means that $\mathfrak{M}$ satisfies both $\mathbb{T}\psi$ and $\mathbb{F}\chi$.
6. The branch is expanded by applying $\forall\mathbb{T}$ to $\mathbb{T}\forall x\psi(x) \in \Gamma$: This results in a new **fórmula assinada** $\mathbb{T}\varphi(t)$ on the branch. Suppose $\mathfrak{M} \models \Gamma$, in particular, $\mathfrak{M} \models \forall x\varphi(x)$. By **Proposição 16.31**, $\mathfrak{M} \models \varphi(t)$. Consequently, $\mathfrak{M}$ satisfies $\mathbb{T}\varphi(t)$.
7. The branch is expanded by applying $\forall\mathbb{F}$ to $\mathbb{F}\forall x\psi(x) \in \Gamma$: This results in a new **fórmula assinada** $\mathbb{F}\varphi(a)$ where a is a **símbolo de constante** not occurring in Γ . Since Γ is satisfiable, there is a $\mathfrak{M}$ such that $\mathfrak{M} \models \Gamma$, in particular $\mathfrak{M} \models \forall x\psi(x)$. We have to show that $\Gamma \cup \{\mathbb{F}\varphi(a)\}$ is satisfiable. To do this, we define a suitable $\mathfrak{M}'$ as follows.
 By **Proposição 16.19**, $\mathfrak{M} \models \forall x\psi(x)$ iff for some s , $\mathfrak{M}, s \models \psi(x)$. Now let $\mathfrak{M}'$ be just like $\mathfrak{M}$, except $a^{\mathfrak{M}'} = s(x)$. By **Corolário 16.21**, for any $\mathbb{T}\chi \in \Gamma$, $\mathfrak{M}' \models \chi$, and for any $\mathbb{F}\chi \in \Gamma$, $\mathfrak{M}' \not\models \chi$, since a does not occur in Γ .
 By **Proposição 16.20**, $\mathfrak{M}', s \not\models \varphi(x)$. By **Proposição 16.23**, $\mathfrak{M}', s \not\models \varphi(a)$. Since $\varphi(a)$ is a **sentença**, by **Proposição 16.18**, $\mathfrak{M}' \not\models \varphi(a)$, O.s., $\mathfrak{M}'$ satisfies $\mathbb{F}\varphi(a)$.
8. The branch is expanded by applying $\exists\mathbb{T}$ to $\mathbb{T}\exists x\psi(x) \in \Gamma$: Exercise.
9. The branch is expanded by applying $\exists\mathbb{F}$ to $\mathbb{F}\exists x\psi(x) \in \Gamma$: Exercise.

Now let's consider the possible inferences that result in a split branch.

1. The branch is expanded by applying $\wedge\mathbb{F}$ to $\mathbb{F}\psi \wedge \chi \in \Gamma$, which results in two branches, a left one continuing through $\mathbb{F}\psi$ and a right one through $\mathbb{F}\chi$. Suppose $\mathfrak{M} \models \Gamma$, in particular $\mathfrak{M} \models \psi \wedge \chi$. Then $\mathfrak{M} \models \psi$ or $\mathfrak{M} \models \chi$. In the former case, $\mathfrak{M}$ satisfies $\mathbb{F}\psi$, O.s., $\mathfrak{M}$ satisfies the formulas on the left branch. In the latter, $\mathfrak{M}$ satisfies $\mathbb{F}\chi$, O.s., $\mathfrak{M}$ satisfies the formulas on the right branch.
2. The branch is expanded by applying $\vee\mathbb{T}$ to $\mathbb{T}\psi \vee \chi \in \Gamma$: Exercise.
3. The branch is expanded by applying $\rightarrow\mathbb{T}$ to $\mathbb{T}\psi \rightarrow \chi \in \Gamma$: Exercise.
4. The branch is expanded by Cut: This results in two branches, one containing $\mathbb{T}\psi$, the other containing $\mathbb{F}\psi$. Since $\mathfrak{M} \models \Gamma$ and either $\mathfrak{M} \models \psi$ or $\mathfrak{M} \not\models \psi$, $\mathfrak{M}$ satisfies either the left or the right branch. $\square$

21.12. TABLEAUX WITH PREDICADO DE IDENTIDADE

Problema 21.8. Complete the proof of [Teorema 21.28](#).

fol:tab:sou: cor:weak-soundness **Corolário 21.29.** *If $\vdash \varphi$ then φ is valid.*

fol:tab:sou: cor:entailment-soundness **Corolário 21.30.** *If $\Gamma \vdash \varphi$ then $\Gamma \models \varphi$.*

Prova. If $\Gamma \vdash \varphi$ then for some $\psi_1, \dots, \psi_n \in \Gamma$, $\{\mathbb{F}\varphi, \mathbb{T}\psi_1, \dots, \mathbb{T}\psi_n\}$ has a closed [tableau](#). By [Teorema 21.28](#), every [estrutura](#) $\mathfrak{M}$ either makes some ψ_i false or makes φ true. Hence, if $\mathfrak{M} \models \Gamma$ then also $\mathfrak{M} \models \varphi$. $\square$

fol:tab:sou: cor:consistency-soundness **Corolário 21.31.** *If Γ is satisfiable, then it is consistent.*

Prova. We prove the contrapositive. Suppose that Γ is not consistent. Then there are $\psi_1, \dots, \psi_n \in \Gamma$ and a closed [tableau](#) for $\{\mathbb{T}\psi_1, \dots, \mathbb{T}\psi_n\}$. By [Teorema 21.28](#), there is no $\mathfrak{M}$ such that $\mathfrak{M} \models \psi_i$ for all $i = 1, \dots, n$. But then Γ is not satisfiable. $\square$

-NoValue-

[content/first-order-logic/tableaux/identity.tex](#)

21.12 Tableaux with Predicado de identidade

fol:tab:ide: sec [Tableaux](#) with [predicado de identidade](#) require additional inference rules. The rules for $=$ are (t , t_1 , and t_2 are closed terms):

$\frac{}{\mathbb{T} t = t} =$	$\frac{\mathbb{T} t_1 = t_2 \quad \mathbb{T} \varphi(t_1)}{\mathbb{T} \varphi(t_2)} =\mathbb{T}$	$\frac{\mathbb{T} t_1 = t_2 \quad \mathbb{F} \varphi(t_1)}{\mathbb{F} \varphi(t_2)} =\mathbb{F}$

Note that in contrast to all the other rules, $=\mathbb{T}$ and $=\mathbb{F}$ require that *two* signed [fórmulas](#) already appear on the branch, namely both $\mathbb{T} t_1 = t_2$ and $\mathbb{S} \varphi(t_1)$.

Exemplo 21.32. If s and t are closed terms, then $s = t, \varphi(s) \vdash \varphi(t)$:

1. $\mathbb{F} \varphi(t)$ Assumption
 2. $\mathbb{T} s = t$ Assumption
 3. $\mathbb{T} \varphi(s)$ Assumption
 4. $\mathbb{T} \varphi(t)$ $=\mathbb{T} 2, 3$
- $\otimes$

This may be familiar as the principle of substitutability of identicals, or Leibniz' Law.

[Tableaux](#) prove that $=$ is symmetric, O.s., that $s_1 = s_2 \vdash s_2 = s_1$:

1.	$\mathbb{F} s_2 = s_1$	Assumption
2.	$\mathbb{T} s_1 = s_2$	Assumption
3.	$\mathbb{T} s_1 = s_1$	$=$
4.	$\mathbb{T} s_2 = s_1$	$=\mathbb{T} 2, 3$
	$\otimes$	

Here, line 2 is the first prerequisite *fórmula* $\mathbb{T} s_1 = s_2$ of $=\mathbb{T}$. Line 3 is the second one, of the form $\mathbb{T} \varphi(s_2)$ —think of $\varphi(x)$ as $x = s_1$, then $\varphi(s_1)$ is $s_1 = s_1$ and $\varphi(s_2)$ is $s_2 = s_1$.

They also prove that $=$ is transitive, O.s., that $s_1 = s_2, s_2 = s_3 \vdash s_1 = s_3$:

1.	$\mathbb{F} s_1 = s_3$	Assumption
2.	$\mathbb{T} s_1 = s_2$	Assumption
3.	$\mathbb{T} s_2 = s_3$	Assumption
4.	$\mathbb{T} s_1 = s_3$	$=\mathbb{T} 3, 2$
	$\otimes$	

In this *tableau*, the first prerequisite *fórmula* of $=\mathbb{T}$ is line 3, $\mathbb{T} s_2 = s_3$ (s_2 plays the role of t_1 , and s_3 the role of t_2). The second prerequisite, of the form $\mathbb{T} \varphi(s_2)$ is line 2. Here, think of $\varphi(x)$ as $s_1 = x$; that makes $\varphi(s_2)$ into $t_1 = t_2$ (O.s., line 2) and $\varphi(s_3)$ into the *fórmula* $s_1 = s_3$ in the conclusion.

Problema 21.9. Give closed *tableaux* for the following:

1. $\mathbb{F} \forall x \forall y ((x = y \wedge \varphi(x)) \rightarrow \varphi(y))$
2. $\mathbb{F} \exists x (\varphi(x) \wedge \forall y (\varphi(y) \rightarrow y = x)),$
 $\mathbb{T} \exists x \varphi(x) \wedge \forall y \forall z ((\varphi(y) \wedge \varphi(z)) \rightarrow y = z)$

-NoValue-

`content/first-order-logic/tableaux/soundness-identity.tex`

21.13 Soundness with Predicado de identidade

Proposição 21.33. *Tableaux with rules for identity are sound: no closed *tableau* is satisfiable.*

fol:tab:sid:
sec

Prova. We just have to show as before that if a *tableau* has a satisfiable branch, the branch resulting from applying one of the rules for $=$ to it is also satisfiable. Let Γ be the set of signed *fórmulas* on the branch, and let $\mathfrak{M}$ be a *estrutura* satisfying Γ .

Suppose the branch is expanded using $=$, O.s., by adding the signed *fórmula* $\mathbb{T} t = t$. Trivially, $\mathfrak{M} \models t = t$, so $\mathfrak{M}$ also satisfies $\Gamma \cup \{\mathbb{T} t = t\}$.

If the branch is expanded using $=\mathbb{T}$, we add a signed *fórmula* $S\varphi(t_2)$, but Γ contains both $\mathbb{T}t_1 = t_2$ and $\mathbb{T}\varphi(t_1)$. Thus we have $\mathfrak{M} \models t_1 = t_2$ and $\mathfrak{M} \models \varphi(t_1)$. Let s be a variable assignment with $s(x) = \text{Val}^{\mathfrak{M}}(t_1)$. By [Proposição 16.18](#), $\mathfrak{M}, s \models \varphi(t_1)$. Since $s \sim_x s$, by [Proposição 16.23](#), $\mathfrak{M}, s \models \varphi(x)$. since $\mathfrak{M} \models t_1 = t_2$, we have $\text{Val}^{\mathfrak{M}}(t_1) = \text{Val}^{\mathfrak{M}}(t_2)$, and hence $s(x) = \text{Val}^{\mathfrak{M}}(t_2)$. By applying [Proposição 16.23](#) again, we also have $\mathfrak{M}, s \models \varphi(t_2)$. By [Proposição 16.18](#), $\mathfrak{M} \models \varphi(t_2)$. The case of $=\mathbb{F}$ is treated similarly. $\square$

Capítulo 22

Axiomatic Derivações

No effort has been made yet to ensure that the material in this chapter respects various tags indicating which connectives and quantifiers are primitive or defined: all are assumed to be primitive, except $\leftrightarrow$ which is assumed to be defined. If the FOL tag is true, we produce a version with quantifiers, otherwise without.

-NoValue-

[content/first-order-logic/axiomatic-deduction/rules-and-proofs.tex](#)

22.1 Rules and Derivações

fol:axd:rul:
sec Axiomatic *derivações* are perhaps the simplest *derivação* system for logic. explanation A *derivação* is just a sequence of *fórmulas*. To count as a *derivação*, every *fórmula* in the sequence must either be an instance of an axiom, or must follow from one or more *fórmulas* that precede it in the sequence by a rule of inference. A *derivação* *deriva* its last *fórmula*.

Definição 22.1 (Derivabilidade). If Γ is a set of *fórmulas* of $\mathcal{L}$ then a *derivação* from Γ is a finite sequence $\varphi_1, \dots, \varphi_n$ of *fórmulas* where for each $i \leq n$ one of the following holds:

1. $\varphi_i \in \Gamma$; or

2. φ_i is an axiom; or
3. φ_i follows from some φ_j (and φ_k) with $j < i$ (and $k < i$) by a rule of inference.

What counts as a correct *derivação* depends on which inference rules we allow (and of course what we take to be axioms). And an inference rule is an if-then statement that tells us that, under certain conditions, a step A_i in a *derivação* is a correct inference step.

Definição 22.2 (Rule of inference). A *rule of inference* gives a sufficient condition for what counts as a correct inference step in a *derivação* from Γ .

For instance, since any one-element sequence φ with $\varphi \in \Gamma$ trivially counts as a *derivação*, the following might be a very simple rule of inference:

If $\varphi \in \Gamma$, then φ is always a correct inference step in any *derivação* from Γ .

Similarly, if φ is one of the axioms, then φ by itself is a *derivação*, and so this is also a rule of inference:

If φ is an axiom, then φ is a correct inference step.

It gets more interesting if the rule of inference appeals to *fórmulas* that appear before the step considered. The following rule is called *modus ponens*:

If $\psi \rightarrow \varphi$ and ψ occur higher up in the *derivação*, then φ is a correct inference step.

If this is the only rule of inference, then our definition of *derivação* above amounts to this: $\varphi_1, \dots, \varphi_n$ is a *derivação* iff for each $i \leq n$ one of the following holds:

1. $\varphi_i \in \Gamma$; or
2. φ_i is an axiom; or
3. for some $j < i$, φ_j is $\psi \rightarrow \varphi_i$, and for some $k < i$, φ_k is ψ .

The last clause says that φ_i follows from φ_j ($\psi \rightarrow \varphi_i$) and φ_k (ψ) by modus ponens. If we can go from 1 to n , and each time we find a *fórmula* φ_i that is either in Γ , an axiom, or which a rule of inference tells us that it is a correct inference step, then the entire sequence counts as a correct *derivação*.

Definição 22.3 (Derivabilidade). A *fórmula* φ is *derivável* from Γ , written $\Gamma \vdash \varphi$, if there is a *derivação* from Γ ending in φ .

Definição 22.4 (Theorems). A *fórmula* φ is a *theorem* if there is a *derivação* of φ from the empty set. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

-NoValue-

22.2. AXIOMS AND RULES FOR THE PROPOSITIONAL CONNECTIVES

22.2 Axioms and Rules for the Propositional Connectives

fol:axd:prp:
sec

Definição 22.5 (Axioms). The set of Ax_0 of *axioms* for the propositional connectives comprises all *fórmulas* of the following forms:

fol:axd:prp:	$(\varphi \wedge \psi) \rightarrow \varphi$	(22.1)
ax:land1	$(\varphi \wedge \psi) \rightarrow \psi$	(22.2)
fol:axd:prp:	$\varphi \rightarrow (\psi \rightarrow (\varphi \wedge \psi))$	(22.3)
ax:land3	$\varphi \rightarrow (\varphi \vee \psi)$	(22.4)
fol:axd:prp:	$\varphi \rightarrow (\psi \vee \varphi)$	(22.5)
ax:lor2	$(\varphi \rightarrow \chi) \rightarrow ((\psi \rightarrow \chi) \rightarrow ((\varphi \vee \psi) \rightarrow \chi))$	(22.6)
fol:axd:prp:	$\varphi \rightarrow (\psi \rightarrow \varphi)$	(22.7)
ax:lif1	$(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$	(22.8)
fol:axd:prp:	$(\varphi \rightarrow \psi) \rightarrow ((\varphi \rightarrow \neg\psi) \rightarrow \neg\varphi)$	(22.9)
ax:lnot1	$\neg\varphi \rightarrow (\varphi \rightarrow \psi)$	(22.10)
fol:axd:prp:	$\top$	(22.11)
ax:ltrue	$\perp \rightarrow \varphi$	(22.12)
fol:axd:prp:	$(\varphi \rightarrow \perp) \rightarrow \neg\varphi$	(22.13)
ax:lfalse1	$\neg\neg\varphi \rightarrow \varphi$	(22.14)
fol:axd:prp:		
ax:dne		

Definição 22.6 (Modus ponens). If ψ and $\psi \rightarrow \varphi$ already occur in a *derivação*, then φ is a correct inference step.

We'll abbreviate the rule modus ponens as “MP.”
-NoValue-

content/first-order-logic/axiomatic-deduction/axioms-rules-quantifiers.tex

22.3 Axioms and Rules for Quantifiers

fol:axd:qua:
sec

Definição 22.7 (Axioms for quantifiers). The *axioms* governing quantifiers are all instances of the following:

fol:axd:qua:	$\forall x \psi \rightarrow \psi(t),$	(22.15)
ax:q1	$\psi(t) \rightarrow \exists x \psi.$	(22.16)
fol:axd:qua:		
ax:q2		

for any closed term t .

Definição 22.8 (Rules for quantifiers).

If $\psi \rightarrow \varphi(a)$ already occurs in the **derivação** and a does not occur in Γ or ψ , then $\psi \rightarrow \forall x \varphi(x)$ is a correct inference step.

If $\varphi(a) \rightarrow \psi$ already occurs in the **derivação** and a does not occur in Γ or ψ , then $\exists x \varphi(x) \rightarrow \psi$ is a correct inference step.

We'll abbreviate either of these by “QR.”
-NoValue-

`content/first-order-logic/axiomatic-deduction/proving-things.tex`

22.4 Examples of Derivações

Exemplo 22.9. Suppose we want to prove $(\neg\theta \vee \alpha) \rightarrow (\theta \rightarrow \alpha)$. Clearly, this is not an instance of any of our axioms, so we have to use the MP rule to **deriva** it. Our only rule is MP, which given φ and $\varphi \rightarrow \psi$ allows us to justify ψ . One strategy would be to use **Eq. (22.6)** with φ being $\neg\theta$, ψ being α , and χ being $\theta \rightarrow \alpha$. O.s., the instance

$$(\neg\theta \rightarrow (\theta \rightarrow \alpha)) \rightarrow ((\alpha \rightarrow (\theta \rightarrow \alpha)) \rightarrow ((\neg\theta \vee \alpha) \rightarrow (\theta \rightarrow \alpha))).$$

Why? Two applications of MP yield the last part, which is what we want. And we easily see that $\neg\theta \rightarrow (\theta \rightarrow \alpha)$ is an instance of **Eq. (22.10)**, and $\alpha \rightarrow (\theta \rightarrow \alpha)$ is an instance of **Eq. (22.7)**. So our **derivação** is:

1.	$\neg\theta \rightarrow (\theta \rightarrow \alpha)$	Eq. (22.10)
2.	$(\neg\theta \rightarrow (\theta \rightarrow \alpha)) \rightarrow$ $((\alpha \rightarrow (\theta \rightarrow \alpha)) \rightarrow ((\neg\theta \vee \alpha) \rightarrow (\theta \rightarrow \alpha)))$	Eq. (22.6)
3.	$(\alpha \rightarrow (\theta \rightarrow \alpha)) \rightarrow ((\neg\theta \vee \alpha) \rightarrow (\theta \rightarrow \alpha))$	1, 2, MP
4.	$\alpha \rightarrow (\theta \rightarrow \alpha)$	Eq. (22.7)
5.	$(\neg\theta \vee \alpha) \rightarrow (\theta \rightarrow \alpha)$	3, 4, MP

Exemplo 22.10. Let's try to find a **derivação** of $\theta \rightarrow \theta$. It is not an instance of an axiom, so we have to use MP to **deriva** it. **Eq. (22.7)** is an axiom of the form $\varphi \rightarrow \psi$ to which we could apply MP. To be useful, of course, the ψ which MP would justify as a correct step in this case would have to be $\theta \rightarrow \theta$, since this is what we want to **deriva**. That means φ would also have to be θ . O.s., we might look at this instance of **Eq. (22.7)**:

$$\theta \rightarrow (\theta \rightarrow \theta)$$

In order to apply MP, we would also need to justify the corresponding second premise, namely φ . But in our case, that would be θ , and we won't be able to **deriva** θ by itself. So we need a different strategy.

The other axiom involving just $\rightarrow$ is **Eq. (22.8)**, O.s.,

$$(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$$

22.4. EXAMPLES OF DERIVAÇÕES

We could get to the last nested conditional by applying MP twice. Again, that would mean that we want an instance of Eq. (22.8) where $\varphi \rightarrow \chi$ is $\theta \rightarrow \theta$, the *fórmula* we are aiming for. Then of course, φ and χ are both θ . How should we pick ψ so that both $\varphi \rightarrow (\psi \rightarrow \chi)$ and $\varphi \rightarrow \psi$, O.s., in our case $\theta \rightarrow (\psi \rightarrow \theta)$ and $\theta \rightarrow \psi$, are also *derivável*? Well, the first of these is already an instance of Eq. (22.7), whatever we decide ψ to be. And $\theta \rightarrow \psi$ would be another instance of Eq. (22.7) if ψ were $(\theta \rightarrow \theta)$. So, our *derivação* is:

1. $\theta \rightarrow ((\theta \rightarrow \theta) \rightarrow \theta)$ Eq. (22.7)
2. $(\theta \rightarrow ((\theta \rightarrow \theta) \rightarrow \theta)) \rightarrow$
 $((\theta \rightarrow (\theta \rightarrow \theta)) \rightarrow (\theta \rightarrow \theta))$ Eq. (22.8)
3. $(\theta \rightarrow (\theta \rightarrow \theta)) \rightarrow (\theta \rightarrow \theta)$ 1, 2, MP
4. $\theta \rightarrow (\theta \rightarrow \theta)$ Eq. (22.7)
5. $\theta \rightarrow \theta$ 3, 4, MP

Exemplo 22.11. Sometimes we want to show that there is a *derivação* of some *fórmula* from some other *fórmulas* Γ . For instance, let's show that we can *deriva* $\varphi \rightarrow \chi$ from $\Gamma = \{\varphi \rightarrow \psi, \psi \rightarrow \chi\}$.

1. $\varphi \rightarrow \psi$ HYP
2. $\psi \rightarrow \chi$ HYP
3. $(\psi \rightarrow \chi) \rightarrow (\varphi \rightarrow (\psi \rightarrow \chi))$ Eq. (22.7)
4. $\varphi \rightarrow (\psi \rightarrow \chi)$ 2, 3, MP
5. $(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow$
 $((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$ Eq. (22.8)
6. $((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$ 4, 5, MP
7. $\varphi \rightarrow \chi$ 1, 6, MP

The lines labelled “HYP” (for “hypothesis”) indicate that the *fórmula* on that line is *um membro* of Γ .

Proposição 22.12. *If $\Gamma \vdash \varphi \rightarrow \psi$ and $\Gamma \vdash \psi \rightarrow \chi$, then $\Gamma \vdash \varphi \rightarrow \chi$*

Prova. Suppose $\Gamma \vdash \varphi \rightarrow \psi$ and $\Gamma \vdash \psi \rightarrow \chi$. Then there is a *derivação* of $\varphi \rightarrow \psi$ from Γ ; and a *derivação* of $\psi \rightarrow \chi$ from Γ as well. Combine these into a single *derivação* by concatenating them. Now add lines 3–7 of the *derivação* in the preceding example. This is a *derivação* of $\varphi \rightarrow \chi$ —which is the last line of the new *derivação*—from Γ . Note that the justifications of lines 4 and 7 remain valid if the reference to line number 1 is replaced by reference to the last line of the *derivação* of $\varphi \rightarrow \psi$, and reference to line number 2 by reference to the last line of the *derivação* of $\psi \rightarrow \chi$. $\square$

Problema 22.1. Show that the following hold by exhibiting *derivações* from the axioms:

1. $(\varphi \wedge \psi) \rightarrow (\psi \wedge \varphi)$
2. $((\varphi \wedge \psi) \rightarrow \chi) \rightarrow (\varphi \rightarrow (\psi \rightarrow \chi))$

3. $\neg(\varphi \vee \psi) \rightarrow \neg\varphi$

-NoValue-

`content/first-order-logic/axiomatic-deduction/proving-things-quant.tex`

22.5 Derivações with Quantifiers

Exemplo 22.13. Let us give a *derivação* of $(\forall x \varphi(x) \wedge \forall y \psi(y)) \rightarrow \forall x (\varphi(x) \wedge \psi(x))$. fol:axd:prq:
sec

First, note that

$$(\forall x \varphi(x) \wedge \forall y \psi(y)) \rightarrow \forall x \varphi(x)$$

is an instance of **Eq. (22.1)**, and

$$\forall x \varphi(x) \rightarrow \varphi(a)$$

of **Eq. (22.15)**. So, by **Proposição 22.12**, we know that

$$(\forall x \varphi(x) \wedge \forall y \psi(y)) \rightarrow \varphi(a)$$

is *derivável*. Likewise, since

$$\begin{aligned} (\forall x \varphi(x) \wedge \forall y \psi(y)) &\rightarrow \forall y \psi(y) && \text{and} \\ \forall y \psi(y) &\rightarrow \psi(a) \end{aligned}$$

are instances of **Eq. (22.2)** and **Eq. (22.15)**, respectively,

$$(\forall x \varphi(x) \wedge \forall y \psi(y)) \rightarrow \psi(a)$$

is derivable by **Proposição 22.12**. Using an appropriate instance of **Eq. (22.3)** and two applications of MP, we see that

$$(\forall x \varphi(x) \wedge \forall y \psi(y)) \rightarrow (\varphi(a) \wedge \psi(a))$$

is derivable. We can now apply QR to obtain

$$(\forall x \varphi(x) \wedge \forall y \psi(y)) \rightarrow \forall x (\varphi(x) \wedge \psi(x)).$$

-NoValue-

`content/first-order-logic/axiomatic-deduction/proof-theoretic-notions.tex`

22.6. PROOF-THEORETIC NOTIONS

22.6 Proof-Theoretic Notions

fol:axd:ptn: sec Just as we've defined a number of important semantic notions (validity, entailment, satisfiability), we now define corresponding *proof-theoretic notions*. explanation These are not defined by appeal to satisfaction of *sentenças* in *estruturas*, but by appeal to the *derivabilidade* or *não derivabilidade* of certain formulas. It was an important discovery that these notions coincide. That they do is the content of the *soundness* and *completeness theorems*.

Definição 22.14 (Derivabilidade). A fórmula φ is *derivável* from Γ , written $\Gamma \vdash \varphi$, if there is a *derivação* from Γ ending in φ .

Definição 22.15 (Theorems). A fórmula φ is a *theorem* if there is a *derivação* of φ from the empty set. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

Definição 22.16 (Consistency). A set Γ of *fórmulas* is *consistent* if and only if $\Gamma \nvdash \perp$; it is *inconsistent* otherwise.

fol:axd:ptn: prop:reflexivity **Proposição 22.17 (Reflexivity).** If $\varphi \in \Gamma$, then $\Gamma \vdash \varphi$.

Prova. The fórmula φ by itself is a *derivação* of φ from Γ . □

fol:axd:ptn: prop:monotonicity **Proposição 22.18 (Monotonicity).** If $\Gamma \subseteq \Delta$ and $\Gamma \vdash \varphi$, then $\Delta \vdash \varphi$.

Prova. Any *derivação* of φ from Γ is also a *derivação* of φ from Δ . □

fol:axd:ptn: prop:transitivity **Proposição 22.19 (Transitivity).** If $\Gamma \vdash \varphi$ and $\{\varphi\} \cup \Delta \vdash \psi$, then $\Gamma \cup \Delta \vdash \psi$.

Prova. Suppose $\{\varphi\} \cup \Delta \vdash \psi$. Then there is a *derivação* $\psi_1, \dots, \psi_l = \psi$ from $\{\varphi\} \cup \Delta$. Some of the steps in that *derivação* will be correct because of a rule which refers to a prior line $\psi_i = \varphi$. By hypothesis, there is a *derivação* of φ from Γ . O.s., a *derivação* $\varphi_1, \dots, \varphi_k = \varphi$ where every φ_i is an axiom, *um membro* of Γ , or correct by a rule of inference. Now consider the sequence

$$\varphi_1, \dots, \varphi_k = \varphi, \psi_1, \dots, \psi_l = \psi.$$

This is a correct *derivação* of ψ from $\Gamma \cup \Delta$ since every $B_i = \varphi$ is now justified by the same rule which justifies $\varphi_k = \varphi$. □

Note that this means that in particular if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. It follows also that if $\varphi_1, \dots, \varphi_n \vdash \psi$ and $\Gamma \vdash \varphi_i$ for each i , then $\Gamma \vdash \psi$.

fol:axd:ptn: prop:incons **Proposição 22.20.** Γ is inconsistent iff $\Gamma \vdash \varphi$ for every φ .

Prova. Exercise. □

Problema 22.2. Prove *Proposição 22.20*.

Proposição 22.21 (Compactness).

fol:axd:ptn:
prop:proves-compact

1. If $\Gamma \vdash \varphi$ then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vdash \varphi$.
2. If every finite subset of Γ is consistent, then Γ is consistent.

Prova. 1. If $\Gamma \vdash \varphi$, then there is a finite sequence of fórmulas $\varphi_1, \dots, \varphi_n$ so that $\varphi \equiv \varphi_n$ and each φ_i is either a logical axiom, um membro of Γ or follows from previous fórmulas by modus ponens. Take Γ_0 to be those φ_i which are in Γ . Then the derivação is likewise a derivação from Γ_0 , and so $\Gamma_0 \vdash \varphi$.

2. This is the contrapositive of (1) for the special case $\varphi \equiv \perp$. □

-NoValue-

content/first-order-logic/axiomatic-deduction/deduction-theorem.tex

22.7 The Deduction Theorem

As we've seen, giving derivações in an axiomatic system is cumbersome, and derivações may be hard to find. Rather than actually write out long lists of fórmulas, it is generally easier to argue that such derivações exist, by making use of a few simple results. We've already established three such results: **Proposição 22.17** says we can always assert that $\Gamma \vdash \varphi$ when we know that $\varphi \in \Gamma$. **Proposição 22.18** says that if $\Gamma \vdash \varphi$ then also $\Gamma \cup \{\psi\} \vdash \varphi$. And **Proposição 22.19** implies that if $\Gamma \vdash \varphi$ and $\varphi \vdash \psi$, then $\Gamma \vdash \psi$. Here's another simple result, a "meta"-version of modus ponens:

fol:axd:ded:
sec

Proposição 22.22. If $\Gamma \vdash \varphi$ and $\Gamma \vdash \varphi \rightarrow \psi$, then $\Gamma \vdash \psi$.

fol:axd:ded:
prop:mp

Prova. We have that $\{\varphi, \varphi \rightarrow \psi\} \vdash \psi$:

1. φ Hyp.
2. $\varphi \rightarrow \psi$ Hyp.
3. ψ 1, 2, MP

By **Proposição 22.19**, $\Gamma \vdash \psi$. □

The most important result we'll use in this context is the deduction theorem:

Teorema 22.23 (Deduction Theorem). $\Gamma \cup \{\varphi\} \vdash \psi$ if and only if $\Gamma \vdash \varphi \rightarrow \psi$.

fol:axd:ded:
thm:deduction-thm

Prova. The "if" direction is immediate. If $\Gamma \vdash \varphi \rightarrow \psi$ then also $\Gamma \cup \{\varphi\} \vdash \varphi \rightarrow \psi$ by **Proposição 22.18**. Also, $\Gamma \cup \{\varphi\} \vdash \varphi$ by **Proposição 22.17**. So, by **Proposição 22.22**, $\Gamma \cup \{\varphi\} \vdash \psi$.

For the "only if" direction, we proceed by induction on the length of the derivação of ψ from $\Gamma \cup \{\varphi\}$.

22.7. THE DEDUCTION THEOREM

For the induction basis, we prove the claim for every **derivação** of length 1. A **derivação** of ψ from $\Gamma \cup \{\varphi\}$ of length 1 consists of ψ by itself; and if it is correct ψ is either $\in \Gamma \cup \{\varphi\}$ or is an axiom. If $\psi \in \Gamma$ or is an axiom, then $\Gamma \vdash \psi$. We also have that $\Gamma \vdash \psi \rightarrow (\varphi \rightarrow \psi)$ by **Eq. (22.7)**, and **Proposição 22.22** gives $\Gamma \vdash \varphi \rightarrow \psi$. If $\psi \in \{\varphi\}$ then $\Gamma \vdash \varphi \rightarrow \psi$ because the last **sentença** $\varphi \rightarrow \psi$ is the same as $\varphi \rightarrow \varphi$, and we have **derivad** that in **Exemplo 22.10**.

For the inductive step, suppose a **derivação** of ψ from $\Gamma \cup \{\varphi\}$ ends with a step ψ which is justified by modus ponens. (If it is not justified by modus ponens, $\psi \in \Gamma$, $\psi \equiv \varphi$, or ψ is an axiom, and the same reasoning as in the induction basis applies.) Then some previous steps in the **derivação** are $\chi \rightarrow \psi$ and χ , for some **fórmula** χ . O.s., $\Gamma \cup \{\varphi\} \vdash \chi \rightarrow \psi$ and $\Gamma \cup \{\varphi\} \vdash \chi$, and the respective **derivações** are shorter, so the inductive hypothesis applies to them. We thus have both:

$$\begin{aligned}\Gamma \vdash \varphi \rightarrow (\chi \rightarrow \psi); \\ \Gamma \vdash \varphi \rightarrow \chi.\end{aligned}$$

But also

$$\Gamma \vdash (\varphi \rightarrow (\chi \rightarrow \psi)) \rightarrow ((\varphi \rightarrow \chi) \rightarrow (\varphi \rightarrow \psi)),$$

by **Eq. (22.8)**, and two applications of **Proposição 22.22** give $\Gamma \vdash \varphi \rightarrow \psi$, as required. $\square$

Notice how **Eq. (22.7)** and **Eq. (22.8)** were chosen precisely so that the Deduction Theorem would hold.

The following are some useful facts about **derivabilidade**, which we leave as exercises.

Proposição 22.24.

*fol:axd:ded:
prop:derivfacts*

1. $\vdash (\varphi \rightarrow \psi) \rightarrow ((\psi \rightarrow \chi) \rightarrow (\varphi \rightarrow \chi));$
2. If $\Gamma \cup \{\neg\varphi\} \vdash \neg\psi$ then $\Gamma \cup \{\psi\} \vdash \varphi$ (Contraposition);
3. $\{\varphi, \neg\varphi\} \vdash \psi$ (Ex Falso Quodlibet, Explosion);
4. $\{\neg\neg\varphi\} \vdash \varphi$ (Double Negation Elimination);
5. If $\Gamma \vdash \neg\neg\varphi$ then $\Gamma \vdash \varphi$;

*fol:axd:ded:
derivfacts:a*

*fol:axd:ded:
derivfacts:b*

*fol:axd:ded:
derivfacts:c*

*fol:axd:ded:
derivfacts:d*

*fol:axd:ded:
derivfacts:e*

Problema 22.3. Prove **Proposição 22.24**

-NoValue-

content/first-order-logic/axiomatic-deduction/deduction-theorem-quantifiers.tex

22.8 The Deduction Theorem with Quantifiers

Teorema 22.25 (Deduction Theorem). *If $\Gamma \cup \{\varphi\} \vdash \psi$, then $\Gamma \vdash \varphi \rightarrow \psi$.*

fol:axd:ddq:
sec
fol:axd:ddq:
thm:deduction-thm-q

Prova. We again proceed by induction on the length of the *derivação* of ψ from $\Gamma \cup \{\varphi\}$.

The proof of the induction basis is identical to that in the proof of **Teorema 22.23**.

For the inductive step, suppose again that the *derivação* of ψ from $\Gamma \cup \{\varphi\}$ ends with a step ψ which is justified by an inference rule. If the inference rule is modus ponens, we proceed as in the proof of **Teorema 22.23**. If the inference rule is QR, we know that $\psi \equiv \chi \rightarrow \forall x \theta(x)$ and a *fórmula* of the form $\chi \rightarrow \theta(a)$ appears earlier in the *derivação*, where a does not occur in χ , φ , or Γ . We thus have that

$$\Gamma \cup \{\varphi\} \vdash \chi \rightarrow \theta(a),$$

and the induction hypothesis applies, O.s., we have that

$$\Gamma \vdash \varphi \rightarrow (\chi \rightarrow \theta(a)).$$

By

$$\vdash (\varphi \rightarrow (\chi \rightarrow \theta(a))) \rightarrow ((\varphi \wedge \chi) \rightarrow \theta(a))$$

and modus ponens we get

$$\Gamma \vdash (\varphi \wedge \chi) \rightarrow \theta(a).$$

Since the eigenvariable condition still applies, we can add a step to this *derivação* justified by QR, and get

$$\Gamma \vdash (\varphi \wedge \chi) \rightarrow \forall x \theta(x).$$

We also have

$$\vdash ((\varphi \wedge \chi) \rightarrow \forall x \theta(x)) \rightarrow (\varphi \rightarrow (\chi \rightarrow \forall x \theta(x))),$$

so by modus ponens,

$$\Gamma \vdash \varphi \rightarrow (\chi \rightarrow \forall x \theta(x)),$$

O.s., $\Gamma \vdash \psi$.

We leave the case where ψ is justified by the rule QR, but is of the form $\exists x \theta(x) \rightarrow \chi$, as an exercise. $\square$

Problema 22.4. Complete the proof of **Teorema 22.25**.

-NoValue-

22.9 Derivabilidade and Consistency

fol:axd:prv:
sec

We will now establish a number of properties of the **derivabilidade** relation. They are independently interesting, but each will play a role in the proof of the completeness theorem.

fol:axd:prv:
prop:provability-contr

Proposição 22.26. *If $\Gamma \vdash \varphi$ and $\Gamma \cup \{\varphi\}$ is inconsistent, then Γ is inconsistent.*

Prova. If $\Gamma \cup \{\varphi\}$ is inconsistent, then $\Gamma \cup \{\varphi\} \vdash \perp$. By **Proposição 22.17**, $\Gamma \vdash \psi$ for every $\psi \in \Gamma$. Since also $\Gamma \vdash \varphi$ by hypothesis, $\Gamma \vdash \psi$ for every $\psi \in \Gamma \cup \{\varphi\}$. By **Proposição 22.19**, $\Gamma \vdash \perp$, O.s., Γ is inconsistent. $\square$

fol:axd:prv:
prop:prov-incons

Proposição 22.27. *$\Gamma \vdash \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ is inconsistent.*

Prova. First suppose $\Gamma \vdash \varphi$. Then $\Gamma \cup \{\neg\varphi\} \vdash \varphi$ by **Proposição 22.18**. $\Gamma \cup \{\neg\varphi\} \vdash \neg\varphi$ by **Proposição 22.17**. We also have $\vdash \neg\varphi \rightarrow (\varphi \rightarrow \perp)$ by **Eq. (22.10)**. So by two applications of **Proposição 22.22**, we have $\Gamma \cup \{\neg\varphi\} \vdash \perp$.

Now assume $\Gamma \cup \{\neg\varphi\}$ is inconsistent, O.s., $\Gamma \cup \{\neg\varphi\} \vdash \perp$. By the deduction theorem, $\Gamma \vdash \neg\varphi \rightarrow \perp$. $\Gamma \vdash (\neg\varphi \rightarrow \perp) \rightarrow \neg\neg\varphi$ by **Eq. (22.13)**, so $\Gamma \vdash \neg\neg\varphi$ by **Proposição 22.22**. Since $\Gamma \vdash \neg\neg\varphi \rightarrow \varphi$ (**Eq. (22.14)**), we have $\Gamma \vdash \varphi$ by **Proposição 22.22** again. $\square$

Problema 22.5. Prove that $\Gamma \vdash \neg\varphi$ iff $\Gamma \cup \{\varphi\}$ is inconsistent.

fol:axd:prv:
prop:explicit-inc

Proposição 22.28. *If $\Gamma \vdash \varphi$ and $\neg\varphi \in \Gamma$, then Γ is inconsistent.*

Prova. $\Gamma \vdash \neg\varphi \rightarrow (\varphi \rightarrow \perp)$ by **Eq. (22.10)**. $\Gamma \vdash \perp$ by two applications of **Proposição 22.22**. $\square$

fol:axd:prv:
prop:provability-exhaustive

Proposição 22.29. *If $\Gamma \cup \{\varphi\}$ and $\Gamma \cup \{\neg\varphi\}$ are both inconsistent, then Γ is inconsistent.*

Prova. Exercise. $\square$

Problema 22.6. Prove **Proposição 22.29**

-NoValue-

content/first-order-logic/axiomatic-deduction/provability-propositional.tex

22.10 Derivabilidade and the Propositional Connectives

explanation We establish that the **derivabilidade** relation $\vdash$ of axiomatic deduction is strong enough to establish some basic facts involving the propositional connectives, such as that $\varphi \wedge \psi \vdash \varphi$ and $\varphi, \varphi \rightarrow \psi \vdash \psi$ (modus ponens). These facts are needed for the proof of the completeness theorem. fol:axd:ppr: sec

Proposição 22.30.

1. Both $\varphi \wedge \psi \vdash \varphi$ and $\varphi \wedge \psi \vdash \psi$
2. $\varphi, \psi \vdash \varphi \wedge \psi$.

fol:axd:ppr:
prop:provability-land
fol:axd:ppr:
prop:provability-land-left
fol:axd:ppr:
prop:provability-land-right

Prova. 1. From **Eq. (22.1)** and **Eq. (22.1)** by modus ponens.
2. From **Eq. (22.3)** by two applications of modus ponens. □

Proposição 22.31.

1. $\varphi \vee \psi, \neg\varphi, \neg\psi$ is inconsistent.
2. Both $\varphi \vdash \varphi \vee \psi$ and $\psi \vdash \varphi \vee \psi$.

fol:axd:ppr:
prop:provability-lor

Prova. 1. From **Eq. (22.9)** we get $\vdash \neg\varphi \rightarrow (\varphi \rightarrow \perp)$ and $\vdash \neg\psi \rightarrow (\psi \rightarrow \perp)$. So by the deduction theorem, we have $\{\neg\varphi\} \vdash \varphi \rightarrow \perp$ and $\{\neg\psi\} \vdash \psi \rightarrow \perp$. From **Eq. (22.6)** we get $\{\neg\varphi, \neg\psi\} \vdash (\varphi \vee \psi) \rightarrow \perp$. By the deduction theorem, $\{\varphi \vee \psi, \neg\varphi, \neg\psi\} \vdash \perp$.
2. From **Eq. (22.4)** and **Eq. (22.5)** by modus ponens. □

Proposição 22.32.

1. $\varphi, \varphi \rightarrow \psi \vdash \psi$.
2. Both $\neg\varphi \vdash \varphi \rightarrow \psi$ and $\psi \vdash \varphi \rightarrow \psi$.

fol:axd:ppr:
prop:provability-lif
fol:axd:ppr:
prop:provability-lif-left
fol:axd:ppr:
prop:provability-lif-right

Prova. 1. We can **deriva**:

1.	φ	HYP
2.	$\varphi \rightarrow \psi$	HYP
3.	ψ	1, 2, MP

2. By **Eq. (22.10)** and **Eq. (22.7)** and the deduction theorem, respectively. □

-NoValue-

22.11 Derivabilidade and the Quantifiers

fol:axd:qpr:sec The completeness theorem also requires that axiomatic deductions yield the explanation facts about $\vdash$ established in this section.

fol:axd:qpr:thm:strong-generalization **Teorema 22.33.** *If c is a **símbolo de constante** not occurring in Γ or $\varphi(x)$ and $\Gamma \vdash \varphi(c)$, then $\Gamma \vdash \forall x \varphi(x)$.*

Prova. By the deduction theorem, $\Gamma \vdash \top \rightarrow \varphi(c)$. Since c does not occur in Γ or $\top$, we get $\Gamma \vdash \top \rightarrow \forall x \varphi(x)$. By the deduction theorem again, $\Gamma \vdash \forall x \varphi(x)$. $\square$

fol:axd:qpr:prop:provability-quantifiers

Proposição 22.34.

1. $\varphi(t) \vdash \exists x \varphi(x)$.
2. $\forall x \varphi(x) \vdash \varphi(t)$.

Prova. 1. By **Eq. (22.16)** and the deduction theorem.

2. By **Eq. (22.15)** and the deduction theorem. $\square$

-NoValue-

content/first-order-logic/axiomatic-deduction/soundness.tex

22.12 Soundness

fol:axd:sou:sec A **derivação** system, such as axiomatic deduction, is *sound* if it cannot **deriva** explanation things that do not actually hold. Soundness is thus a kind of guaranteed safety property for **derivação** systems. Depending on which proof theoretic property is in question, we would like to know for instance, that

1. every **derivável** φ is valid;
2. if φ is **derivável** from some others Γ , it is also a consequence of them;
3. if a set of **fórmulas** Γ is inconsistent, it is unsatisfiable.

These are important properties of a **derivação** system. If any of them do not hold, the **derivação** system is deficient—it would **deriva** too much. Consequently, establishing the soundness of a **derivação** system is of the utmost importance.

Proposição 22.35. *If φ is an axiom, then $\mathfrak{M}, s \models \varphi$ for each **estrutura** $\mathfrak{M}$ and assignment s .*

Prova. We have to verify that all the axioms are valid. For instance, here is the case for **Eq. (22.15)**: suppose t is **livre para** x in φ , and assume $\mathfrak{M}, s \models \forall x \varphi$. Then by definition of satisfaction, for each $s' \sim_x s$, also $\mathfrak{M}, s' \models \varphi$, and in particular this holds when $s'(x) = \text{Val}_s^{\mathfrak{M}}(t)$. By **Proposição 16.23**, $\mathfrak{M}, s \models \varphi[t/x]$. This shows that $\mathfrak{M}, s \models (\forall x \varphi \rightarrow \varphi[t/x])$. $\square$

Teorema 22.36 (Soundness). *If $\Gamma \vdash \varphi$ then $\Gamma \models \varphi$.*

*fol:axd:sou:
thm:soundness*

Prova. By induction on the length of the **derivação** of φ from Γ . If there are no steps justified by inferences, then all **fórmulas** in the **derivação** are either instances of axioms or are in Γ . By the previous proposition, all the axioms are valid, and hence if φ is an axiom then $\Gamma \models \varphi$. If $\varphi \in \Gamma$, then trivially $\Gamma \models \varphi$.

If the last step of the **derivação** of φ is justified by modus ponens, then there are **fórmulas** ψ and $\psi \rightarrow \varphi$ in the **derivação**, and the induction hypothesis applies to the part of the **derivação** ending in those **fórmulas** (since they contain at least one fewer step justified by an inference). So, by induction hypothesis, $\Gamma \models \psi$ and $\Gamma \models \psi \rightarrow \varphi$. Then $\Gamma \models \varphi$ by **Teorema 16.30**.

Now suppose the last step is justified by QR. Then that step has the form $\chi \rightarrow \forall x B(x)$ and there is a preceding step $\chi \rightarrow \psi(c)$ with c not in Γ , χ , or $\forall x B(x)$. By induction hypothesis, $\Gamma \models \chi \rightarrow \psi(c)$. By **Teorema 16.30**, $\Gamma \cup \{\chi\} \models \psi(c)$.

Consider some structure $\mathfrak{M}$ such that $\mathfrak{M} \models \Gamma \cup \{\chi\}$. We need to show that $\mathfrak{M} \models \forall x \psi(x)$. Since $\forall x \psi(x)$ is a **sentença**, this means we have to show that for every variable assignment s , $\mathfrak{M}, s \models \psi(x)$ (**Proposição 16.19**). Since $\Gamma \cup \{\chi\}$ consists entirely of sentences, $\mathfrak{M}, s \models \theta$ for all $\theta \in \Gamma$ by **Definição 16.11**. Let $\mathfrak{M}'$ be like $\mathfrak{M}$ except that $c^{\mathfrak{M}'} = s(x)$. Since c does not occur in Γ or χ , $\mathfrak{M}' \models \Gamma \cup \{\chi\}$ by **Corolário 16.21**. Since $\Gamma \cup \{\chi\} \models \psi(c)$, $\mathfrak{M}' \models \psi(c)$. Since $\psi(c)$ is a **sentença**, $\mathfrak{M}', s \models \psi(c)$ by **Proposição 16.18**. $\mathfrak{M}', s \models \psi(x)$ iff $\mathfrak{M}', s \models \psi(c)$ by **Proposição 16.23** (recall that $\psi(c)$ is just $\psi(x)[c/x]$). So, $\mathfrak{M}', s \models \psi(x)$. Since c does not occur in $\psi(x)$, by **Proposição 16.20**, $\mathfrak{M}, s \models \psi(x)$. But s was an arbitrary variable assignment, so $\mathfrak{M} \models \forall x \psi(x)$. Thus $\Gamma \cup \{\chi\} \models \forall x \psi(x)$. By **Teorema 16.30**, $\Gamma \models \chi \rightarrow \forall x \psi(x)$.

The case where φ is justified by QR but is of the form $\exists x \psi(x) \rightarrow \chi$ is left as an exercise. $\square$

Problema 22.7. Complete the proof of **Teorema 22.36**.

Corolário 22.37. *If $\Gamma \vdash \varphi$, then φ is valid.*

*fol:axd:sou:
cor:weak-soundness*

Corolário 22.38. *If Γ is satisfiable, then it is consistent.*

*fol:axd:sou:
cor:consistency-soundness*

Prova. We prove the contrapositive. Suppose that Γ is not consistent. Then $\Gamma \vdash \perp$, O.s., there is a **derivação** of $\perp$ from Γ . By **Teorema 22.36**, any **estrutura** $\mathfrak{M}$ that satisfies Γ must satisfy $\perp$. Since $\mathfrak{M} \not\models \perp$ for every **estrutura** $\mathfrak{M}$, no $\mathfrak{M}$ can satisfy Γ , O.s., Γ is not satisfiable. $\square$

-NoValue-

22.13 Derivações with Predicado de identidade

fol:axd:ide: sec In order to accommodate = in **derivações**, we simply add new axiom schemas. The definition of **derivação** and $\vdash$ remains the same, we just also allow the new axioms.

Definição 22.39 (Axioms for **predicado de identidade**).

$$\text{fol:axd:ide: ax:id1} \quad t = t, \quad (22.17)$$

$$\text{fol:axd:ide: ax:id2} \quad t_1 = t_2 \rightarrow (\psi(t_1) \rightarrow \psi(t_2)), \quad (22.18)$$

for any closed terms t, t_1, t_2 .

fol:axd:ide: prop:sound **Proposição 22.40.** *The axioms Eq. (22.17) and Eq. (22.18) are valid.*

Prova. Exercise. □

Problema 22.8. Prove **Proposição 22.40**.

fol:axd:ide: prop:iden1 **Proposição 22.41.** $\Gamma \vdash t = t$, for any term t and set Γ .

fol:axd:ide: prop:iden2 **Proposição 22.42.** If $\Gamma \vdash \varphi(t_1)$ and $\Gamma \vdash t_1 = t_2$, then $\Gamma \vdash \varphi(t_2)$.

Prova. The **fórmula**

$$(t_1 = t_2 \rightarrow (\varphi(t_1) \rightarrow \varphi(t_2)))$$

is an instance of **Eq. (22.18)**. The conclusion follows by two applications of MP. □

Capítulo 23

O Teorema da Completude

23.1 Introdução

O teorema da completude é um dos resultados mais fundamentais sobre lógica. Ele vem em duas formulações, cuja equivalência iremos provar. Em sua primeira formulação, ele diz algo fundamental sobre a relação entre consequência semântica e nosso sistema de **derivação**: se a **sentença** φ segue de algumas **sentenças** Γ , então também existe a **derivação** que estabelece $\Gamma \vdash \varphi$. Assim, o sistema de **derivação** é tão forte quanto pode possivelmente ser sem provar coisas que de fato não se seguem. fol:com:int:
sec

Em sua segunda formulação, ele pode ser enunciado como um resultado de existência de modelo: todo conjunto consistente de **sentenças** é satisfatível. A consistência é uma noção da teoria da prova: ela diz que nosso sistema de **derivação** é incapaz de produzir certas **derivações**. Mas quem garante que, só porque não há **derivações** de um certo tipo a partir de Γ , está garantido que existe a **estrutura** $\mathfrak{M}$? Antes de o teorema da completude ser provado pela primeira vez — na verdade, antes de termos os sistemas de **derivação** que temos hoje — o grande matemático alemão David Hilbert defendia a visão de que a consistência das teorias matemáticas garante a existência dos objetos de que elas tratam. Ele colocou isso da seguinte forma em uma carta a Gottlob Frege:

Se os axiomas dados arbitrariamente não se contradizem uns aos outros com todas as suas consequências, então eles são verdadeiros e as coisas definidas pelos axiomas existem. Este é para mim o critério de verdade e existência.

Frege discordou veementemente. A segunda formulação do teorema da completude mostra que Hilbert estava certo pelo menos no sentido de que, se os axiomas são consistentes, então existe *alguma estrutura* que os torna todos verdadeiros.

Estas não são as únicas razões pelas quais o teorema da completude — ou melhor, sua prova — é importante. Ele tem uma série de consequências importantes, algumas das quais discutiremos separadamente. Por exemplo, como toda **derivação** que mostra $\Gamma \vdash \varphi$ é finita e, portanto, só pode usar uma quantidade finita das **sentenças** em Γ , segue do teorema da completude que, se φ é uma consequência de Γ , então já é uma consequência de um subconjunto finito de Γ . Isso é chamado de *compacidade*. Equivalentemente, se todo subconjunto finito de Γ é consistente, então o próprio Γ deve ser consistente.

Embora o teorema da compacidade decorra do teorema da completude pelo desvio através de **derivações**, também é possível usar a *prova do* teorema da completude para estabelecê-lo diretamente. Pois o que a prova faz é tomar um conjunto de **sentenças** com uma certa propriedade — a consistência — e construir a **estrutura** a partir desse conjunto que possui certas propriedades (neste caso, que ela satisfaz o conjunto). Quase exatamente a mesma construção pode ser usada para estabelecer a compacidade diretamente, partindo de conjuntos “finitamente satisfatíveis” de **sentenças** em vez de conjuntos consistentes. A construção também produz outras consequências, por exemplo, que todo conjunto satisfatível de **sentenças** tem um modelo finito ou **infinitos**

23.2. ESBOÇO DA PROVA

enumeráveis. (Esse resultado é chamado de teorema de Löwenheim–Skolem.) Em geral, a construção de **estruturas** a partir de conjuntos de **sentenças** é usada com frequência em lógica, e às vezes até em filosofia.

content/first-order-logic/completeness/outline.tex

23.2 Esboço da Prova

fol:com:out:
sec

A prova do teorema da completude é um pouco complexa e, em uma primeira leitura, é fácil se perder. Então, vamos esboçar a prova. O primeiro passo é uma mudança de perspectiva, que nos permite enxergar um caminho para uma prova. Quando a completude é pensada como “sempre que $\Gamma \models \varphi$ então $\Gamma \vdash \varphi$ ”, pode ser difícil até mesmo ter uma ideia: pois, para mostrar que $\Gamma \vdash \varphi$, temos que encontrar **a derivação**, e não parece que a hipótese de que $\Gamma \models \varphi$ nos ajude nisso de alguma forma. Para alguns sistemas de prova, é possível construir diretamente **a derivação**, mas adotaremos uma abordagem um pouco diferente. A mudança de perspectiva necessária é esta: a completude também pode ser formulada como: “se Γ é consistente, então é satisfatível”. Talvez possamos usar a informação em Γ juntamente com a hipótese de que ela é consistente para construir **a estrutura** que satisfaça toda **sentença** em Γ . Afinal, sabemos que tipo de **estrutura** estamos procurando: uma que seja como Γ a descreve!

Se Γ contém apenas **sentenças** atômicas, é fácil construir um modelo para ele. Suponha que as **sentenças** atômicas sejam todas da forma $P(a_1, \dots, a_n)$ onde os a_i são **símbolos de constante**. Tudo o que temos a fazer é apresentar **a domínio** $|\mathfrak{M}|$ e uma atribuição para P de modo que $\mathfrak{M} \models P(a_1, \dots, a_n)$. Mas isso não é muito difícil: faça $|\mathfrak{M}| = \mathbb{N}$, $c_i^{\mathfrak{M}} = i$ e, para cada $P(a_1, \dots, a_n) \in \Gamma$, coloque a tupla $\langle k_1, \dots, k_n \rangle$ em $P^{\mathfrak{M}}$, onde k_i é o índice do símbolo de constante a_i (isto é, $a_i \equiv c_{k_i}$).

Agora suponha que Γ contenha alguma **fórmula** $\neg\psi$, com ψ atômica. Poderíamos temer que a construção de $\mathfrak{M}$ interfira na possibilidade de tornar $\neg\psi$ verdadeira. Mas é aqui que entra a consistência de Γ : se $\neg\psi \in \Gamma$, então $\psi \notin \Gamma$, caso contrário Γ seria inconsistente. E se $\psi \notin \Gamma$, então, de acordo com nossa construção de $\mathfrak{M}$, $\mathfrak{M} \not\models \psi$, logo $\mathfrak{M} \models \neg\psi$. Até aqui, tudo bem.

E se Γ contiver fórmulas complexas, não atômicas? Digamos que ele contenha $\varphi \wedge \psi$. Para tornar isso verdadeiro, devemos proceder como se tanto φ quanto ψ estivessem em Γ . E se $\varphi \vee \psi \in \Gamma$, então teremos que tornar pelo menos um deles verdadeiro, isto é, proceder como se um deles estivesse em Γ .

Isso sugere a seguinte ideia: adicionamos **fórmulas** adicionais a Γ de modo a (a) manter o conjunto resultante consistente e (b) garantir que, para toda **sentença** φ atômica possível, ou φ está no conjunto resultante, ou $\neg\varphi$ está, e (c) tais que, sempre que $\varphi \wedge \psi$ está no conjunto, então tanto φ quanto ψ também estão; se $\varphi \vee \psi$ está no conjunto, pelo menos um dentre φ ou ψ também está, etc. Continuamos fazendo isso (potencialmente para sempre). Chame o conjunto de todas as **fórmulas** assim adicionadas de Γ^* . Então a nossa construção acima nos forneceria **a estrutura** $\mathfrak{M}$ para a qual poderíamos provar, por indução, que

ela satisfaz todas as sentenças em Γ^* e, portanto, também todas as sentenças em Γ , já que $\Gamma \subseteq \Gamma^*$. Acontece que garantir (a) e (b) é suficiente. Um conjunto de sentenças para o qual (b) vale é chamado de *completo*. Assim, nossa tarefa será estender o conjunto consistente Γ a um conjunto consistente e completo Γ^* .

Há um problema neste plano: se $\exists x \varphi(x) \in \Gamma$, esperaríamos poder escolher algum *símbolo de constante* c e adicionar $\varphi(c)$ neste processo. Mas como sabemos que sempre podemos fazer isso? Talvez tenhamos apenas alguns *símbolos de constante* em nossa linguagem, e, para cada um deles, tenhamos $\neg\varphi(c) \in \Gamma$. Também não podemos adicionar $\varphi(c)$, já que isso tornaria o conjunto inconsistente, e não saberíamos se $\mathfrak{M}$ tem que tornar $\varphi(c)$ ou $\neg\varphi(c)$ verdadeira. Além disso, pode acontecer de Γ conter apenas sentenças em uma linguagem que não tem símbolo de constante algum (por exemplo, a linguagem da teoria dos conjuntos).

A solução para este problema é simplesmente adicionar infinitas constantes no começo, além de sentenças que as conectam com os quantificadores da maneira correta. (Claro, temos que verificar que isso não pode introduzir uma inconsistência.)

Nossa construção original funciona bem se tivermos apenas *símbolos de constante* nas sentenças atômicas. Mas a linguagem também pode conter *símbolos de função*. Nesse caso, pode ser complicado encontrar as funções certas em $\mathbb{N}$ para atribuir a esses *símbolos de função* de modo a fazer tudo funcionar. Então, aqui vai outro truque: em vez de usar i para interpretar c_i , basta tomar o próprio conjunto de *símbolos de constante* como o domínio. Então $\mathfrak{M}$ pode atribuir cada *símbolo de constante* a si mesmo: $c_i^{\mathfrak{M}} = c_i$. Mas por que não ir até o fim: seja $|\mathfrak{M}|$ o conjunto de todos os *termos* da linguagem! Se fizermos isso, há uma atribuição óbvia de funções (que tomam termos como argumentos e têm termos como valores) aos *símbolos de função*: atribuímos ao *símbolo de função* f_i^n a função que, dados n termos $t_1, \dots, t_n$ como entrada, produz o termo $f_i^n(t_1, \dots, t_n)$ como valor.

A última peça do quebra-cabeça é o que fazer com $=$. O *símbolo de predicado* $=$ tem uma interpretação fixa: $\mathfrak{M} \models t = t'$ se e somente se $\text{Val}^{\mathfrak{M}}(t) = \text{Val}^{\mathfrak{M}}(t')$. Agora, se ajustarmos as coisas de modo que o *valor* de um termo t seja o próprio t , então esta *estrutura* não tornará verdadeira *nenhuma* sentença da forma $t = t'$ a menos que t e t' sejam um único e mesmo termo. E, claro, isso é um problema, já que basicamente toda teoria interessante em uma linguagem com *símbolos de função* terá como teoremas sentenças $t = t'$ onde t e t' não são o mesmo termo (por exemplo, em teorias da aritmética: $(0 + 0) = 0$). Para resolver este problema, mudamos o domínio de $\mathfrak{M}$: em vez de usar termos como os objetos em $|\mathfrak{M}|$, usamos conjuntos de termos, e cada conjunto é tal que contém todos aqueles termos que as sentenças em Γ exigem que sejam iguais. Assim, por exemplo, se Γ é uma teoria da aritmética, um desses conjuntos conterá: 0 , $(0 + 0)$, (0×0) , etc. Este será o conjunto que atribuímos a 0 , e acontecerá que este conjunto também é o valor de todos os termos nele, por exemplo, também de $(0 + 0)$. Portanto, a sentença $(0 + 0) = 0$ será verdadeira nesta *estrutura* revisada.

23.3. CONJUNTOS CONSISTENTES COMPLETOS DE SENTENÇAS

Então, aqui está o que faremos. Primeiro investigamos as propriedades dos conjuntos consistentes **completos**; em particular, provamos que um conjunto consistente **completo** contém $\varphi \wedge \psi$ se e somente se contém tanto φ quanto ψ , contém $\varphi \vee \psi$ se e somente se contém pelo menos um deles, etc. (**Proposição 23.2**). Em seguida, definimos e investigamos conjuntos “saturados” de sentenças. Um conjunto saturado é aquele que contém condicionais que ligam cada **sentença** quantificada às suas instâncias (**Definição 23.5**). Mostramos que qualquer conjunto consistente Γ sempre pode ser estendido a um conjunto saturado Γ' (**Lema 23.6**). Se um conjunto é consistente, saturado e **completo**, ele também tem a propriedade de conter $\exists x \varphi(x)$ se e somente se contém $\varphi(t)$ para algum termo fechado t e $\forall x \varphi(x)$ se e somente se contém $\varphi(t)$ para todo termo fechado t (**Proposição 23.7**). Em seguida, tomaremos o conjunto consistente saturado Γ' e mostraremos que ele pode ser estendido a um conjunto saturado, consistente e **completo** Γ^* (**Lema 23.8**). Este conjunto Γ^* é o que usaremos para definir nosso modelo de termos $\mathfrak{M}(\Gamma^*)$. O modelo de termos tem o conjunto de termos fechados como seu domínio, e a interpretação de seus **símbolos de predicado** é dada pelas **sentenças** atômicas em Γ^* (**Definição 23.9**). Usaremos as propriedades dos conjuntos consistentes completos e saturados para mostrar que, de fato, $\mathfrak{M}(\Gamma^*) \models \varphi$ se e somente se $\varphi \in \Gamma^*$ (**Lema 23.12**) e, portanto, em particular, $\mathfrak{M}(\Gamma^*) \models \Gamma$. Por fim, consideraremos como definir um modelo de termos se Γ contém = também (**Definição 23.16**) e mostraremos que ele satisfaz Γ^* (**Lema 23.19**).

content/first-order-logic/completeness/complete-consistent-sets.tex

23.3 Conjuntos Consistentes Completos de Sentenças

fol:com:ccs:
sec
fol:com:ccs:
def:complete-set

Definição 23.1 (Conjunto completo). Um conjunto Γ de **sentenças** é **completo** se e somente se, para qualquer **sentença** φ , ou $\varphi \in \Gamma$ ou $\neg\varphi \in \Gamma$.

Conjuntos **completos** de sentenças não deixam questões sem resposta. Para qualquer **sentença** φ , Γ “diz” se φ é verdadeira ou falsa. A importância dos conjuntos **completos** estende-se para além da prova do teorema da completude. Uma teoria que é completa e axiomatizável, por exemplo, é sempre decidível. explanation

Conjuntos consistentes **completos** são importantes na prova de completude já que podemos garantir que todo conjunto consistente de **sentenças** Γ está contido em um conjunto consistente **completo** Γ^* . Um conjunto consistente **completo** contém, para cada **sentença** φ , ou φ ou sua negação $\neg\varphi$, mas não ambas. Isso é verdade em particular para **sentenças** atômicas, de modo que, a partir de um conjunto consistente **completo** em uma linguagem adequadamente expandida por **símbolos de constante**, podemos construir **a estrutura** onde a interpretação dos **símbolos de predicado** é definida de acordo com quais **sentenças** atômicas estão em Γ^* . Pode-se então mostrar que esta **estrutura** torna verdadeiras todas as **sentenças** em Γ^* (e, portanto, também todas aque- explanation

las em Γ). A prova deste último fato requer que $\neg\varphi \in \Gamma^*$ se e somente se $\varphi \notin \Gamma^*$, $(\varphi \vee \psi) \in \Gamma^*$ se e somente se $\varphi \in \Gamma^*$ ou $\psi \in \Gamma^*$, etc.

No que se segue, usaremos com frequência, de modo tácito, as propriedades de reflexividade, monotonicidade e transitividade de $\vdash$ (ver Seções 19.8, 20.7, 21.7 e 22.6).

Proposição 23.2. *Suponha que Γ seja completo e consistente. Então:*

1. Se $\Gamma \vdash \varphi$, então $\varphi \in \Gamma$.
2. $\varphi \wedge \psi \in \Gamma$ se e somente se $\varphi \in \Gamma$ e $\psi \in \Gamma$.
3. $\varphi \vee \psi \in \Gamma$ se e somente se $\varphi \in \Gamma$ ou $\psi \in \Gamma$.
4. $\varphi \rightarrow \psi \in \Gamma$ se e somente se $\varphi \notin \Gamma$ ou $\psi \in \Gamma$.

*fol:com:ccs:
prop:ccs

fol:com:ccs:
prop:ccs-prov-in

fol:com:ccs:
prop:ccs-and

fol:com:ccs:
prop:ccs-or

fol:com:ccs:
prop:ccs-if*

Prova. Suponhamos, para tudo o que se segue, que Γ seja completo e consistente.

1. Se $\Gamma \vdash \varphi$, então $\varphi \in \Gamma$.

Suponha que $\Gamma \vdash \varphi$. Suponha, ao contrário, que $\varphi \notin \Gamma$. Como Γ é completo, $\neg\varphi \in \Gamma$. Por Proposições 19.20, 20.20, 21.20 e 22.28, Γ é inconsistente. Isso contradiz a suposição de que Γ é consistente. Logo, não pode ser o caso que $\varphi \notin \Gamma$, de modo que $\varphi \in \Gamma$.

2. $\varphi \wedge \psi \in \Gamma$ se e somente se $\varphi \in \Gamma$ e $\psi \in \Gamma$:

Para a direção direta, suponha $\varphi \wedge \psi \in \Gamma$. Então por Proposições 19.22, 20.22, 21.22 e 22.30, item (1), $\Gamma \vdash \varphi$ e $\Gamma \vdash \psi$. Por (1), $\varphi \in \Gamma$ e $\psi \in \Gamma$, como requerido.

Para a direção reversa, sejam $\varphi \in \Gamma$ e $\psi \in \Gamma$. Por Proposições 19.22, 20.22, 21.22 e 22.30, item (2), $\Gamma \vdash \varphi \wedge \psi$. Por (1), $\varphi \wedge \psi \in \Gamma$.

3. Primeiro mostramos que, se $\varphi \vee \psi \in \Gamma$, então $\varphi \in \Gamma$ ou $\psi \in \Gamma$. Suponha $\varphi \vee \psi \in \Gamma$ mas $\varphi \notin \Gamma$ e $\psi \notin \Gamma$. Como Γ é completo, $\neg\varphi \in \Gamma$ e $\neg\psi \in \Gamma$. Por Proposições 19.23, 20.23, 21.23 e 22.31, item (1), Γ é inconsistente, uma contradição. Logo, $\varphi \in \Gamma$ ou $\psi \in \Gamma$.

Para a direção reversa, suponha que $\varphi \in \Gamma$ ou $\psi \in \Gamma$. Por Proposições 19.23, 20.23, 21.23 e 22.31, item (2), $\Gamma \vdash \varphi \vee \psi$. Por (1), $\varphi \vee \psi \in \Gamma$, como requerido.

4. Para a direção direta, suponha $\varphi \rightarrow \psi \in \Gamma$ e suponha, ao contrário, que $\varphi \in \Gamma$ e $\psi \notin \Gamma$. Sob essas suposições, $\varphi \rightarrow \psi \in \Gamma$ e $\varphi \in \Gamma$. Por Proposições 19.24, 20.24, 21.24 e 22.32, item (1), $\Gamma \vdash \psi$. Mas então, por (1), $\psi \in \Gamma$, contradizendo a suposição de que $\psi \notin \Gamma$.

Para a direção reversa, considere primeiro o caso em que $\varphi \notin \Gamma$. Como Γ é completo, $\neg\varphi \in \Gamma$. Por Proposições 19.24, 20.24, 21.24 e 22.32, item (2), $\Gamma \vdash \varphi \rightarrow \psi$. Novamente por (1), obtemos que $\varphi \rightarrow \psi \in \Gamma$, como requerido.

23.4. EXPANSÃO DE HENKIN

Agora considere o caso em que $\psi \in \Gamma$. Por [Proposições 19.24, 20.24, 21.24 e 22.32](#), item (2) novamente, $\Gamma \vdash \varphi \rightarrow \psi$. Por [\(1\)](#), $\varphi \rightarrow \psi \in \Gamma$. $\square$

Problema 23.1. Complete a prova de [Proposição 23.2](#).

[content/first-order-logic/completeness/henkin-expansions.tex](#)

23.4 Expansão de Henkin

[fol:com:hen:](#)
[sec](#)

Parte do desafio em provar o teorema da completude é que o modelo que construímos a partir de um conjunto consistente completo Γ deve tornar verdadeiras todas as [fórmulas](#) quantificadas em Γ . Para garantir isso, usamos um truque devido a Leon Henkin. Em essência, o truque consiste em expandir a linguagem por infinitos [símbolos de constante](#) e adicionar, para cada [fórmula](#) com uma [variável](#) livre $\varphi(x)$, uma fórmula da forma $\exists x \varphi(x) \rightarrow \varphi(c)$, onde c é um dos novos [símbolos de constante](#). Quando construímos a [estrutura](#) que satisfaz Γ , isso garantirá que cada sentença existencial verdadeira tenha uma testemunha entre as novas constantes.

[explanation](#)

[fol:com:hen:](#)
[prop:lang-exp](#)

Proposição 23.3. *Se Γ é consistente em $\mathcal{L}$ e $\mathcal{L}'$ é obtida de $\mathcal{L}$ adicionando-se [a infinito enumerável](#) conjunto de novos [símbolos de constante](#) $d_0, d_1, \dots$, então Γ é consistente em $\mathcal{L}'$.*

Definição 23.4 (Conjunto saturado). Um conjunto Γ de [fórmulas](#) de uma linguagem $\mathcal{L}$ é *saturado* se e somente se, para cada [fórmula](#) $\varphi(x) \in \text{Frm}(\mathcal{L})$ com uma [variável](#) livre x , existe [a símbolo de constante](#) $c \in \mathcal{L}$ tal que $\exists x \varphi(x) \rightarrow \varphi(c) \in \Gamma$.

A definição a seguir será usada na prova do próximo teorema.

[fol:com:hen:](#)
[defn:henkin-exp](#)

Definição 23.5. Seja $\mathcal{L}'$ como em [Proposição 23.3](#). Fixe uma enumeração $\varphi_0(x_0), \varphi_1(x_1), \dots$ de todas as [fórmulas](#) $\varphi_i(x_i)$ de $\mathcal{L}'$ nas quais uma variável (x_i) ocorre livre. Definimos as [sentenças](#) θ_n por indução sobre n .

Seja c_0 o primeiro [símbolo de constante](#) dentre os d_i que adicionamos a $\mathcal{L}$ que não ocorre em $\varphi_0(x_0)$. Supondo que $\theta_0, \dots, \theta_{n-1}$ já tenham sido definidas, seja c_n o primeiro dentre os novos [símbolos de constante](#) d_i que não ocorre nem em $\theta_0, \dots, \theta_{n-1}$ nem em $\varphi_n(x_n)$.

Agora seja θ_n a [fórmula](#) $\exists x_n \varphi_n(x_n) \rightarrow \varphi_n(c_n)$.

[fol:com:hen:](#)
[lem:henkin](#)

Lema 23.6. *Todo conjunto consistente Γ pode ser estendido a um conjunto saturado consistente Γ' .*

Prova. Dado um conjunto consistente de sentenças Γ em uma linguagem $\mathcal{L}$, expanda a linguagem adicionando [a infinito enumerável](#) conjunto de novos

símbolos de constante para formar $\mathcal{L}'$. Por [Proposição 23.3](#), Γ ainda é consistente na linguagem mais rica. Além disso, seja θ_i como em [Definição 23.5](#). Seja

$$\begin{aligned}\Gamma_0 &= \Gamma \\ \Gamma_{n+1} &= \Gamma_n \cup \{\theta_n\}\end{aligned}$$

isto é, $\Gamma_{n+1} = \Gamma \cup \{\theta_0, \dots, \theta_n\}$, e seja $\Gamma' = \bigcup_n \Gamma_n$. Γ' é claramente saturado.

Se Γ' fosse inconsistente, então, para algum n , Γ_n seria inconsistente (Exercício: explique por quê). Portanto, para mostrar que Γ' é consistente, basta mostrar, por indução sobre n , que cada conjunto Γ_n é consistente.

A base da indução é simplesmente a afirmação de que $\Gamma_0 = \Gamma$ é consistente, que é a hipótese do teorema. Para o passo de indução, suponha que Γ_n seja consistente mas $\Gamma_{n+1} = \Gamma_n \cup \{\theta_n\}$ seja inconsistente. Lembre que θ_n é $\exists x_n \varphi_n(x_n) \rightarrow \varphi_n(c_n)$, onde $\varphi_n(x_n)$ é a fórmula de $\mathcal{L}'$ com apenas a variável x_n livre. Pela maneira como escolhemos os c_n (ver [Definição 23.5](#)), c_n não ocorre em $\varphi_n(x_n)$ nem em Γ_n .

Se $\Gamma_n \cup \{\theta_n\}$ é inconsistente, então $\Gamma_n \vdash \neg\theta_n$ e, portanto, ambos os seguintes valem:

$$\Gamma_n \vdash \exists x_n \varphi_n(x_n) \quad \Gamma_n \vdash \neg\varphi_n(c_n)$$

Como c_n não ocorre em Γ_n ou em $\varphi_n(x_n)$, [Teoremas 19.25, 20.25, 21.25 e 22.33](#) se aplica. De $\Gamma_n \vdash \neg\varphi_n(c_n)$, obtemos $\Gamma_n \vdash \forall x_n \neg\varphi_n(x_n)$. Assim, temos que ambos $\Gamma_n \vdash \exists x_n \varphi_n(x_n)$ e $\Gamma_n \vdash \forall x_n \neg\varphi_n(x_n)$, de modo que o próprio Γ_n é inconsistente. (Note que $\forall x_n \neg\varphi_n(x_n) \vdash \neg\exists x_n \varphi_n(x_n)$.) Contradição: suponha-se que Γ_n fosse consistente. Logo, $\Gamma_n \cup \{\theta_n\}$ é consistente. $\square$

explanation

Mostraremos agora que conjuntos *completos*, consistentes que são saturados têm a propriedade de que contêm uma *sentença* universalmente quantificada se e somente se contêm todas as suas instâncias e contêm uma *sentença* existencialmente quantificada se e somente se contêm pelo menos uma instância. Usaremos isso para mostrar que a *estrutura* que geraremos a partir de um conjunto completo, consistente, saturado torna todas as suas sentenças quantificadas verdadeiras.

Proposição 23.7. *Suponha que Γ seja completo, consistente e saturado.*

fol:com:hen:
prop:saturated-instances

1. $\exists x \varphi(x) \in \Gamma$ se e somente se $\varphi(t) \in \Gamma$ para pelo menos um termo fechado t .
2. $\forall x \varphi(x) \in \Gamma$ se e somente se $\varphi(t) \in \Gamma$ para todo termo fechado t .

Prova. 1. Suponha primeiro que $\exists x \varphi(x) \in \Gamma$. Como Γ é saturado, $(\exists x \varphi(x) \rightarrow \varphi(c)) \in \Gamma$ para algum *símbolo de constante* c . Por [Proposições 19.24, 20.24, 21.24 e 22.32](#), item (1), e [Proposição 23.2\(1\)](#), $\varphi(c) \in \Gamma$.

Para a outra direção, a saturação não é necessária: Suponha $\varphi(t) \in \Gamma$. Então $\Gamma \vdash \exists x \varphi(x)$ por [Proposições 19.26, 20.26, 21.26 e 22.34](#), item (1). Por [Proposição 23.2\(1\)](#), $\exists x \varphi(x) \in \Gamma$.

23.5. LEMA DE LINDENBAUM

2. Suponha que $\varphi(t) \in \Gamma$ para todo termo fechado t . Por contradição, assumamos $\forall x \varphi(x) \notin \Gamma$. Como Γ é completo, $\neg \forall x \varphi(x) \in \Gamma$. Por saturação, $(\exists x \neg \varphi(x) \rightarrow \neg \varphi(c)) \in \Gamma$ para algum **símbolo de constante** c . Por hipótese, como c é um termo fechado, $\varphi(c) \in \Gamma$. Mas isso tornaria Γ inconsistente. (Exercício: dê a **derivação** que mostra que

$$\neg \forall x \varphi(x), \exists x \neg \varphi(x) \rightarrow \neg \varphi(c), \varphi(c)$$

é inconsistente.)

Para a direção reversa, não precisamos de saturação: Suponha $\forall x \varphi(x) \in \Gamma$. Então $\Gamma \vdash \varphi(t)$ por **Proposições 19.26, 20.26, 21.26 e 22.34**, item (2). Obtemos $\varphi(t) \in \Gamma$ por **Proposição 23.2**. $\square$

content/first-order-logic/completeness/lindenbaums-lemma.tex

23.5 Lema de Lindenbaum

fol:com:lin:
sec

Provamos agora um lema que mostra que qualquer conjunto consistente de **sentenças** está contido em algum conjunto de sentenças que não é apenas consistente, mas também **completo**. A prova funciona adicionando uma **sentença** de cada vez, garantindo a cada passo que o conjunto permanece consistente. Fazemos isso de modo que, para todo φ , ou φ ou $\neg \varphi$ seja adicionada em algum estágio. A união de todos os estágios dessa construção então contém ou φ ou sua negação $\neg \varphi$ e é, portanto, completa. Ela também é consistente, já que nos asseguramos, a cada estágio, de não introduzir uma inconsistência.

explanation

fol:com:lin:
lem:lindenbaum

Lema 23.8 (Lema de Lindenbaum). *Todo conjunto consistente Γ em uma linguagem $\mathcal{L}$ pode ser estendido a um conjunto consistente **completo** Γ^* .*

Prova. Seja Γ consistente. Sejam $\varphi_0, \varphi_1, \dots$ uma enumeração de todas as **sentenças** de $\mathcal{L}$. Defina $\Gamma_0 = \Gamma$, e

$$\Gamma_{n+1} = \begin{cases} \Gamma_n \cup \{\varphi_n\} & \text{se } \Gamma_n \cup \{\varphi_n\} \text{ é consistente;} \\ \Gamma_n \cup \{\neg \varphi_n\} & \text{caso contrário.} \end{cases}$$

Seja $\Gamma^* = \bigcup_{n \geq 0} \Gamma_n$.

Cada Γ_n é consistente: Γ_0 é consistente por definição. Se $\Gamma_{n+1} = \Gamma_n \cup \{\varphi_n\}$, isso é porque este último é consistente. Se não é, $\Gamma_{n+1} = \Gamma_n \cup \{\neg \varphi_n\}$. Temos que verificar que $\Gamma_n \cup \{\neg \varphi_n\}$ é consistente. Suponha que não seja. Então **ambos** $\Gamma_n \cup \{\varphi_n\}$ e $\Gamma_n \cup \{\neg \varphi_n\}$ são inconsistentes. Isso significa que Γ_n seria inconsistente por **Proposições 19.21, 20.21, 21.21 e 22.29**, contrariando a hipótese de indução.

Para todo n e todo $i < n$, $\Gamma_i \subseteq \Gamma_n$. Isso segue por uma simples indução sobre n . Para $n = 0$, não há $i < 0$, de modo que a afirmação vale automaticamente. Para o passo indutivo, suponha que ela seja verdadeira para n .

Mostramos que, se $i < n + 1$, então $\Gamma_i \subseteq \Gamma_{n+1}$. Temos $\Gamma_{n+1} = \Gamma_n \cup \{\varphi_n\}$ ou $= \Gamma_n \cup \{\neg\varphi_n\}$ por construção. Logo $\Gamma_n \subseteq \Gamma_{n+1}$. Se $i < n + 1$, então $\Gamma_i \subseteq \Gamma_n$ por hipótese indutiva (se $i < n$) ou pelo fato trivial de que $\Gamma_n \subseteq \Gamma_n$ (se $i = n$). Obtemos que $\Gamma_i \subseteq \Gamma_{n+1}$ por transitividade de $\subseteq$.

Disso segue que Γ^* é consistente. Eis o porquê: Seja $\Gamma' \subseteq \Gamma^*$ finito. Cada $\psi \in \Gamma'$ também está em Γ_i para algum i . Seja n o maior destes. Como $\Gamma_i \subseteq \Gamma_n$ se $i \leq n$, todo $\psi \in \Gamma'$ também está em Γ_n , isto é, $\Gamma' \subseteq \Gamma_n$, e Γ_n é consistente. Assim, todo subconjunto finito $\Gamma' \subseteq \Gamma^*$ é consistente. Por **Proposições 19.17, 20.17, 21.17 e 22.21**, Γ^* é consistente.

Toda **sentença** de $\text{Frm}(\mathcal{L})$ aparece na lista usada para definir Γ^* . Se $\varphi_n \notin \Gamma^*$, então isso é porque $\Gamma_n \cup \{\varphi_n\}$ era inconsistente. Mas então $\neg\varphi_n \in \Gamma^*$, de modo que Γ^* é **completo**. $\square$

`content/first-order-logic/completeness/construction-of-model.tex`

23.6 Construção de um Modelo

explanation

Por ora, não nos preocupamos com $=$, isto é, queremos apenas mostrar que um conjunto consistente Γ de **sentenças** que não contém $=$ é satisfatível. Primeiro estendemos Γ a um conjunto consistente, **completo** e saturado Γ^* . Nesse caso, a definição de um modelo $\mathfrak{M}(\Gamma^*)$ é simples: Tomamos o conjunto de termos fechados de $\mathcal{L}'$ como o domínio. Atribuímos cada **símbolo de constante** a si mesmo e nos asseguramos de que, mais geralmente, para todo termo fechado t , $\text{Val}^{\mathfrak{M}(\Gamma^*)}(t) = t$. Aos **símbolos de predicado** são atribuídas extensões de tal forma que uma **sentença** atômica é verdadeira em $\mathfrak{M}(\Gamma^*)$ se e somente se está em Γ^* . Isso obviamente tornará verdadeiras em $\mathfrak{M}(\Gamma^*)$ todas as **sentenças** atômicas em Γ^* . As demais são verdadeiras desde que o Γ^* com que começamos seja consistente, completo e saturado.

fol:com:mod:sec

Definição 23.9 (Modelo de termos). Seja Γ^* um conjunto consistente, **completo** e saturado de **sentenças** em uma linguagem $\mathcal{L}$. O *modelo de termos* $\mathfrak{M}(\Gamma^*)$ de Γ^* é a **estrutura** definida como segue:

fol:com:mod:defn:termmodel

1. O **domínio** $|\mathfrak{M}(\Gamma^*)|$ é o conjunto de todos os termos fechados de $\mathcal{L}$.
2. A interpretação de **a símbolo de constante** c é o próprio c : $c^{\mathfrak{M}(\Gamma^*)} = c$.
3. Ao **símbolo de função** f é atribuída a função que, dados como argumentos os termos fechados $t_1, \dots, t_n$, tem como valor o termo fechado $f(t_1, \dots, t_n)$:

$$f^{\mathfrak{M}(\Gamma^*)}(t_1, \dots, t_n) = f(t_1, \dots, t_n)$$

4. Se R é um **símbolo de predicado** de n lugares, então

$$\langle t_1, \dots, t_n \rangle \in R^{\mathfrak{M}(\Gamma^*)} \text{ se e somente se } R(t_1, \dots, t_n) \in \Gamma^*.$$

Verificaremos agora que de fato temos $\text{Val}^{\mathfrak{M}(\Gamma^*)}(t) = t$.

23.6. CONSTRUÇÃO DE UM MODELO

fol:com:mod: lem:val-in-termmodel **Lema 23.10.** *Seja $\mathfrak{M}(\Gamma^*)$ o modelo de termos de Definição 23.9; então $\text{Val}^{\mathfrak{M}(\Gamma^*)}(t) = t$.*

Prova. A prova é por indução sobre t , onde o caso base, quando t é a **símbolo de constante**, segue diretamente da definição do modelo de termos. Para o passo de indução, assumamos que $t_1, \dots, t_n$ são termos fechados tais que $\text{Val}^{\mathfrak{M}(\Gamma^*)}(t_i) = t_i$ e que f é a **símbolo de função** n -ária. Então

$$\begin{aligned} \text{Val}^{\mathfrak{M}(\Gamma^*)}(f(t_1, \dots, t_n)) &= f^{\mathfrak{M}(\Gamma^*)}(\text{Val}^{\mathfrak{M}(\Gamma^*)}(t_1), \dots, \text{Val}^{\mathfrak{M}(\Gamma^*)}(t_n)) \\ &= f^{\mathfrak{M}(\Gamma^*)}(t_1, \dots, t_n) \\ &= f(t_1, \dots, t_n), \end{aligned}$$

e, assim, por indução, isto vale para todo termo fechado t . $\square$

A estrutura $\mathfrak{M}$ pode tornar verdadeira uma **sentença** $\exists x \varphi(x)$ existencialmente quantificada sem que haja uma instância $\varphi(t)$ que ela torne verdadeira. A **estrutura** $\mathfrak{M}$ pode tornar verdadeiras todas as instâncias $\varphi(t)$ de uma **sentença** $\forall x \varphi(x)$ universalmente quantificada sem tornar $\forall x \varphi(x)$ verdadeira. Isso ocorre porque, em geral, nem todo **membro** de $|\mathfrak{M}|$ é o valor de um termo fechado ($\mathfrak{M}$ pode não ser coberta). Esta é a razão pela qual a relação de satisfação é definida por meio de atribuições de variáveis. Entretanto, para o nosso modelo de termos $\mathfrak{M}(\Gamma^*)$ isso não seria necessário — porque ela é coberta. Este é o conteúdo do próximo resultado. explanation

fol:com:mod: prop:quant-termmodel **Proposição 23.11.** *Seja $\mathfrak{M}(\Gamma^*)$ o modelo de termos de Definição 23.9.*

1. $\mathfrak{M}(\Gamma^*) \models \exists x \varphi(x)$ se e somente se $\mathfrak{M}(\Gamma^*) \models \varphi(t)$ para pelo menos um termo fechado t .
2. $\mathfrak{M}(\Gamma^*) \models \forall x \varphi(x)$ se e somente se $\mathfrak{M}(\Gamma^*) \models \varphi(t)$ para todo termo fechado t .

Prova. 1. Por **Proposição 16.19**, $\mathfrak{M}(\Gamma^*) \models \exists x \varphi(x)$ se e somente se, para pelo menos uma atribuição de variáveis s , $\mathfrak{M}(\Gamma^*), s \models \varphi(x)$. Como $|\mathfrak{M}(\Gamma^*)|$ consiste nos termos fechados de $\mathcal{L}$, este é o caso se e somente se há pelo menos um termo fechado t tal que $s(x) = t$ e $\mathfrak{M}(\Gamma^*), s \models \varphi(x)$. Por **Proposição 16.23**, $\mathfrak{M}(\Gamma^*), s \models \varphi(x)$ se e somente se $\mathfrak{M}(\Gamma^*), s \models \varphi(t)$, onde $s(x) = t$. Por **Proposição 16.18**, $\mathfrak{M}(\Gamma^*), s \models \varphi(t)$ se e somente se $\mathfrak{M}(\Gamma^*) \models \varphi(t)$, já que $\varphi(t)$ é uma sentença.

2. Por **Proposição 16.19**, $\mathfrak{M}(\Gamma^*) \models \forall x \varphi(x)$ se e somente se, para toda atribuição de variáveis s , $\mathfrak{M}(\Gamma^*), s \models \varphi(x)$. Lembre que $|\mathfrak{M}(\Gamma^*)|$ consiste nos termos fechados de $\mathcal{L}$, de modo que, para todo termo fechado t , $s(x) = t$ é uma tal atribuição de variáveis, e, para qualquer atribuição de variáveis, $s(x)$ é algum termo fechado t . Por **Proposição 16.23**, $\mathfrak{M}(\Gamma^*), s \models \varphi(x)$ se e somente se $\mathfrak{M}(\Gamma^*), s \models \varphi(t)$, onde $s(x) = t$. Por **Proposição 16.18**, $\mathfrak{M}(\Gamma^*), s \models \varphi(t)$ se e somente se $\mathfrak{M}(\Gamma^*) \models \varphi(t)$, já que $\varphi(t)$ é uma sentença. $\square$

Lema 23.12 (Lema da Verdade). *Suponha que φ não contenha $=$. Então $\mathfrak{M}(I^*) \models \varphi$ se e somente se $\varphi \in I^*$.* fol:com:mod:
lem:truth

Prova. Provamos ambas as direções simultaneamente e por indução sobre φ .

1. $\varphi \equiv \perp$: $\mathfrak{M}(I^*) \not\models \perp$ por definição de satisfação. Por outro lado, $\perp \notin I^*$ já que I^* é consistente.
2. $\varphi \equiv \top$: $\mathfrak{M}(I^*) \models \top$ por definição de satisfação. Por outro lado, $\top \in I^*$ já que I^* é consistente e **completo**, e $I^* \vdash \top$.
3. $\varphi \equiv R(t_1, \dots, t_n)$: $\mathfrak{M}(I^*) \models R(t_1, \dots, t_n)$ se e somente se $\langle t_1, \dots, t_n \rangle \in R^{\mathfrak{M}(I^*)}$ (pela definição de satisfação) se e somente se $R(t_1, \dots, t_n) \in I^*$ (pela construção de $\mathfrak{M}(I^*)$).
4. $\varphi \equiv \neg\psi$: $\mathfrak{M}(I^*) \models \varphi$ se e somente se $\mathfrak{M}(I^*) \not\models \psi$ (por definição de satisfação). Por hipótese de indução, $\mathfrak{M}(I^*) \not\models \psi$ se e somente se $\psi \notin I^*$. Como I^* é consistente e **completo**, $\psi \notin I^*$ se e somente se $\neg\psi \in I^*$.
5. $\varphi \equiv \psi \wedge \chi$: $\mathfrak{M}(I^*) \models \varphi$ se e somente se temos ambos $\mathfrak{M}(I^*) \models \psi$ e $\mathfrak{M}(I^*) \models \chi$ (por definição de satisfação) se e somente se $\psi \in I^*$ e $\chi \in I^*$ (pela hipótese de indução). Por **Proposição 23.2(2)**, este é o caso se e somente se $(\psi \wedge \chi) \in I^*$.
6. $\varphi \equiv \psi \vee \chi$: $\mathfrak{M}(I^*) \models \varphi$ se e somente se $\mathfrak{M}(I^*) \models \psi$ ou $\mathfrak{M}(I^*) \models \chi$ (por definição de satisfação) se e somente se $\psi \in I^*$ ou $\chi \in I^*$ (por hipótese de indução). Este é o caso se e somente se $(\psi \vee \chi) \in I^*$ (por **Proposição 23.2(3)**).
7. $\varphi \equiv \psi \rightarrow \chi$: $\mathfrak{M}(I^*) \models \varphi$ se e somente se $\mathfrak{M}(I^*) \not\models \psi$ ou $\mathfrak{M}(I^*) \models \chi$ (por definição de satisfação) se e somente se $\psi \notin I^*$ ou $\chi \in I^*$ (por hipótese de indução). Este é o caso se e somente se $(\psi \rightarrow \chi) \in I^*$ (por **Proposição 23.2(4)**).
8. $\varphi \equiv \forall x \psi(x)$: $\mathfrak{M}(I^*) \models \varphi$ se e somente se $\mathfrak{M}(I^*) \models \psi(t)$ para todos os termos t (**Proposição 23.11**). Por hipótese de indução, este é o caso se e somente se $\psi(t) \in I^*$ para todos os termos t ; por **Proposição 23.7**, isto, por sua vez, é o caso se e somente se $\forall x \psi(x) \in I^*$.
9. $\varphi \equiv \exists x \psi(x)$: $\mathfrak{M}(I^*) \models \varphi$ se e somente se $\mathfrak{M}(I^*) \models \psi(t)$ para pelo menos um termo t (**Proposição 23.11**). Por hipótese de indução, este é o caso se e somente se $\psi(t) \in I^*$ para pelo menos um termo t . Por **Proposição 23.7**, isto, por sua vez, é o caso se e somente se $\exists x \psi(x) \in I^*$. □

23.7 Identidade

fol:com:ide: sec A construção do modelo de termos dada na seção anterior é suficiente para explanation estabelecer a completude da lógica de primeira ordem para conjuntos Γ que não contêm $=$. O modelo de termos satisfaz todo $\varphi \in \Gamma^*$ que não contém $=$ (e, portanto, todo $\varphi \in \Gamma$). Ele não funciona, no entanto, se $=$ está presente. A razão é que Γ^* pode então conter **a sentença** $t = t'$, mas no modelo de termos o valor de qualquer termo é esse próprio termo. Logo, se t e t' são termos diferentes, seus valores no modelo de termos — isto é, t e t' , respectivamente — são diferentes, e assim $t = t'$ é falsa. Podemos corrigir isso, no entanto, usando uma construção conhecida como “fatoração”.

Definição 23.13. Seja Γ^* um conjunto consistente e **completo** de sentenças em $\mathcal{L}$. Definimos a relação $\approx$ sobre o conjunto de termos fechados de $\mathcal{L}$ por

$$t \approx t' \quad \text{se e somente se} \quad t = t' \in \Gamma^*$$

fol:com:ide: prop:approx-equiv **Proposição 23.14.** A relação $\approx$ tem as seguintes propriedades:

1. $\approx$ é reflexiva.
2. $\approx$ é simétrica.
3. $\approx$ é transitiva.
4. Se $t \approx t'$, f é **a símbolo de função**, e $t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n$ são termos fechados, então

$$f(t_1, \dots, t_{i-1}, t, t_{i+1}, \dots, t_n) \approx f(t_1, \dots, t_{i-1}, t', t_{i+1}, \dots, t_n).$$

5. Se $t \approx t'$, R é **a símbolo de predicado**, e $t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n$ são termos fechados, então

$$R(t_1, \dots, t_{i-1}, t, t_{i+1}, \dots, t_n) \in \Gamma^* \quad \text{se e somente se} \quad R(t_1, \dots, t_{i-1}, t', t_{i+1}, \dots, t_n) \in \Gamma^*.$$

Prova. Como Γ^* é consistente e **completo**, $t = t' \in \Gamma^*$ se e somente se $\Gamma^* \vdash t = t'$. Assim, basta mostrar o seguinte:

1. $\Gamma^* \vdash t = t$ para todos os termos fechados t .
2. Se $\Gamma^* \vdash t = t'$ então $\Gamma^* \vdash t' = t$.
3. Se $\Gamma^* \vdash t = t'$ e $\Gamma^* \vdash t' = t''$, então $\Gamma^* \vdash t = t''$.
4. Se $\Gamma^* \vdash t = t'$, então

$$\Gamma^* \vdash f(t_1, \dots, t_{i-1}, t, t_{i+1}, \dots, t_n) = f(t_1, \dots, t_{i-1}, t', t_{i+1}, \dots, t_n)$$

para todo **símbolo de função** f de n lugares e termos fechados $t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n$.

5. Se $\Gamma^* \vdash t = t'$ e $\Gamma^* \vdash R(t_1, \dots, t_{i-1}, t, t_{i+1}, \dots, t_n)$, então $\Gamma^* \vdash R(t_1, \dots, t_{i-1}, t', t_{i+1}, \dots, t_n)$ para todo **símbolo de predicado** R de n lugares e termos fechados $t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n$. $\square$

Problema 23.2. Complete a prova de **Proposição 23.14**.

Definição 23.15. Suponha que Γ^* seja um conjunto consistente e **completo** em uma linguagem $\mathcal{L}$, t seja um termo fechado e $\approx$ como na definição anterior. Então:

$$[t]_{\approx} = \{t' : t' \in \text{Trm}(\mathcal{L}), t \approx t'\}$$

$$\text{e } \text{Trm}(\mathcal{L})/\approx = \{[t]_{\approx} : t \in \text{Trm}(\mathcal{L})\}.$$

Definição 23.16. Seja $\mathfrak{M} = \mathfrak{M}(\Gamma^*)$ o modelo de termos para Γ^* de **Definição 23.9**. Então $\mathfrak{M}/\approx$ é a seguinte **estrutura**:

fol:com:ide:
defn:term-model-factor

1. $|\mathfrak{M}/\approx| = \text{Trm}(\mathcal{L})/\approx$.
2. $c^{\mathfrak{M}/\approx} = [c]_{\approx}$
3. $f^{\mathfrak{M}/\approx}([t_1]_{\approx}, \dots, [t_n]_{\approx}) = [f(t_1, \dots, t_n)]_{\approx}$
4. $\langle [t_1]_{\approx}, \dots, [t_n]_{\approx} \rangle \in R^{\mathfrak{M}/\approx}$ se e somente se $\mathfrak{M} \models R(t_1, \dots, t_n)$, isto é, se e somente se $R(t_1, \dots, t_n) \in \Gamma^*$.

explanation

Note que definimos $f^{\mathfrak{M}/\approx}$ e $R^{\mathfrak{M}/\approx}$ para elementos de $\text{Trm}(\mathcal{L})/\approx$ referindo-nos a eles como $[t]_{\approx}$, isto é, por meio de *representantes* $t \in [t]_{\approx}$. Temos que nos assegurar de que essas definições não dependam da escolha desses representantes, isto é, de que, para algumas outras escolhas t' que determinam as mesmas classes de equivalência ($[t]_{\approx} = [t']_{\approx}$), as definições produzam o mesmo resultado. Por exemplo, se R é um **símbolo de predicado** de um lugar, a última cláusula da definição diz que $[t]_{\approx} \in R^{\mathfrak{M}/\approx}$ se e somente se $\mathfrak{M} \models R(t)$. Se, para algum outro termo t' com $t \approx t'$, $\mathfrak{M} \not\models R(t)$, então a definição exigiria $[t']_{\approx} \notin R^{\mathfrak{M}/\approx}$. Se $t \approx t'$, então $[t]_{\approx} = [t']_{\approx}$, mas não podemos ter ao mesmo tempo $[t]_{\approx} \in R^{\mathfrak{M}/\approx}$ e $[t]_{\approx} \notin R^{\mathfrak{M}/\approx}$. Entretanto, **Proposição 23.14** garante que isso não pode acontecer.

Proposição 23.17. $\mathfrak{M}/\approx$ está bem definida, isto é, se $t_1, \dots, t_n, t'_1, \dots, t'_n$ são termos fechados, e $t_i \approx t'_i$, então

1. $[f(t_1, \dots, t_n)]_{\approx} = [f(t'_1, \dots, t'_n)]_{\approx}$, isto é,

$$f(t_1, \dots, t_n) \approx f(t'_1, \dots, t'_n)$$

e

2. $\mathfrak{M} \models R(t_1, \dots, t_n)$ se e somente se $\mathfrak{M} \models R(t'_1, \dots, t'_n)$, isto é,

$$R(t_1, \dots, t_n) \in \Gamma^* \text{ se e somente se } R(t'_1, \dots, t'_n) \in \Gamma^*.$$

23.8. O TEOREMA DA COMPLETUDE

Prova. Segue de [Proposição 23.14](#) por indução sobre n . □

Como no caso do modelo de termos, antes de provar o lema da verdade precisamos do seguinte lema.

fol:com:ide: lem:val-in-termmodel-factored **Lema 23.18.** *Seja $\mathfrak{M} = \mathfrak{M}(\Gamma^*)$; então $\text{Val}^{\mathfrak{M}/\approx}(t) = [t]_{\approx}$.*

Prova. A prova é similar à de [Lema 23.10](#). □

Problema 23.3. Complete a prova de [Lema 23.18](#).

fol:com:ide: lem:truth **Lema 23.19.** *$\mathfrak{M}/\approx \models \varphi$ se e somente se $\varphi \in \Gamma^*$ para todas as sentenças φ .*

Prova. Por indução sobre φ , exatamente como na prova de [Lema 23.12](#). O único caso que precisa de atenção adicional é quando $\varphi \equiv t = t'$.

$\mathfrak{M}/\approx \models t = t'$ se e somente se $[t]_{\approx} = [t']_{\approx}$ (por definição de $\mathfrak{M}/\approx$)
se e somente se $t \approx t'$ (por definição de $[t]_{\approx}$)
se e somente se $t = t' \in \Gamma^*$ (por definição de $\approx$). □

Note que, embora $\mathfrak{M}(\Gamma^*)$ seja sempre [enumerável](#) e infinita, $\mathfrak{M}/\approx$ pode ser finita, já que pode ocorrer que haja apenas finitas classes $[t]_{\approx}$. Isso é de se esperar, já que Γ pode conter [sentenças](#) que exigem que qualquer [estrutura](#) na qual elas sejam verdadeiras seja finita. Por exemplo, $\forall x \forall y x = y$ é [a sentença](#) consistente, mas é satisfeita apenas em [estruturas](#) com [a domínio](#) que contém exatamente um [membro](#). digression

[content/first-order-logic/completeness/completeness-thm.tex](#)

23.8 O Teorema da Completude

fol:com:cth: sec Vamos combinar nossos resultados: chegamos ao teorema da completude. explanation

fol:com:cth: thm:completeness **Teorema 23.20 (Teorema da Completude).** *Seja Γ um conjunto de [sentenças](#). Se Γ é consistente, então é satisfatível.*

Prova. Suponha que Γ seja consistente. Por [Lema 23.6](#), existe um conjunto saturado consistente $\Gamma' \supseteq \Gamma$. Por [Lema 23.8](#), existe um $\Gamma^* \supseteq \Gamma'$ que é consistente e [completo](#). Como $\Gamma' \subseteq \Gamma^*$, para cada [fórmula](#) $\varphi(x)$, Γ^* contém [a sentença](#) da forma $\exists x \varphi(x) \rightarrow \varphi(c)$ e, assim, Γ^* é saturado. Se Γ não contém $=$, então, por [Lema 23.12](#), $\mathfrak{M}(\Gamma^*) \models \varphi$ se e somente se $\varphi \in \Gamma^*$. Disso segue, em particular, que, para todo $\varphi \in \Gamma$, $\mathfrak{M}(\Gamma^*) \models \varphi$, de modo que Γ é satisfatível. Se Γ de fato contém $=$, então, por [Lema 23.19](#), para todas as [sentenças](#) φ , $\mathfrak{M}/\approx \models \varphi$ se e somente se $\varphi \in \Gamma^*$. Em particular, $\mathfrak{M}/\approx \models \varphi$ para todo $\varphi \in \Gamma$, de modo que Γ é satisfatível. □

fol:com:cth: cor:completeness **Corolário 23.21 (Teorema da Completude, Segunda Versão).** *Para todo Γ e [sentenças](#) φ : se $\Gamma \models \varphi$ então $\Gamma \vdash \varphi$.*

Prova. Note que os Γ em [Corolário 23.21](#) e [Teorema 23.20](#) são universalmente quantificados. Para nos certificarmos de que não nos confundimos, reformulemos [Teorema 23.20](#) usando uma variável diferente: para qualquer conjunto de [sentenças](#) Δ , se Δ é consistente, então é satisfatível. Por contraposição, se Δ não é satisfatível, então Δ é inconsistente. Usaremos isso para provar o corolário.

Suponha que $\Gamma \models \varphi$. Então $\Gamma \cup \{\neg\varphi\}$ é insatisfatível por [Proposição 16.28](#). Tomando $\Gamma \cup \{\neg\varphi\}$ como nosso Δ , a versão anterior de [Teorema 23.20](#) nos dá que $\Gamma \cup \{\neg\varphi\}$ é inconsistente. Por [Proposições 19.19, 20.19, 21.19 e 22.27](#), $\Gamma \vdash \varphi$. $\square$

Problema 23.4. Use [Corolário 23.21](#) para provar [Teorema 23.20](#), mostrando assim que as duas formulações do teorema da completude são equivalentes.

Problema 23.5. Para que um sistema de [derivação](#) seja completo, suas regras devem ser fortes o suficiente para provar que todo conjunto insatisfatível é inconsistente. Quais das regras de [derivação](#) foram necessárias para provar a completude? Alguma dessas regras não é usada em nenhum lugar da prova? Para responder a essas perguntas, faça uma lista ou diagrama que mostre quais das regras de [derivação](#) foram usadas em quais resultados que levam à prova de [Teorema 23.20](#). Certifique-se de anotar quaisquer usos tácitos de regras nessas provas.

[content/first-order-logic/completeness/compactness.tex](#)

23.9 O Teorema da Compacidade

Uma consequência importante do teorema da completude é o teorema da compacidade. O teorema da compacidade afirma que, se cada subconjunto *finito* de um conjunto de [sentenças](#) é satisfatível, o conjunto inteiro é satisfatível — mesmo que o próprio conjunto seja infinito. Isso está longe de ser óbvio. Não há nada que pareça descartar, ao menos à primeira vista, a possibilidade de existirem conjuntos infinitos de [sentenças](#) que sejam contraditórios, mas em que a contradição só surge, por assim dizer, do número infinito. O teorema da compacidade diz que tal cenário pode ser descartado: não existem conjuntos infinitos insatisfatíveis de [sentenças](#) tais que cada subconjunto finito deles seja satisfatível. Como o teorema da completude, ele tem uma versão relacionada à consequência lógica: se um conjunto infinito de [sentenças](#) implica algo, então já um subconjunto finito o implica.

[fol:com:com:](#)
[sec](#)

Definição 23.22. Um conjunto Γ de [fórmulas](#) é *finitamente satisfatível* se e somente se todo $\Gamma_0 \subseteq \Gamma$ finito é satisfatível.

Teorema 23.23 (Teorema da Compacidade). *O seguinte vale para quaisquer sentenças Γ e φ :*

[fol:com:com:](#)
[thm:compactness](#)

23.9. O TEOREMA DA COMPACIDADE

1. $\Gamma \models \varphi$ se e somente se existe um $\Gamma_0 \subseteq \Gamma$ finito tal que $\Gamma_0 \models \varphi$.

2. Γ é satisfatível se e somente se é finitamente satisfatível.

Prova. Provamos (2). Se Γ é satisfatível, então existe a estrutura $\mathfrak{M}$ tal que $\mathfrak{M} \models \varphi$ para todo $\varphi \in \Gamma$. É claro que essa $\mathfrak{M}$ também satisfaz todo subconjunto finito de Γ , de modo que Γ é finitamente satisfatível.

Agora suponha que Γ seja finitamente satisfatível. Então todo subconjunto finito $\Gamma_0 \subseteq \Gamma$ é satisfatível. Pela correção (Corolários 19.31, 20.29, 21.31 e 22.38), todo subconjunto finito é consistente. Então o próprio Γ deve ser consistente por Proposições 19.17, 20.17, 21.17 e 22.21. Pela completude (Teorema 23.20), como Γ é consistente, ele é satisfatível. $\square$

Problema 23.6. Prove (1) de Teorema 23.23.

Exemplo 23.24. Em todo modelo $\mathfrak{M}$ de uma teoria Γ , cada termo t obviamente designa um membro de $|\mathfrak{M}|$. Podemos garantir que também seja verdade que todo membro de $|\mathfrak{M}|$ seja designado por algum termo? Em outras palavras, existem teorias Γ cujos modelos são todos cobertos? O teorema da compacidade mostra que esse não é o caso se Γ tem modelos infinitos. Eis como ver isso: Seja $\mathfrak{M}$ um modelo infinito de Γ , e seja c a símbolo de constante que não está na linguagem de Γ . Seja Δ o conjunto de todas as sentenças $c \neq t$ para t um termo na linguagem $\mathcal{L}$ de Γ , isto é,

$$\Delta = \{c \neq t : t \in \text{Trm}(\mathcal{L})\}.$$

Um subconjunto finito de $\Gamma \cup \Delta$ pode ser escrito como $\Gamma' \cup \Delta'$, com $\Gamma' \subseteq \Gamma$ e $\Delta' \subseteq \Delta$. Como Δ' é finito, ele pode conter apenas finitos termos. Seja $a \in |\mathfrak{M}|$ um membro de $|\mathfrak{M}|$ que não é designado por nenhum deles, e seja $\mathfrak{M}'$ a estrutura que é exatamente como $\mathfrak{M}$, mas com $c^{\mathfrak{M}'} = a$ também. Como $a \neq \text{Val}^{\mathfrak{M}}(t)$ para todo t que ocorre em Δ' , $\mathfrak{M}' \models \Delta'$. Como $\mathfrak{M} \models \Gamma$, $\Gamma' \subseteq \Gamma$, e c não ocorre em Γ , também $\mathfrak{M}' \models \Gamma'$. Juntando, $\mathfrak{M}' \models \Gamma' \cup \Delta'$ para todo subconjunto finito $\Gamma' \cup \Delta'$ de $\Gamma \cup \Delta$. Assim, todo subconjunto finito de $\Gamma \cup \Delta$ é satisfatível. Por compacidade, o próprio $\Gamma \cup \Delta$ é satisfatível. Logo, existem modelos $\mathfrak{M} \models \Gamma \cup \Delta$. Todo tal $\mathfrak{M}$ é um modelo de Γ , mas não é coberto, já que $\text{Val}^{\mathfrak{M}}(c) \neq \text{Val}^{\mathfrak{M}}(t)$ para todos os termos t de $\mathcal{L}$.

Exemplo 23.25. Considere a linguagem $\mathcal{L}$ contendo o símbolo de predicado $<$, símbolos de constante 0 , 1 , e símbolos de função $+$, $\times$, e $-$. Seja Γ o conjunto de todas as sentenças nesta linguagem verdadeiras na estrutura $\mathfrak{Q}$ com domínio $\mathbb{Q}$ e as interpretações óbvias. Γ é o conjunto de todas as sentenças de $\mathcal{L}$ verdadeiras a respeito dos números racionais. É claro que, em $\mathbb{Q}$ (e mesmo em $\mathbb{R}$), não há números r que sejam maiores que 0 mas menores que $1/k$ para todo $k \in \mathbb{Z}^+$. Tal número, se existisse, seria um *infinitesimal*: não nulo, mas infinitamente pequeno. O teorema da compacidade pode ser usado para mostrar que há modelos de Γ nos quais infinitesimais existem. Não temos a símbolo de função para a divisão em nossa linguagem (a divisão por zero é indefinida, e símbolos

de função têm de ser interpretadas por funções totais). Contudo, ainda podemos expressar que $r < 1/k$, já que isso ocorre se e somente se $r \cdot k < 1$. Agora seja c a **símbolo de constante** nova e seja Δ

$$\{0 < c\} \cup \{c \times \bar{k} < 1 : k \in \mathbb{Z}^+\}$$

(onde $\bar{k} = (1 + (1 + \dots + (1 + 1) \dots))$ com k 1's). Para qualquer subconjunto finito Δ_0 de Δ existe um K tal que todas as **sentenças** $c \times \bar{k} < 1$ em Δ_0 têm $k < K$. Se expandirmos $\mathfrak{Q}$ para $\mathfrak{Q}'$ com $c^{\mathfrak{Q}'} = 1/K$ temos que $\mathfrak{Q}' \models \Gamma_0 \cup \Delta_0$ para qualquer $\Gamma_0 \subseteq \Gamma$ finito, e assim $\Gamma \cup \Delta$ é finitamente satisfatível (Exercício: prove isso em detalhe). Por compacidade, $\Gamma \cup \Delta$ é satisfatível. Qualquer modelo $\mathfrak{S}$ de $\Gamma \cup \Delta$ contém um infinitesimal, a saber $c^{\mathfrak{S}}$.

Problema 23.7. No modelo padrão da aritmética $\mathfrak{N}$, não há **membro** $k \in |\mathfrak{N}|$ que satisfaça toda fórmula $\bar{n} < x$ (onde $\bar{n}$ é $0' \dots'$ com n 1's). Use o teorema da compacidade para mostrar que o conjunto de **sentenças** na linguagem da aritmética que são verdadeiras no modelo padrão da aritmética $\mathfrak{N}$ também são verdadeiras em a **estrutura** $\mathfrak{N}'$ que contém **um membro** que *de fato* satisfaz toda fórmula $\bar{n} < x$.

Exemplo 23.26. Sabemos que a lógica de primeira ordem com **predicado de identidade** pode expressar que o tamanho do **domínio** deve ter algum tamanho mínimo: A sentença $\varphi_{\geq n}$ (que diz “existem pelo menos n objetos distintos”) é verdadeira apenas em estruturas onde $|\mathfrak{M}|$ tem pelo menos n objetos. Assim, se tomarmos

$$\Delta = \{\varphi_{\geq n} : n \geq 1\}$$

então qualquer modelo de Δ deve ser infinito. Desse modo, podemos garantir que uma teoria só tenha modelos infinitos adicionando Δ a ela: os modelos de $\Gamma \cup \Delta$ são todos os modelos infinitos de Γ , e apenas eles.

Portanto, a lógica de primeira ordem pode expressar a infinitude. O teorema da compacidade mostra que ela não pode expressar a finitude, no entanto. Pois suponha que algum conjunto de sentenças Λ fosse satisfeito em todas as **estruturas** finitas, e apenas nelas. Então $\Delta \cup \Lambda$ é finitamente satisfatível. Por quê? Suponha que $\Delta' \cup \Lambda' \subseteq \Delta \cup \Lambda$ seja finito, com $\Delta' \subseteq \Delta$ e $\Lambda' \subseteq \Lambda$. Seja n o maior número tal que $\varphi_{\geq n} \in \Delta'$. Λ , sendo satisfeito em todas as estruturas finitas, tem um modelo $\mathfrak{M}$ com um número finito de **membros**, mas $\geq n$ deles. Mas então $\mathfrak{M} \models \Delta' \cup \Lambda'$. Por compacidade, $\Delta \cup \Lambda$ tem um modelo infinito, contradizendo a suposição de que Λ é satisfeito apenas em **estruturas** finitas.

[content/first-order-logic/completeness/compactness-direct.tex](#)

23.10 Uma Prova Direta do Teorema da Compacidade

fol:com:cpd:
sec

23.10. UMA PROVA DIRETA DO TEOREMA DA COMPACIDADE

Podemos provar o Teorema da Compacidade diretamente, sem apelar ao Teorema da Completude, usando as mesmas ideias da prova do teorema da completude. Na prova do Teorema da Completude, começamos com um conjunto Γ consistente de **sentenças**, expandimos-o para um conjunto Γ^* de **sentenças** consistente, saturado, e **completo**, e então mostramos que no modelo de termos $\mathfrak{M}(\Gamma^*)$ construído a partir de Γ^* , todas as **sentenças** de Γ são verdadeiras, de modo que Γ é satisfatível.

Podemos usar o mesmo método para mostrar que um conjunto finitamente satisfatível de sentenças é satisfatível. Só temos que provar as versões correspondentes dos resultados que levam ao lema da verdade, substituindo “consistente” por “finitamente satisfatível”.

fol:com:cpd: prop:fsat-ccs **Proposição 23.27.** *Suponha que Γ é **completo** e finitamente satisfatível. Então:*

1. $(\varphi \wedge \psi) \in \Gamma$ se, e somente se, tanto $\varphi \in \Gamma$ quanto $\psi \in \Gamma$.
2. $(\varphi \vee \psi) \in \Gamma$ se, e somente se, $\varphi \in \Gamma$ ou $\psi \in \Gamma$.
3. $(\varphi \rightarrow \psi) \in \Gamma$ se, e somente se, $\varphi \notin \Gamma$ ou $\psi \in \Gamma$.

Problema 23.8. Prove **Proposição 23.27**. Evite o uso de $\vdash$.

fol:com:cpd: lem:fsat-henkin **Lema 23.28.** *Todo conjunto finitamente satisfatível Γ pode ser estendido a um conjunto saturado finitamente satisfatível Γ' .*

Problema 23.9. Prove **Lema 23.28**. (Dica: o passo crucial é mostrar que se Γ_n é finitamente satisfatível, então $\Gamma_n \cup \{\theta_n\}$ também o é, sem qualquer apelo a **derivações** ou consistência.)

fol:com:cpd: prop:fsat-instances **Proposição 23.29.** *Suponha que Γ é completo, finitamente satisfatível e saturado.*

1. $\exists x \varphi(x) \in \Gamma$ se, e somente se, $\varphi(t) \in \Gamma$ para pelo menos um termo fechado t .
2. $\forall x \varphi(x) \in \Gamma$ se, e somente se, $\varphi(t) \in \Gamma$ para todos os termos fechados t .

Problema 23.10. Prove **Proposição 23.29**.

fol:com:cpd: lem:fsat-lindenbaum **Lema 23.30.** *Todo conjunto finitamente satisfatível Γ pode ser estendido a um conjunto **completo** e finitamente satisfatível Γ^* .*

Problema 23.11. Prove **Lema 23.30**. (Dica: o passo crucial é mostrar que se Γ_n é finitamente satisfatível, então ou $\Gamma_n \cup \{\varphi_n\}$ ou $\Gamma_n \cup \{\neg\varphi_n\}$ é finitamente satisfatível.)

fol:com:cpd: thm:compactness-direct **Teorema 23.31 (Compacidade).** *Γ é satisfatível se, e somente se, é finitamente satisfatível.*

Prova. Se Γ é satisfatível, então há a estrutura $\mathfrak{M}$ tal que $\mathfrak{M} \models \varphi$ para todo $\varphi \in \Gamma$. É claro que esta $\mathfrak{M}$ também satisfaz todo subconjunto finito de Γ , de modo que Γ é finitamente satisfatível.

Agora suponha que Γ é finitamente satisfatível. Por [Lema 23.28](#), há um conjunto finitamente satisfatível e saturado $\Gamma' \supseteq \Gamma$. Por [Lema 23.30](#), Γ' pode ser estendido a um conjunto completo e finitamente satisfatível Γ^* , e Γ^* ainda é saturado. Construa o modelo de termos $\mathfrak{M}(\Gamma^*)$ como em [Definição 23.9](#). Note que [Proposição 23.11](#) não dependeu do fato de que Γ^* é consistente (ou completo ou saturado, aliás), mas apenas do fato de que $\mathfrak{M}(\Gamma^*)$ é coberta. A prova do Lema da Verdade ([Lema 23.12](#)) funciona se substituírmos referências a [Proposição 23.2](#) e [Proposição 23.7](#) por referências a [Proposição 23.27](#) e [Proposição 23.29](#). $\square$

Problema 23.12. Escreva a prova completa do Lema da Verdade ([Lema 23.12](#)) na versão necessária para a prova de [Teorema 23.31](#).

[content/first-order-logic/completeness/downward-ls.tex](#)

23.11 O Teorema de Löwenheim–Skolem

O Teorema de Löwenheim–Skolem diz que se uma teoria tem um modelo infinito, então ela também tem um modelo que é no máximo infinito enumerável. Uma consequência imediata desse fato é que a lógica de primeira ordem não pode expressar que o tamanho de a estrutura é não enumerável: qualquer sentença ou conjunto de sentenças satisfeito em todas as estruturas não enumeráveis também é satisfeito em alguma estrutura enumerável.

[fol:com:dls:](#)
[sec](#)

Teorema 23.32. *Se Γ é consistente, então ela tem um modelo enumerável, isto é, é satisfatível em a estrutura cujo domínio é finito ou infinito enumerável.*

[fol:com:dls:](#)
[thm:downward-ls](#)

Prova. Se Γ é consistente, a estrutura $\mathfrak{M}$ fornecida pela prova do teorema da completude tem um domínio $|\mathfrak{M}|$ que não é maior do que o conjunto dos termos da linguagem $\mathcal{L}$. Assim $\mathfrak{M}$ é no máximo infinito enumerável. $\square$

Teorema 23.33. *Se Γ é um conjunto consistente de sentenças na linguagem da lógica de primeira ordem sem identidade, então ela tem um modelo infinito enumerável, isto é, é satisfatível em a estrutura cujo domínio é infinito e enumerável.*

[fol:com:dls:](#)
[noidentity-ls](#)

Prova. Se Γ é consistente e não contém sentenças em que a identidade aparece, então a estrutura $\mathfrak{M}$ fornecida pela prova do teorema da completude tem um domínio $|\mathfrak{M}|$ idêntico ao conjunto dos termos da linguagem $\mathcal{L}'$. Assim $\mathfrak{M}$ é infinito enumerável, pois $\text{Trm}(\mathcal{L}')$ o é. $\square$

Exemplo 23.34 (Paradoxo de Skolem). A teoria dos conjuntos de Zermelo–Fraenkel **ZFC** é um arcabouço muito poderoso no qual praticamente todas as afirmações matemáticas podem ser expressas, incluindo fatos sobre os tamanhos de conjuntos. Assim, por exemplo, **ZFC** pode provar que o conjunto $\mathbb{R}$ dos números reais é **não enumerável**, pode provar o Teorema de Cantor de que o conjunto potência de qualquer conjunto é maior do que o próprio conjunto, etc. Se **ZFC** é consistente, seus modelos são todos infinitos e, além disso, todos contêm **membros** sobre os quais a teoria diz que são **não enumeráveis**, como o elemento que torna verdadeiro o teorema de **ZFC** de que o conjunto potência dos números naturais existe. Pelo Teorema de Löwenheim–Skolem, **ZFC** também tem modelos **enumeráveis**—modelos que contêm conjuntos “**não enumeráveis**”, mas que eles mesmos são **enumeráveis**.

Capítulo 24

Além da lógica de primeira ordem

Este capítulo, adaptado das notas de lógica de Jeremy Avigad, fornece o mais breve vislumbre de quais outros sistemas lógicos existem. Pretende ser um capítulo que sugere tópicos adicionais para estudo em um curso que não os cobre. Cada um dos tópicos mencionados aqui irá — espero — eventualmente receber seu próprio tratamento em nível de parte no Projeto Open Logic.

`content/first-order-logic/beyond/introduction.tex`

24.1 Visão Geral

fol:byd:int:
sec

A lógica de primeira ordem não é o único sistema de lógica de interesse: há muitas extensões e variações da lógica de primeira ordem. Uma lógica normalmente consiste na especificação formal de uma linguagem, geralmente, mas

nem sempre, em um sistema dedutivo e, geralmente, mas nem sempre, em uma semântica pretendida. Mas o uso técnico do termo levanta uma pergunta óbvia: o que as lógicas que não são lógicas de primeira ordem têm a ver com a palavra “lógica”, usada no sentido intuitivo ou filosófico? Todos os sistemas descritos abaixo são projetados para modelar o raciocínio de uma forma ou de outra; podemos dizer o que os torna lógicos?

Não há respostas fáceis. A palavra “lógica” é usada de diferentes maneiras e em diferentes contextos, e a noção, como a de “verdade”, tem sido analisada a partir de numerosos posicionamentos filosóficos. Por exemplo, pode-se considerar que o objetivo do raciocínio lógico é a determinação de quais afirmações são necessariamente verdadeiras, verdadeiras a priori, verdadeiras independentemente da interpretação dos termos não lógicos, verdadeiras em virtude de sua forma, ou verdadeiras por convenção linguística; e cada uma dessas concepções requer bastante esclarecimento. Mesmo que se restrinja a atenção ao tipo de lógica usada na matemática, há pouco acordo quanto ao seu alcance. Por exemplo, no *Principia Mathematica*, Russell e Whitehead tentaram desenvolver a matemática com base na lógica, na tradição *logicista* iniciada por Frege. Seu sistema de lógica era uma forma de lógica de tipo superior semelhante à descrita abaixo. No final, foram forçados a introduzir axiomas que, pela maioria dos padrões, não parecem puramente lógicos (notavelmente, o axioma do infinito e o axioma da redutibilidade), mas pode-se, ainda assim, sustentar que algumas formas de raciocínio de ordem superior devem ser aceitas como lógicas. Em contraste, Quine, cuja ontologia não admite “proposições” como objetos legítimos de discurso, argumenta que a lógica de segunda ordem e a de ordem superior são realmente manifestações da teoria dos conjuntos em pele de cordeiro; em outras palavras, sistemas que envolvem quantificação sobre predicados não são puramente lógicos.

Por enquanto, é melhor deixar tais questões filosóficas para um dia chuvoso e simplesmente pensar nos sistemas abaixo como idealizações formais de vários tipos de raciocínio, lógico ou não.

`content/first-order-logic/beyond/many-sorted-logic.tex`

24.2 Lógica de Múltiplas Ordenações

Na lógica de primeira ordem, variáveis e quantificadores variam sobre um único domínio. Mas é frequentemente útil ter vários domínios (disjuntos): por exemplo, pode-se querer ter **a** domínio de números, **a** domínio de objetos geométricos, **a** domínio de funções de números para números, **a** domínio de grupos abelianos, e assim por diante.

A lógica de múltiplas ordenações fornece esse tipo de arcabouço. Começa-se com uma lista de “ordenações”—a “ordenação” de um objeto indica o “domínio” que ele deveria habitar. Tem-se então **variáveis** e quantificadores para cada ordenação e (geralmente) uma **predicado de identidade** para cada ordenação. Funções e relações também são “tipadas” pelas ordenações dos

fol:byd:msl:
sec

24.3. LÓGICA DE SEGUNDA ORDEM

objetos que podem receber como argumentos. Caso contrário, mantêm-se as regras usuais da lógica de primeira ordem, com versões das regras de quantificadores repetidas para cada ordenação.

Por exemplo, para estudar relações internacionais, poderíamos escolher uma linguagem com dois tipos de objetos, cidadãos franceses e cidadãos alemães. Poderíamos ter uma relação unária, “bebe vinho”, para objetos da primeira ordenação; outra relação unária, “come wurst”, para objetos da segunda ordenação; e uma relação binária, “forma um casal multinacional”, que recebe dois argumentos, onde o primeiro argumento é da primeira ordenação e o segundo argumento é da segunda ordenação. Se usarmos variáveis a, b, c para variar sobre cidadãos franceses e x, y, z para variar sobre cidadãos alemães, então

$$\forall a \forall x [(CasadoCom(a, x) \rightarrow (BebeVinho(a) \vee \neg ComeWurst(x)))]$$

afirma que, se algum francês for casado com um alemão, ou o francês bebe vinho ou o alemão não come wurst.

A lógica de múltiplas ordenações pode ser embutida na lógica de primeira ordem de maneira natural, reunindo todos os objetos dos domínios de múltiplas ordenações em um único domínio de primeira ordem, usando símbolos de predicado unários para acompanhar as ordenações e relativizando quantificadores. Por exemplo, a linguagem de primeira ordem correspondente ao exemplo acima teria símbolos de predicado unários “*Alemao*” e “*Frances*”, além das outras relações descritas, com os requisitos de ordenação eliminados. Um quantificador ordenado $\forall x \varphi$, onde x é a variável da ordenação alemã, traduz-se para

$$\forall x (Alemao(x) \rightarrow \varphi).$$

Precisamos adicionar axiomas que garantam que as ordenações sejam separadas—por exemplo, $\forall x \neg (Alemao(x) \wedge Frances(x))$ —, bem como axiomas que garantam que “bebe vinho” só vale de objetos que satisfazem o símbolo de predicado *Frances*(x), etc. Com essas convenções e axiomas, não é difícil mostrar que sentenças de múltiplas ordenações traduzem-se em sentenças de primeira ordem, e derivações de múltiplas ordenações traduzem-se em derivações de primeira ordem. Além disso, estruturas de múltiplas ordenações “traduzem-se” em estruturas de primeira ordem correspondentes e vice-versa, de modo que também temos um teorema de completude para a lógica de múltiplas ordenações.

content/first-order-logic/beyond/second-order-logic.tex

24.3 Lógica de Segunda Ordem

fol:byd:sol: A linguagem da lógica de segunda ordem permite quantificar não apenas sobre
sec a domínio de indivíduos, mas também sobre relações nesse domínio. Dada uma linguagem de primeira ordem $\mathcal{L}$, para cada k adiciona-se variáveis R que

variam sobre relações k -árias, e permite-se quantificação sobre essas variáveis. Se R é a **variável** para uma relação k -ária, e $t_1, \dots, t_k$ são termos ordinários (de primeira ordem), $R(t_1, \dots, t_k)$ é uma **fórmula** atômica. Caso contrário, o conjunto de **fórmulas** é definido exatamente como no caso da lógica de primeira ordem, com cláusulas adicionais para quantificação de segunda ordem. Note que temos apenas a **predicado de identidade** para termos de primeira ordem: se R e S são **variáveis** de relação de mesma aridade k , podemos definir $R = S$ como uma abreviação de

$$\forall x_1 \dots \forall x_k (R(x_1, \dots, x_k) \leftrightarrow S(x_1, \dots, x_k)).$$

As regras para a lógica de segunda ordem simplesmente estendem as regras de quantificadores às novas variáveis de segunda ordem. Aqui, porém, é preciso ter um pouco de cuidado para explicar como essas variáveis interagem com os **símbolos de predicado** de $\mathcal{L}$, e com **fórmulas** de $\mathcal{L}$ de modo mais geral. No mínimo, variáveis de relação contam como termos, de modo que temos inferências da forma

$$\varphi(R) \vdash \exists R \varphi(R)$$

Mas se $\mathcal{L}$ é a linguagem da aritmética com um símbolo de relação constante $<$, também esperaríamos que a seguinte inferência fosse válida:

$$x < y \vdash \exists R R(x, y)$$

ou, para uma dada **fórmula** φ ,

$$\varphi(x_1, \dots, x_k) \vdash \exists R R(x_1, \dots, x_k)$$

Mais geralmente, podemos querer permitir inferências da forma

$$\varphi[\lambda \vec{x}. \psi(\vec{x})/R] \vdash \exists R \varphi$$

onde $\varphi[\lambda \vec{x}. \psi(\vec{x})/R]$ denota o resultado de substituir cada **fórmula** atômica da forma $Rt_1, \dots, t_k$ em φ por $\psi(t_1, \dots, t_k)$. Esta última regra é equivalente a ter um *esquema de compreensão*, isto é, um axioma da forma

$$\exists R \forall x_1, \dots, x_k (\varphi(x_1, \dots, x_k) \leftrightarrow R(x_1, \dots, x_k)),$$

um para cada **fórmula** φ na linguagem de segunda ordem, em que R não é uma variável livre. (Exercício: mostre que se R for permitida em φ , esse esquema é inconsistente!)

Quando lógicos se referem aos “axiomas da lógica de segunda ordem”, geralmente querem dizer a extensão mínima da lógica de primeira ordem pelas regras de quantificadores de segunda ordem juntamente com o esquema de compreensão. Mas é frequentemente interessante estudar subsistemas mais fracos desses axiomas e regras. Por exemplo, note que em toda a sua generalidade o esquema de axiomas de compreensão é *impredicativo*: ele permite afirmar a existência de uma relação $R(x_1, \dots, x_k)$ que é “definida” por a **fórmula** com

24.3. LÓGICA DE SEGUNDA ORDEM

quantificadores de segunda ordem; e esses quantificadores variam sobre o conjunto de todas essas relações—um conjunto que inclui R em si! Por volta da virada do século XX, uma reação comum ao paradoxo de Russell foi atribuir a culpa a tais definições e evitá-las ao desenvolver os fundamentos da matemática. Se se proíbe o uso de quantificadores de segunda ordem na fórmula φ , tem-se uma forma *predicativa* de compreensão, que é um pouco mais fraca.

Do ponto de vista semântico, pode-se pensar em uma *estrutura* de segunda ordem como consistindo em uma *estrutura* de primeira ordem para a linguagem, juntamente com um conjunto de relações no *domínio* sobre o qual variem os quantificadores de segunda ordem (mais precisamente, para cada k há um conjunto de relações de aridade k). É claro que, se a compreensão estiver incluída no sistema de *derivação*, então temos o requisito adicional de que há relações suficientes na “parte de segunda ordem” para satisfazer os axiomas de compreensão—caso contrário, o sistema de *derivação* não é correto! Uma maneira fácil de garantir que há relações suficientes é tomar a parte de segunda ordem como consistindo de *todas* as relações sobre a parte de primeira ordem. Uma tal *estrutura* é chamada *plena*, e, de certo modo, é realmente a “*estrutura* pretendida” para a linguagem. Se restringirmos a atenção a *estruturas* plenas, temos o que se conhece como a semântica de segunda ordem *plena*. Nesse caso, especificar uma *estrutura* reduz-se a especificar a parte de primeira ordem, pois o conteúdo da parte de segunda ordem segue disso implicitamente.

Para resumir, há certa ambiguidade ao falar de lógica de segunda ordem. Em termos do sistema de *derivação*, pode-se ter em mente

1. Um sistema de *derivação* de segunda ordem “mínimo”, juntamente com alguns axiomas de compreensão.
2. O sistema de *derivação* de segunda ordem “padrão”, com compreensão plena.

Em termos da semântica, pode interessar

1. A semântica “fraca”, em que a *estrutura* consiste em uma parte de primeira ordem, juntamente com uma parte de segunda ordem grande o suficiente para satisfazer os axiomas de compreensão.
2. A semântica de segunda ordem “padrão”, em que se consideram apenas *estruturas* plenas.

Quando lógicos não especificam o sistema de *derivação* ou a semântica que têm em mente, geralmente se referem ao segundo item de cada lista. A vantagem de usar essa semântica é que, como veremos, ela nos dá descrições categóricas de muitas *estruturas* matemáticas naturais; ao mesmo tempo, o sistema de *derivação* é bastante forte, e correto para essa semântica. A desvantagem é que o sistema de *derivação* não é completo para a semântica; de fato, *nenhum* sistema de *derivação* efetivamente dado é completo para a semântica plena de segunda ordem. Por outro lado, veremos que o sistema de *derivação* é completo para a semântica enfraquecida; isso implica que, se uma sentença não

for demonstrável, então há *alguma estrutura*, não necessariamente a plena, na qual ela é falsa.

A linguagem da lógica de segunda ordem é bastante rica. Pode-se identificar relações unárias com subconjuntos do *domínio*, e assim, em particular, pode-se quantificar sobre esses conjuntos; por exemplo, pode-se expressar indução para os números naturais com um único axioma

$$\forall R ((R(0) \wedge \forall x (R(x) \rightarrow R(x')))) \rightarrow \forall x R(x)).$$

Se se toma a linguagem da aritmética com os símbolos $0, \iota, +, \times$ e $<$, pode-se adicionar os seguintes axiomas para descrever seu comportamento:

1. $\forall x \neg x' = 0$
2. $\forall x \forall y (s(x) = s(y) \rightarrow x = y)$
3. $\forall x (x + 0) = x$
4. $\forall x \forall y (x + y') = (x + y)'$
5. $\forall x (x \times 0) = 0$
6. $\forall x \forall y (x \times y') = ((x \times y) + x)$
7. $\forall x \forall y (x < y \leftrightarrow \exists z y = (x + z'))$

Não é difícil mostrar que esses axiomas, juntamente com o axioma de indução acima, fornecem uma descrição categórica da *estrutura* $\mathfrak{N}$, o modelo padrão da aritmética, desde que estejamos usando a semântica plena de segunda ordem. Dada qualquer *estrutura* $\mathfrak{M}$ na qual esses axiomas são verdadeiros, defina uma função f de $\mathbb{N}$ para o *domínio* de $\mathfrak{M}$ usando recursão ordinária em $\mathbb{N}$, de modo que $f(0) = 0^{\mathfrak{M}}$ e $f(x + 1) = \iota^{\mathfrak{M}}(f(x))$. Usando indução ordinária em $\mathbb{N}$ e o fato de que os axiomas (1) e (2) valem em $\mathfrak{M}$, vemos que f é *injetiva*. Para ver que f é *sobrejetiva*, seja P o conjunto de elementos de $|\mathfrak{M}|$ que estão no alcance de f . Como $\mathfrak{M}$ é plena, P está no *domínio* de segunda ordem. Pela construção de f , sabemos que $0^{\mathfrak{M}}$ está em P , e que P é fechado sob $\iota^{\mathfrak{M}}$. O fato de que o axioma de indução vale em $\mathfrak{M}$ (em particular, para P) garante que P é igual a todo o *domínio* de primeira ordem de $\mathfrak{M}$. Isso mostra que f é *uma bijeção*. Mostrar que f é um homomorfismo não é mais difícil, usando indução ordinária em $\mathbb{N}$ repetidamente.

Em termos teórico-conjuntistas, uma função é apenas um tipo especial de relação; por exemplo, uma função unária f pode ser identificada com uma relação binária R que satisfaz $\forall x \exists! y R(x, y)$. Como resultado, pode-se também quantificar sobre funções. Usando a semântica plena, pode-se então definir a classe de *estruturas* infinitas como a classe de *estruturas* $\mathfrak{M}$ para as quais há uma função injetiva do *domínio* de $\mathfrak{M}$ para um subconjunto próprio de si mesma:

$$\exists f (\forall x \forall y (f(x) = f(y) \rightarrow x = y) \wedge \exists y \forall x f(x) \neq y).$$

A negação dessa sentença define então a classe de *estruturas* finitas.

24.3. LÓGICA DE SEGUNDA ORDEM

Além disso, pode-se definir a classe de ordens bem-fundadas, adicionando o seguinte à definição de uma ordenação linear:

$$\forall P (\exists x P(x) \rightarrow \exists x (P(x) \wedge \forall y (y < x \rightarrow \neg P(y)))).$$

Isso afirma que todo conjunto não vazio tem um elemento mínimo, módulo a identificação de “conjunto” com “relação unária”. Como outro exemplo, pode-se expressar a noção de conexidade para grafos, dizendo que não há separação não trivial dos vértices em partes desconectadas:

$$\neg \exists A (\exists x A(x) \wedge \exists y \neg A(y) \wedge \forall w \forall z ((A(w) \wedge \neg A(z)) \rightarrow \neg R(w, z))).$$

Como mais um exemplo, pode-se tentar, como exercício, definir a classe de **estruturas** finitas cujo **domínio** tem tamanho par. De modo ainda mais impressionante, pode-se fornecer uma descrição categórica dos números reais como um corpo ordenado completo que contém os racionais.

Em suma, a lógica de segunda ordem é muito mais expressiva do que a lógica de primeira ordem. Essa é a boa notícia; agora a má. Já mencionamos que não há sistema de **derivação** efetivo que seja completo para a semântica plena de segunda ordem. Para o bem ou para o mal, muitas das propriedades da lógica de primeira ordem estão ausentes, incluindo compacidade e os teoremas de Löwenheim–Skolem.

Por outro lado, se se está disposto a abrir mão da semântica plena de segunda ordem em favor da mais fraca, então o sistema de **derivação** mínimo de segunda ordem é completo para essa semântica. Em outras palavras, se lemos $\vdash$ como “demonstra no sistema mínimo” e $\models$ como “implica logicamente na semântica fraca”, podemos mostrar que sempre que $\Gamma \models \varphi$, então $\Gamma \vdash \varphi$. Se se quer incluir axiomas específicos de compreensão no sistema de **derivação**, é preciso restringir a semântica a **estruturas** de segunda ordem que satisfaçam esses axiomas: por exemplo, se Δ consiste em um conjunto de axiomas de compreensão (possivelmente todos eles), temos que, se $\Gamma \cup \Delta \models \varphi$, então $\Gamma \cup \Delta \vdash \varphi$. Em particular, se φ não for demonstrável usando os axiomas de compreensão que estamos considerando, então há um modelo de $\neg \varphi$ no qual esses axiomas de compreensão ainda assim valem.

A maneira mais fácil de ver que o teorema de completude vale para a semântica fraca é pensar na lógica de segunda ordem como uma lógica de múltiplas ordenações, da seguinte forma. Uma ordenação é interpretada como o **domínio** ordinário de “primeira ordem”, e então, para cada k , temos a **domínio** de “relações de aridade k .” Tomamos a linguagem com símbolos de relação incorporados “ $verd_k(R, x_1, \dots, x_k)$ ”, destinados a afirmar que R vale de $x_1, \dots, x_k$, onde R é uma variável da ordenação “relação k -ária” e $x_1, \dots, x_k$ são objetos da ordenação de primeira ordem.

Com essa identificação, a semântica fraca de segunda ordem é essencialmente a semântica usual da lógica de múltiplas ordenações; e já observamos que a lógica de múltiplas ordenações pode ser embutida na lógica de primeira ordem. Módulo as traduções de ida e volta, então, a concepção mais fraca de lógica de segunda ordem é realmente uma forma disfarçada de lógica de

primeira ordem, em que o domínio contém tanto “objetos” quanto “relações” governados pelos axiomas apropriados.

`content/first-order-logic/beyond/higher-order-logic.tex`

24.4 Lógica de Ordem Superior

Passar da lógica de primeira ordem para a de segunda ordem nos permitiu falar sobre conjuntos de objetos no domínio de primeira ordem, dentro da linguagem formal. Por que parar aí? Por exemplo, a lógica de terceira ordem deveria nos permitir lidar com conjuntos de conjuntos de objetos, ou talvez até conjuntos que contenham tanto objetos quanto conjuntos de objetos. E a lógica de quarta ordem nos permitirá falar sobre conjuntos de objetos desse tipo. Como se pode imaginar, pode-se iterar essa ideia arbitrariamente.

Na prática, a lógica de ordem superior é frequentemente formulada em termos de funções em vez de relações. (Módulo as identificações naturais, essa diferença é inessencial.) Dadas algumas “ordenações” básicas $A, B, C, \dots$ (que agora chamaremos de “tipos”), podemos criar novos estipulando

fol:byd:hol:
sec

Se σ e τ são tipos finitos, então $\sigma \rightarrow \tau$ também o é.

Pense nos tipos como “rótulos” sintáticos que classificam os objetos que queremos em nosso domínio; $\sigma \rightarrow \tau$ descreve aqueles objetos que são funções que levam objetos de tipo σ a objetos de tipo τ . Por exemplo, podemos querer ter um tipo Ω de valores de verdade, “verdadeiro” e “falso”, e um tipo $\mathbb{N}$ de números naturais. Nesse caso, pode-se pensar em objetos de tipo $\mathbb{N} \rightarrow \Omega$ como relações unárias, ou subconjuntos de $\mathbb{N}$; objetos de tipo $\mathbb{N} \rightarrow \mathbb{N}$ são funções de números naturais para números naturais; e objetos de tipo $(\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ são “funcionais”, isto é, funções de tipo superior que levam funções a números.

Como no caso da lógica de segunda ordem, pode-se pensar na lógica de ordem superior como uma espécie de lógica de múltiplas ordenações, em que há uma ordenação para cada tipo de objeto que queremos considerar. Mas é usualmente mais claro simplesmente definir a sintaxe da lógica de tipos superiores a partir dos fundamentos. Por exemplo, podemos definir um conjunto de tipos finitos indutivamente, da seguinte forma:

1. $\mathbb{N}$ é um tipo finito.
2. Se σ e τ são tipos finitos, então $\sigma \rightarrow \tau$ também o é.
3. Se σ e τ são tipos finitos, então $\sigma \times \tau$ também o é.

Intuitivamente, $\mathbb{N}$ denota o tipo dos números naturais, $\sigma \rightarrow \tau$ denota o tipo de funções de σ para τ , e $\sigma \times \tau$ denota o tipo de pares de objetos, um de σ e outro de τ . Podemos então definir um conjunto de termos indutivamente, da seguinte forma:

1. Para cada tipo σ , há um estoque de variáveis $x, y, z, \dots$ de tipo σ

24.4. LÓGICA DE ORDEM SUPERIOR

2. o é um termo de tipo $\mathbb{N}$
3. S (sucessor) é um termo de tipo $\mathbb{N} \rightarrow \mathbb{N}$
4. Se s é um termo de tipo σ , e t é um termo de tipo $\mathbb{N} \rightarrow (\sigma \rightarrow \sigma)$, então R_{st} é um termo de tipo $\mathbb{N} \rightarrow \sigma$
5. Se s é um termo de tipo $\tau \rightarrow \sigma$ e t é um termo de tipo τ , então $s(t)$ é um termo de tipo σ
6. Se s é um termo de tipo σ e x é uma variável de tipo τ , então $\lambda x. s$ é um termo de tipo $\tau \rightarrow \sigma$.
7. Se s é um termo de tipo σ e t é um termo de tipo τ , então $\langle s, t \rangle$ é um termo de tipo $\sigma \times \tau$.
8. Se s é um termo de tipo $\sigma \times \tau$, então $p_1(s)$ é um termo de tipo σ e $p_2(s)$ é um termo de tipo τ .

Intuitivamente, R_{st} denota a função definida recursivamente por

$$\begin{aligned} R_{st}(0) &= s \\ R_{st}(x+1) &= t(x, R_{st}(x)), \end{aligned}$$

$\langle s, t \rangle$ denota o par cuja primeira componente é s e cuja segunda componente é t , e $p_1(s)$ e $p_2(s)$ denotam o primeiro e o segundo elementos (“projeções”) de s . Finalmente, $\lambda x. s$ denota a função f definida por

$$f(x) = s$$

para qualquer x de tipo σ ; assim o item (6) nos dá uma forma de compreensão, permitindo-nos definir funções usando termos. As **fórmulas** são construídas a partir de afirmações de **predicado de identidade** $s = t$ entre termos do mesmo tipo, dos conectivos proposicionais usuais e da quantificação de ordem superior. Pode-se então tomar os axiomas do sistema como as equações básicas que governam os termos definidos acima, juntamente com as regras usuais da lógica com quantificadores e **predicado de identidade**.

Se se aumenta o sistema de tipos finitos com um tipo Ω de valores de verdade, é preciso incluir axiomas que governem seu uso também. De fato, se se for astuto, pode-se eliminar **fórmulas** complexas inteiramente, substituindo-as por termos de tipo Ω ! O sistema de prova pode então ser modificado de acordo. O resultado é essencialmente a *teoria simples dos tipos* apresentada por Alonzo Church na década de 1930.

Como no caso da lógica de segunda ordem, há diferentes versões de semântica de tipos superiores que se pode querer usar. Na versão completa, variáveis de tipo $\sigma \rightarrow \tau$ variam sobre o conjunto de *todas* as funções dos objetos de tipo σ para objetos de tipo τ . Como se pode imaginar, essa semântica é forte demais para admitir um sistema de **derivação** completo e efetivo. Mas pode-se

considerar uma semântica mais fraca, em que a **estrutura** consiste em conjuntos de elementos T_τ para cada tipo τ , juntamente com operações apropriadas para aplicação, projeção, etc. Se os detalhes forem desenvolvidos corretamente, pode-se obter teoremas de completude para os tipos de sistemas de **derivação** descritos acima.

A lógica de ordem superior é atraente porque fornece um arcabouço no qual podemos embutir boa parte da matemática de maneira natural: a partir de $\mathbb{N}$, pode-se definir números reais, funções contínuas e assim por diante. É também particularmente atraente no contexto da lógica intuicionista, pois os tipos têm interpretações “construtivas” claras. De fato, pode-se desenvolver versões construtivas da semântica de tipos superiores (baseadas na lógica intuicionista, em vez da clássica) que esclarecem bastante essas interpretações construtivas e são, em muitos aspectos, mais interessantes do que as contrapartes clássicas.

`content/first-order-logic/beyond/intuitionistic-logic.tex`

24.5 Lógica Intuicionista

Em contraste com a lógica de segunda ordem e de ordem superior, a lógica intuicionista de primeira ordem representa uma restrição da versão clássica, destinada a modelar um tipo de raciocínio mais “construtivo”. Os exemplos a seguir podem ilustrar algumas das motivações subjacentes. fol:byd:il:
sec

Suponha que alguém se aproximasse de você um dia e anunciasse que havia determinado um número natural x , com a propriedade de que se x é primo, a hipótese de Riemann é verdadeira, e se x é composto, a hipótese de Riemann é falsa. Ótimas notícias! Se a hipótese de Riemann é verdadeira ou não é uma das grandes questões em aberto da matemática, e aqui parecem ter reduzido o problema a um cálculo, isto é, à determinação de se um número específico é primo ou não.

Qual é o valor mágico de x ? Eles o descrevem assim: x é o número natural que é igual a 7 se a hipótese de Riemann é verdadeira, e 9 caso contrário.

Com raiva, você exige seu dinheiro de volta. De um ponto de vista clássico, a descrição acima de fato determina um valor único de x ; mas o que você realmente quer é um valor de x que seja dado *explicitamente*.

Para tomar outro exemplo, talvez menos artificial, considere a seguinte questão. Sabemos que é possível elevar um número irracional a uma potência racional e obter um resultado racional. Por exemplo, $\sqrt{2}^2 = 2$. O que é menos claro é se é possível elevar um número irracional a uma potência *irracional* e obter um resultado racional. O seguinte teorema responde afirmativamente:

Teorema 24.1. *Existem números irracionais a e b tais que a^b é racional.*

Prova. Considere $\sqrt{2}^{\sqrt{2}}$. Se isto é racional, terminamos: podemos tomar $a = b = \sqrt{2}$. Caso contrário, é irracional. Então temos

$$(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{\sqrt{2} \cdot \sqrt{2}} = \sqrt{2}^2 = 2,$$

24.5. LÓGICA INTUICIONISTA

que é certamente racional. Assim, neste caso, tome a como $\sqrt{2}^{\sqrt{2}}$, e b como $\sqrt{2}$. $\square$

Isso constitui uma prova válida? A maioria dos matemáticos sente que sim. Mas, novamente, há algo um pouco insatisfatório aqui: provamos a existência de um par de números reais com uma certa propriedade, sem poder dizer *qual* par de números é. É possível provar o mesmo resultado, mas de tal forma que o par a, b é dado na prova: tome $a = \sqrt{3}$ e $b = \log_3 4$. Então

$$a^b = \sqrt{3}^{\log_3 4} = 3^{1/2 \cdot \log_3 4} = (3^{\log_3 4})^{1/2} = 4^{1/2} = 2,$$

pois $3^{\log_3 x} = x$.

A lógica intuicionista foi concebida para modelar um tipo de raciocínio no qual movimentos como o da primeira prova são proibidos. Provar a existência de um x que satisfaz $\varphi(x)$ significa que se deve fornecer um x específico e uma prova de que ele satisfaz φ , como na segunda prova. Provar que φ ou ψ vale requer que se possa provar um ou outro.

Formalmente falando, a lógica intuicionista de primeira ordem é o que se obtém se se restringe um sistema de **derivação** para a lógica de primeira ordem de certa maneira. Similarmente, há versões intuicionistas da lógica de segunda ou de ordem superior. Do ponto de vista matemático, estes são apenas sistemas dedutivos formais, mas, como já observado, são destinados a modelar um tipo de raciocínio matemático. Pode-se tomar isso como o tipo de raciocínio justificado por uma certa visão filosófica da matemática (como o intuicionismo de Brouwer); pode-se tomá-lo como um tipo de raciocínio matemático mais “concreto” e satisfatório (na linha do construtivismo de Bishop); e pode-se discutir se a descrição formal captura ou não a motivação informal. Mas quaisquer que sejam as posições filosóficas que possamos ter, podemos estudar a lógica intuicionista como uma lógica formalmente apresentada; e por quaisquer razões, muitos lógicos matemáticos acham interessante fazê-lo.

Há uma interpretação construtiva informal dos conectivos intuicionistas, usualmente conhecida como interpretação BHK (em homenagem a Brouwer, Heyting e Kolmogorov). Ela funciona assim: uma prova de $\varphi \wedge \psi$ consiste em uma prova de φ emparelhada com uma prova de ψ ; uma prova de $\varphi \vee \psi$ consiste em uma prova de φ , ou uma prova de ψ , onde temos informação explícita sobre qual é o caso; uma prova de $\varphi \rightarrow \psi$ consiste em um procedimento que transforma uma prova de φ em uma prova de ψ ; uma prova de $\forall x \varphi(x)$ consiste em um procedimento que retorna uma prova de $\varphi(x)$ para qualquer valor de x ; e uma prova de $\exists x \varphi(x)$ consiste em um valor de x , juntamente com uma prova de que esse valor satisfaz φ . Pode-se descrever a interpretação em termos computacionais conhecidos como o “isomorfismo de Curry–Howard” ou o “paradigma **fórmulas-como-tipos**”: pense em **a fórmula** como especificando um certo tipo de dado, e provas como objetos computacionais desses tipos de dados que nos permitem ver que **a fórmula** correspondente é verdadeira.

A lógica intuicionista é frequentemente pensada como a lógica clássica “menos” a lei do terceiro excluído. O seguinte teorema torna isso mais preciso.

Teorema 24.2. *Intuicionisticamente, os seguintes esquemas de axiomas são equivalentes:*

1. $(\neg\varphi \rightarrow \perp) \rightarrow \varphi$.
2. $\varphi \vee \neg\varphi$
3. $\neg\neg\varphi \rightarrow \varphi$

Obter instâncias de um esquema a partir de qualquer um dos outros é um bom exercício em lógica intuicionista.

Os primeiros sistemas dedutivos para a lógica proposicional intuicionista, apresentados como formalizações do intuicionismo de Brouwer, são devidos, independentemente, a Kolmogorov, Glivenko e Heyting. A primeira formalização da lógica intuicionista de primeira ordem (e partes da matemática intuicionista) é devida a Heyting. Embora vários esquemas válidos classicamente não sejam válidos intuicionisticamente, muitos o são.

A *tradução de dupla negação* descreve uma relação importante entre a lógica clássica e a intuicionista. Ela é definida indutivamente da seguinte forma (pense em φ^N como a tradução “intuicionista” da fórmula clássica φ):

$$\begin{aligned}\varphi^N &\equiv \neg\neg\varphi \quad \text{para fórmulas atômicas } \varphi \\ (\varphi \wedge \psi)^N &\equiv (\varphi^N \wedge \psi^N) \\ (\varphi \vee \psi)^N &\equiv \neg\neg(\varphi^N \vee \psi^N) \\ (\varphi \rightarrow \psi)^N &\equiv (\varphi^N \rightarrow \psi^N) \\ (\forall x \varphi)^N &\equiv \forall x \varphi^N \\ (\exists x \varphi)^N &\equiv \neg\neg\exists x \varphi^N\end{aligned}$$

Kolmogorov e Glivenko tiveram versões dessa tradução para a lógica proposicional; para a lógica de predicados, é devida a Gödel e Gentzen, independentemente. Temos

Teorema 24.3. 1. $\varphi \leftrightarrow \varphi^N$ é demonstrável classicamente

2. Se φ é demonstrável classicamente, então φ^N é demonstrável intuicionisticamente.

Podemos agora imaginar o seguinte diálogo. Matemático clássico: “Provei φ !” Matemático intuicionista: “Sua prova não é válida. O que você realmente provou é φ^N .” Matemático clássico: “Tudo bem para mim!” Quanto ao matemático clássico, o intuicionista está apenas sendo minucioso, pois os dois são equivalentes. Mas o intuicionista insiste que há uma diferença.

Note que a tradução acima diz respeito apenas à lógica pura; ela não aborda a questão de quais são os axiomas *não lógicos* adequados para a matemática clássica e intuicionista, ou qual é a relação entre eles. Mas a seguinte extensão ligeira do teorema acima fornece alguma informação útil:

24.5. LÓGICA INTUICIONISTA

Teorema 24.4. *Se Γ prova φ classicamente, então Γ^N prova φ^N intuicionisticamente.*

Em outras palavras, se φ é demonstrável a partir de algumas hipóteses classicamente, então φ^N é demonstrável a partir de suas traduções de dupla negação.

Para mostrar que uma sentença ou **fórmula** proposicional é intuicionisticamente válida, basta fornecer uma prova. Mas como se pode mostrar que ela não é válida? Para esse propósito, precisamos de uma semântica que seja correta e, de preferência, completa. Uma semântica devida a Kripke se encaixa bem.

Podemos jogar o mesmo jogo que fizemos para a lógica clássica: definir a semântica e provar correção e completude. Vale a pena, no entanto, notar a seguinte distinção. No caso da lógica clássica, a semântica era a “óbvia”, num sentido implícito no significado dos conectivos. Embora se possa fornecer alguma motivação intuitiva para a semântica de Kripke, esta não oferece o mesmo sentimento de inevitabilidade. Além disso, a noção de **estrutura** clássica é uma noção matemática natural, de modo que podemos ou tomar a noção de **estrutura** como uma ferramenta para estudar a lógica clássica de primeira ordem, ou tomar a lógica clássica de primeira ordem como uma ferramenta para estudar **estruturas** matemáticas. Em contraste, as **estruturas** de Kripke só podem ser vistas como um construto lógico; elas não parecem ter interesse matemático independente.

Uma **estrutura** de Kripke $\mathfrak{M} = \langle W, R, V \rangle$ para uma linguagem proposicional consiste em um conjunto W , uma ordem parcial R em W com um menor **membro**, e uma atribuição “monótona” de variáveis proposicionais aos **membros** de W . A intuição é que os **membros** de W representam “mundos”, ou “estados de conhecimento”; um elemento $v \geq u$ representa um “possível estado futuro” de u ; e as variáveis proposicionais atribuídas a u são as proposições que se sabe serem verdadeiras no estado u . A relação de forçamento $\mathfrak{M}, w \Vdash \varphi$ estende essa relação a **fórmulas** arbitrárias na linguagem; leia $\mathfrak{M}, w \Vdash \varphi$ como “ φ é verdadeira no estado w .” A relação é definida indutivamente, da seguinte forma:

1. $\mathfrak{M}, w \Vdash p_i$ se, e somente se, p_i é uma das variáveis proposicionais atribuídas a w .
2. $\mathfrak{M}, w \nVdash \perp$.
3. $\mathfrak{M}, w \Vdash (\varphi \wedge \psi)$ se, e somente se, $\mathfrak{M}, w \Vdash \varphi$ e $\mathfrak{M}, w \Vdash \psi$.
4. $\mathfrak{M}, w \Vdash (\varphi \vee \psi)$ se, e somente se, $\mathfrak{M}, w \Vdash \varphi$ ou $\mathfrak{M}, w \Vdash \psi$.
5. $\mathfrak{M}, w \Vdash (\varphi \rightarrow \psi)$ se, e somente se, sempre que $w' \geq w$ e $\mathfrak{M}, w' \Vdash \varphi$, então $\mathfrak{M}, w' \Vdash \psi$.

É um bom exercício tentar mostrar que $\neg(p \wedge q) \rightarrow (\neg p \vee \neg q)$ não é intuicionisticamente válida, preparando uma **estrutura** de Kripke que forneça um contraexemplo.

24.6 Lógicas Modais

Considere o seguinte exemplo de uma sentença condicional:

fol:byd:mod:
sec

Se Jeremy está sozinho naquela sala, então ele está bêbado, nu e dançando nas cadeiras.

Este é um exemplo de uma asserção condicional que pode ser materialmente verdadeira, mas ainda assim enganosa, pois parece sugerir que há um vínculo mais forte entre o antecedente e a conclusão do que simplesmente o fato de que ou o antecedente é falso ou a conclusão é verdadeira. Ou seja, a formulação sugere que a afirmação não é apenas verdadeira neste mundo particular (onde pode ser trivialmente verdadeira, porque Jeremy não está sozinho na sala), mas que, além disso, a conclusão *teria sido* verdadeira *se* o antecedente tivesse sido verdadeiro. Em outras palavras, pode-se tomar a asserção como significando que a afirmação é verdadeira não apenas neste mundo, mas em qualquer mundo “possível”; ou que ela é *necessariamente* verdadeira, em oposição a ser apenas verdadeira neste mundo particular.

A lógica modal foi concebida para dar sentido a esse tipo de necessidade. Obtém-se a lógica proposicional modal a partir da lógica proposicional ordinária adicionando um operador caixa; isto é, se φ é a fórmula, então $\Box\varphi$ também o é. Intuitivamente, $\Box\varphi$ afirma que φ é *necessariamente* verdadeira, ou verdadeira em qualquer mundo possível. $\Diamond\varphi$ é usualmente tomada como abreviação de $\neg\Box\neg\varphi$, e pode ser lida como afirmando que φ é *possivelmente* verdadeira. É claro que a modalidade também pode ser adicionada à lógica de predicados.

Estruturas de Kripke podem ser usadas para fornecer uma semântica para a lógica modal; de fato, Kripke projetou essa semântica tendo a lógica modal em mente. Em vez de restringir a ordens parciais, de modo mais geral tem-se um conjunto de “mundos possíveis”, P , e uma relação binária de “acessibilidade” $R(x, y)$ entre mundos. Intuitivamente, $R(p, q)$ afirma que o mundo q é compatível com p ; isto é, se estamos “em” o mundo p , temos que considerar a possibilidade de que o mundo pudesse ter sido como q .

A lógica modal é às vezes chamada de lógica “intensional”, em oposição a uma “extensional”. A semântica pretendida para uma lógica extensional, como a lógica clássica, refere-se apenas a um único mundo, o “atual”; enquanto a semântica para uma lógica “intensional” depende de uma ontologia mais elaborada. Além de estruturar a necessidade, pode-se usar a modalidade para estruturar outras construções linguísticas, reinterpretando $\Box$ e $\Diamond$ de acordo com a aplicação. Por exemplo:

1. Na lógica de demonstrabilidade, $\Box\varphi$ é lida como “ φ é demonstrável” e $\Diamond\varphi$ é lida como “ φ é consistente”.

24.7. OUTRAS LÓGICAS

2. Na lógica epistêmica, pode-se ler $\Box\varphi$ como “eu sei φ ” ou “eu acredito em φ ”.
3. Na lógica temporal, pode-se ler $\Box\varphi$ como “ φ é sempre verdadeira” e $\Diamond\varphi$ como “ φ é às vezes verdadeira”.

Gostaríamos de aumentar a lógica com regras e axiomas que lidem com modalidade. Por exemplo, o sistema **S4** consiste nos axiomas e regras ordinários da lógica proposicional, juntamente com os seguintes axiomas:

$$\begin{aligned}\Box(\varphi \rightarrow \psi) &\rightarrow (\Box\varphi \rightarrow \Box\psi) \\ \Box\varphi &\rightarrow \varphi \\ \Box\varphi &\rightarrow \Box\Box\varphi\end{aligned}$$

assim como uma regra, “de φ conclua $\Box\varphi$.” **S5** adiciona o seguinte axioma:

$$\Diamond\varphi \rightarrow \Box\Diamond\varphi$$

Variações desses axiomas podem ser adequadas para diferentes aplicações; por exemplo, S5 é usualmente tomado para caracterizar a noção de necessidade lógica. E o aspecto agradável é que se pode usualmente encontrar uma semântica para a qual o sistema de **derivação** é correto e completo, restringindo a relação de acessibilidade nas **estruturas** de Kripke de maneiras naturais. Por exemplo, **S4** corresponde à classe de **estruturas** de Kripke em que a relação de acessibilidade é reflexiva e transitiva. **S5** corresponde à classe de **estruturas** de Kripke em que a relação de acessibilidade é *universal*, isto é, todo mundo é acessível a partir de qualquer outro; assim $\Box\varphi$ vale se, e somente se, φ vale em todo mundo.

content/first-order-logic/beyond/other-logics.tex

24.7 Outras Lógicas

fol:byd:oth:
sec

Como você já deve ter percebido, não é difícil projetar uma nova lógica. Você também pode criar a sua própria sintaxe, montar um sistema dedutivo e elaborar uma semântica para acompanhá-la. Talvez você tenha que ser um pouco esperto se quiser que o sistema de **derivação** seja completo para a semântica, e pode exigir algum esforço convencer o mundo em geral de que a sua lógica é realmente interessante. Mas, em troca, você pode desfrutar de horas de diversão sadia, explorando as propriedades matemáticas e computacionais da sua lógica.

As últimas décadas testemunharam uma verdadeira explosão de lógicas formais. A lógica difusa é projetada para modelar o raciocínio sobre propriedades vagas. A lógica probabilística é projetada para modelar o raciocínio sobre a incerteza. As lógicas *default* (de exceções) e as lógicas não monotônicas são projetadas para modelar formas revogáveis de raciocínio, isto é, inferências

CAPÍTULO 24. ALÉM DA LÓGICA DE PRIMEIRA ORDEM

“razoáveis” que podem mais tarde ser derrubadas diante de novas informações. Há lógicas epistêmicas, projetadas para modelar o raciocínio sobre o conhecimento; lógicas causais, projetadas para modelar o raciocínio sobre relações causais; e até mesmo lógicas “deônticas”, que são projetadas para modelar o raciocínio sobre obrigações morais e éticas. Dependendo de a principal motivação para a introdução desses sistemas ser filosófica, matemática ou computacional, você pode encontrar estudos sobre essas criaturas sob a rubrica de lógica matemática, lógica filosófica, inteligência artificial, ciência cognitiva ou outro lugar.

A lista não tem fim, e as possibilidades parecem infinitas. Talvez nunca alcancemos o sonho de Leibniz de reduzir toda a razão humana a cálculo — mas isso não pode nos impedir de tentar.

Parte IV

Model Theory

Material on model theory is incomplete and experimental. It is currently simply an adaptation of Aldo Antonelli's notes on model theory, less those topics covered in the part on first-order logic (theories, completeness, compactness). It requires much more introduction, motivation, and explanation, as well as exercises, to be useful for a textbook. Andy Arana is at planning to work on this part specifically ([issue #65](#)).

Capítulo 25

Basics of Model Theory

-NoValue-

[content/model-theory/basics/reducts-and-expansions.tex](#)

25.1 Reducts and Expansions

Often it is useful or necessary to compare languages which have symbols in common, as well as *estruturas* for these languages. The most common case is when all the symbols in a *linguagem* $\mathcal{L}$ are also part of a *linguagem* $\mathcal{L}'$, O.s., $\mathcal{L} \subseteq \mathcal{L}'$. An $\mathcal{L}$ -*estrutura* $\mathfrak{M}$ can then always be expanded to an $\mathcal{L}'$ -*estrutura* by adding interpretations of the additional symbols while leaving the interpretations of the common symbols the same. On the other hand, from an $\mathcal{L}'$ -structure $\mathfrak{M}'$ we can obtain an $\mathcal{L}$ -structure simply by “forgetting” the interpretations of the symbols that do not occur in $\mathcal{L}$.

mod:bas:red: defn:reduct **Definição 25.1.** Suppose $\mathcal{L} \subseteq \mathcal{L}'$, $\mathfrak{M}$ is an $\mathcal{L}$ -*estrutura* and $\mathfrak{M}'$ is an $\mathcal{L}'$ -*estrutura*. $\mathfrak{M}$ is the *reduct* of $\mathfrak{M}'$ to $\mathcal{L}$, and $\mathfrak{M}'$ is an *expansion* of $\mathfrak{M}$ to $\mathcal{L}'$ iff

1. $|\mathfrak{M}| = |\mathfrak{M}'|$

2. For every **símbolo de constante** $c \in \mathcal{L}$, $c^{\mathfrak{M}} = c^{\mathfrak{M}'}$.
3. For every **símbolo de função** $f \in \mathcal{L}$, $f^{\mathfrak{M}} = f^{\mathfrak{M}'}$.
4. For every **símbolo de predicado** $P \in \mathcal{L}$, $P^{\mathfrak{M}} = P^{\mathfrak{M}'}$.

Proposição 25.2. *If an $\mathcal{L}$ -estrutura $\mathfrak{M}$ is a reduct of an $\mathcal{L}'$ -estrutura $\mathfrak{M}'$, mod:bas:red: prop:red:reduct then for all $\mathcal{L}$ -sentenças φ ,*

$$\mathfrak{M} \models \varphi \text{ iff } \mathfrak{M}' \models \varphi.$$

Prova. Exercise. □

Problema 25.1. Prove **Proposição 25.2**.

Definição 25.3. When we have an $\mathcal{L}$ -structure $\mathfrak{M}$, and $\mathcal{L}' = \mathcal{L} \cup \{P\}$ is the expansion of $\mathcal{L}$ obtained by adding a single n -place **símbolo de predicado** P , and $R \subseteq |\mathfrak{M}|^n$ is an n -place relation, then we write $(\mathfrak{M}, R)$ for the expansion $\mathfrak{M}'$ of $\mathfrak{M}$ with $P^{\mathfrak{M}'} = R$.

-NoValue-

[content/model-theory/basics/substructures.tex](#)

25.2 Subestruturas

The **domínio** of a **estrutura** $\mathfrak{M}$ may be a subset of another $\mathfrak{M}'$. But we should obviously only consider $\mathfrak{M}$ a “part” of $\mathfrak{M}'$ if not only $|\mathfrak{M}| \subseteq |\mathfrak{M}'|$, but $\mathfrak{M}$ and $\mathfrak{M}'$ “agree” in how they interpret the symbols of the language at least on the shared part $|\mathfrak{M}|$. mod:bas:sub: sec

Definição 25.4. Given **estruturas** $\mathfrak{M}$ and $\mathfrak{M}'$ for the same language $\mathcal{L}$, we say that $\mathfrak{M}$ is a **subestrutura** of $\mathfrak{M}'$, and $\mathfrak{M}'$ an *extension* of $\mathfrak{M}$, written $\mathfrak{M} \subseteq \mathfrak{M}'$, iff mod:bas:sub: defn:substructure

1. $|\mathfrak{M}| \subseteq |\mathfrak{M}'|$,
2. For each constant $c \in \mathcal{L}$, $c^{\mathfrak{M}} = c^{\mathfrak{M}'}$;
3. For each n -place **símbolo de função** $f \in \mathcal{L}$ $f^{\mathfrak{M}}(a_1, \dots, a_n) = f^{\mathfrak{M}'}(a_1, \dots, a_n)$ for all $a_1, \dots, a_n \in |\mathfrak{M}|$.
4. For each n -place **símbolo de predicado** $R \in \mathcal{L}$, $\langle a_1, \dots, a_n \rangle \in R^{\mathfrak{M}}$ iff $\langle a_1, \dots, a_n \rangle \in R^{\mathfrak{M}'}$ for all $a_1, \dots, a_n \in |\mathfrak{M}|$.

Observação 1. If the language contains no constant or **símbolos de função**, then any $N \subseteq |\mathfrak{M}|$ determines a **subestrutura** $\mathfrak{N}$ of $\mathfrak{M}$ with **domínio** $|\mathfrak{N}| = N$ by putting $R^{\mathfrak{N}} = R^{\mathfrak{M}} \cap N^n$. mod:bas:sub: rem:substructure

-NoValue-

[content/model-theory/basics/overspill.tex](#)

25.3. OVERSPILL

25.3 Overspill

mod:bas:ove:
 mod:bas:ove:
 sec
 overspill

Teorema 25.5. *If a set Γ of sentences has arbitrarily large finite models, then it has an infinite model.*

Prova. Expand the language of Γ by adding countably many new constants $c_0, c_1, \dots$ and consider the set $\Gamma \cup \{c_i \neq c_j : i \neq j\}$. To say that Γ has arbitrarily large finite models means that for every $m > 0$ there is $n \geq m$ such that Γ has a model of cardinality n . This implies that $\Gamma \cup \{c_i \neq c_j : i \neq j\}$ is finitely satisfiable. By compactness, $\Gamma \cup \{c_i \neq c_j : i \neq j\}$ has a model $\mathfrak{M}$ whose domain must be infinite, since it satisfies all inequalities $c_i \neq c_j$. $\square$

mod:bas:ove:
 inf-not-fo

Proposição 25.6. *There is no sentence φ of any first-order language that is true in a estrutura $\mathfrak{M}$ if and only if the domain $|\mathfrak{M}|$ of the estrutura is infinite.*

Prova. If there were such a φ , its negation $\neg\varphi$ would be true in all and only the finite estruturas, and it would therefore have arbitrarily large finite models but it would lack an infinite model, contradicting Teorema 25.5. $\square$

-NoValue-

content/model-theory/basics/isomorphism.tex

25.4 Isomorphic Structures

mod:bas:iso:
 sec

First-order estruturas can be alike in one of two ways. One way in which they can be alike is that they make the same sentenças true. We call such estruturas *elementarily equivalent*. But structures can be very different and still make the same sentenças true—for instance, one can be enumerável and the other not. This is because there are lots of features of a estrutura that cannot be expressed in first-order languages, either because the language is not rich enough, or because of fundamental limitations of first-order logic such as the Löwenheim–Skolem theorem. So another, stricter, aspect in which estruturas can be alike is if they are fundamentally the same, in the sense that they only differ in the objects that make them up, but not in their structural features. A way of making this precise is by the notion of an *isomorphism*.

mod:bas:iso:
 defn:elem-equiv

Definição 25.7. Given two estruturas $\mathfrak{M}$ and $\mathfrak{M}'$ for the same linguagem $\mathcal{L}$, we say that $\mathfrak{M}$ is *elementarily equivalent* to $\mathfrak{M}'$, written $\mathfrak{M} \equiv \mathfrak{M}'$, if and only if for every sentença φ of $\mathcal{L}$, $\mathfrak{M} \models \varphi$ iff $\mathfrak{M}' \models \varphi$.

mod:bas:iso:
 defn:isomorphism

Definição 25.8. Given two estruturas $\mathfrak{M}$ and $\mathfrak{M}'$ for the same linguagem $\mathcal{L}$, we say that $\mathfrak{M}$ is *isomorphic* to $\mathfrak{M}'$, written $\mathfrak{M} \simeq \mathfrak{M}'$, if and only if there is a function $h: |\mathfrak{M}| \rightarrow |\mathfrak{M}'|$ such that:

1. h is *injetiva*: if $h(x) = h(y)$ then $x = y$;
2. h is *sobrejetiva*: for every $y \in |\mathfrak{M}'|$ there is $x \in |\mathfrak{M}|$ such that $h(x) = y$;

3. for every **símbolo de constante** c : $h(c^{\mathfrak{M}}) = c^{\mathfrak{M}'}$;

mod:bas:iso:
defn:iso-const

4. for every n -place **símbolo de predicado** P :

mod:bas:iso:
defn:iso-pred

$$\langle a_1, \dots, a_n \rangle \in P^{\mathfrak{M}} \quad \text{iff} \quad \langle h(a_1), \dots, h(a_n) \rangle \in P^{\mathfrak{M}'};$$

5. for every n -place **símbolo de função** f :

mod:bas:iso:
defn:iso-func

$$h(f^{\mathfrak{M}}(a_1, \dots, a_n)) = f^{\mathfrak{M}'}(h(a_1), \dots, h(a_n)).$$

Teorema 25.9. *If $\mathfrak{M} \simeq \mathfrak{M}'$ then $\mathfrak{M} \equiv \mathfrak{M}'$.*

mod:bas:iso:
thm:isom

Prova. Let h be an isomorphism of $\mathfrak{M}$ onto $\mathfrak{M}'$. For any assignment s , $h \circ s$ is the composition of h and s , O.s., the assignment in $\mathfrak{M}'$ such that $(h \circ s)(x) = h(s(x))$. By induction on t and φ one can prove the stronger claims:

a. $h(\text{Val}_s^{\mathfrak{M}}(t)) = \text{Val}_{h \circ s}^{\mathfrak{M}'}(t)$.

b. $\mathfrak{M}, s \models \varphi$ iff $\mathfrak{M}', h \circ s \models \varphi$.

The first is proved by induction on the complexity of t .

1. If $t \equiv c$, then $\text{Val}_s^{\mathfrak{M}}(c) = c^{\mathfrak{M}}$ and $\text{Val}_{h \circ s}^{\mathfrak{M}'}(c) = c^{\mathfrak{M}'}$. Thus, $h(\text{Val}_s^{\mathfrak{M}}(t)) = h(c^{\mathfrak{M}}) = c^{\mathfrak{M}'}$ (by (3) of Definição 25.8) $= \text{Val}_{h \circ s}^{\mathfrak{M}'}(t)$.

2. If $t \equiv x$, then $\text{Val}_s^{\mathfrak{M}}(x) = s(x)$ and $\text{Val}_{h \circ s}^{\mathfrak{M}'}(x) = h(s(x))$. Thus, $h(\text{Val}_s^{\mathfrak{M}}(x)) = h(s(x)) = \text{Val}_{h \circ s}^{\mathfrak{M}'}(x)$.

3. If $t \equiv f(t_1, \dots, t_n)$, then

$$\text{Val}_s^{\mathfrak{M}}(t) = f^{\mathfrak{M}}(\text{Val}_s^{\mathfrak{M}}(t_1), \dots, \text{Val}_s^{\mathfrak{M}}(t_n)) \quad \text{and}$$

$$\text{Val}_{h \circ s}^{\mathfrak{M}'}(t) = f^{\mathfrak{M}'}(\text{Val}_{h \circ s}^{\mathfrak{M}'}(t_1), \dots, \text{Val}_{h \circ s}^{\mathfrak{M}'}(t_n)).$$

The induction hypothesis is that for each i , $h(\text{Val}_s^{\mathfrak{M}}(t_i)) = \text{Val}_{h \circ s}^{\mathfrak{M}'}(t_i)$. So,

$$\begin{aligned} h(\text{Val}_s^{\mathfrak{M}}(t)) &= h(f^{\mathfrak{M}}(\text{Val}_s^{\mathfrak{M}}(t_1), \dots, \text{Val}_s^{\mathfrak{M}}(t_n))) \\ &= f^{\mathfrak{M}'}(h(\text{Val}_s^{\mathfrak{M}}(t_1)), \dots, h(\text{Val}_s^{\mathfrak{M}}(t_n))) \end{aligned} \quad (25.1)$$

$$= f^{\mathfrak{M}'}(\text{Val}_{h \circ s}^{\mathfrak{M}'}(t_1), \dots, \text{Val}_{h \circ s}^{\mathfrak{M}'}(t_n)) \quad (25.2)$$

$$= \text{Val}_{h \circ s}^{\mathfrak{M}'}(t)$$

mod:bas:iso:
iso-1
mod:bas:iso:
iso-2

Here, Eq. (25.1) follows by (5) of Definição 25.8 and Eq. (25.2) by induction hypothesis.

Part (b) is left as an exercise.

If φ is a sentence, the assignments s and $h \circ s$ are irrelevant, and we have $\mathfrak{M} \models \varphi$ iff $\mathfrak{M}' \models \varphi$. $\square$

25.5. THE THEORY OF A ESTRUTURA

Problema 25.2. Carry out the proof of (b) of [Teorema 25.9](#) in detail. Make sure to note where each of the five properties characterizing isomorphisms of [Definição 25.8](#) is used.

Definição 25.10. An *automorphism* of a structure $\mathfrak{M}$ is an isomorphism of $\mathfrak{M}$ onto itself.

Problema 25.3. Show that for any structure $\mathfrak{M}$, if X is a definable subset of $\mathfrak{M}$, and h is an automorphism of $\mathfrak{M}$, then $X = \{h(x) : x \in X\}$ (O.s., X is fixed under h).

-NoValue-

content/model-theory/basics/theory-of-m.tex

25.5 The Theory of a Estrutura

Every [estrutura](#) $\mathfrak{M}$ makes some [sentença](#) true, and some false. The set of all the [sentenças](#) it makes true is called its *theory*. That set is in fact a theory, since anything it entails must be true in all its models, including $\mathfrak{M}$.

Definição 25.11. Given a [estrutura](#) $\mathfrak{M}$, the *theory* of $\mathfrak{M}$ is the set $\text{Th}(\mathfrak{M})$ of [sentenças](#) that are true in $\mathfrak{M}$, O.s., $\text{Th}(\mathfrak{M}) = \{\varphi : \mathfrak{M} \models \varphi\}$.

We also use the term “theory” informally to refer to sets of [sentenças](#) having an intended interpretation, whether deductively closed or not.

Proposição 25.12. For any $\mathfrak{M}$, $\text{Th}(\mathfrak{M})$ is complete.

Prova. For any [sentença](#) φ either $\mathfrak{M} \models \varphi$ or $\mathfrak{M} \models \neg\varphi$, so either $\varphi \in \text{Th}(\mathfrak{M})$ or $\neg\varphi \in \text{Th}(\mathfrak{M})$. □

mod:bas:thm: prop:equiv **Proposição 25.13.** If $\mathfrak{N} \models \varphi$ for every $\varphi \in \text{Th}(\mathfrak{M})$, then $\mathfrak{M} \equiv \mathfrak{N}$.

Prova. Since $\mathfrak{N} \models \varphi$ for all $\varphi \in \text{Th}(\mathfrak{M})$, $\text{Th}(\mathfrak{M}) \subseteq \text{Th}(\mathfrak{N})$. If $\mathfrak{N} \models \varphi$, then $\mathfrak{N} \not\models \neg\varphi$, so $\neg\varphi \notin \text{Th}(\mathfrak{M})$. Since $\text{Th}(\mathfrak{M})$ is complete, $\varphi \in \text{Th}(\mathfrak{M})$. So, $\text{Th}(\mathfrak{N}) \subseteq \text{Th}(\mathfrak{M})$, and we have $\mathfrak{M} \equiv \mathfrak{N}$. □

mod:bas:thm: remark:R **Observação 2.** Consider $\mathfrak{R} = \langle \mathbb{R}, < \rangle$, the [estrutura](#) whose domain is the set $\mathbb{R}$ of the real numbers, in the [linguagem](#) comprising only a 2-place [símbolo de predicado](#) interpreted as the $<$ relation over the reals. Clearly $\mathfrak{R}$ is [não enumerável](#); however, since $\text{Th}(\mathfrak{R})$ is obviously consistent, by the Löwenheim–Skolem theorem it has a [enumerável](#) model, say $\mathfrak{S}$, and by [Proposição 25.13](#), $\mathfrak{R} \equiv \mathfrak{S}$. Moreover, since $\mathfrak{R}$ and $\mathfrak{S}$ are not isomorphic, this shows that the converse of [Teorema 25.9](#) fails in general.

-NoValue-

content/model-theory/basics/partial-iso.tex

25.6 Partial Isomorphisms

Definição 25.14. Given two **estruturas** $\mathfrak{M}$ and $\mathfrak{N}$, a *partial isomorphism* from $\mathfrak{M}$ to $\mathfrak{N}$ is a finite partial function p taking arguments in $|\mathfrak{M}|$ and returning values in $|\mathfrak{N}|$, which satisfies the isomorphism conditions from **Definição 25.8** on its domain:

1. p is **injetiva**;
2. for every **símbolo de constante** c : if $p(c^{\mathfrak{M}})$ is defined, then $p(c^{\mathfrak{M}}) = c^{\mathfrak{N}}$;
3. for every n -place **símbolo de predicado** P : if $a_1, \dots, a_n$ are in the domain of p , then $\langle a_1, \dots, a_n \rangle \in P^{\mathfrak{M}}$ if and only if $\langle p(a_1), \dots, p(a_n) \rangle \in P^{\mathfrak{N}}$;
4. for every n -place **símbolo de função** f : if $a_1, \dots, a_n$ are in the domain of p , then $p(f^{\mathfrak{M}}(a_1, \dots, a_n)) = f^{\mathfrak{N}}(p(a_1), \dots, p(a_n))$.

That p is finite means that $\text{dom}(p)$ is finite.

Notice that the empty function $\emptyset$ is always a partial isomorphism between any two **estruturas**.

Definição 25.15. Two **estruturas** $\mathfrak{M}$ and $\mathfrak{N}$, are *partially isomorphic*, written $\mathfrak{M} \simeq_p \mathfrak{N}$, if and only if there is a non-empty set I of partial isomorphisms between $\mathfrak{M}$ and $\mathfrak{N}$ satisfying the *back-and-forth* property:

mod:bas:pis:
defn:partialisom

1. (*Forth*) For every $p \in I$ and $a \in |\mathfrak{M}|$ there is $q \in I$ such that $p \subseteq q$ and a is in the domain of q ;
2. (*Back*) For every $p \in I$ and $b \in |\mathfrak{N}|$ there is $q \in I$ such that $p \subseteq q$ and b is in the range of q .

Teorema 25.16. If $\mathfrak{M} \simeq_p \mathfrak{N}$ and $\mathfrak{M}$ and $\mathfrak{N}$ are **enumerável**, then $\mathfrak{M} \simeq \mathfrak{N}$.

mod:bas:pis:
thm:p-isom1

Prova. Since $\mathfrak{M}$ and $\mathfrak{N}$ are **enumerável**, let $|\mathfrak{M}| = \{a_0, a_1, \dots\}$ and $|\mathfrak{N}| = \{b_0, b_1, \dots\}$. Starting with an arbitrary $p_0 \in I$, we define an increasing sequence of partial isomorphisms $p_0 \subseteq p_1 \subseteq p_2 \subseteq \dots$ as follows:

1. if $n + 1$ is odd, say $n = 2r$, then using the Forth property find a $p_{n+1} \in I$ such that $p_n \subseteq p_{n+1}$ and a_r is in the domain of p_{n+1} ;
2. if $n + 1$ is even, say $n + 1 = 2r$, then using the Back property find a $p_{n+1} \in I$ such that $p_n \subseteq p_{n+1}$ and b_r is in the range of p_{n+1} .

If we now put:

$$p = \bigcup_{n \geq 0} p_n,$$

we have that p is an isomorphism between $\mathfrak{M}$ and $\mathfrak{N}$. □

25.6. PARTIAL ISOMORPHISMS

Problema 25.4. Show in detail that p as defined in Teorema 25.16 is in fact an isomorphism.

*mod:bas:pis:
thm:p-isom2* **Teorema 25.17.** Suppose $\mathfrak{M}$ and $\mathfrak{N}$ are *estruturas* for a purely relational *linguagem* (a *linguagem* containing only *símbolos de predicado*, and no *símbolos de função* or constants). Then if $\mathfrak{M} \simeq_p \mathfrak{N}$, also $\mathfrak{M} \equiv \mathfrak{N}$.

Prova. By induction on *fórmulas*, one shows that if $a_1, \dots, a_n$ and $b_1, \dots, b_n$ are such that there is a partial isomorphism p mapping each a_i to b_i and $s_1(x_i) = a_i$ and $s_2(x_i) = b_i$ (for $i = 1, \dots, n$), then $\mathfrak{M}, s_1 \models \varphi$ if and only if $\mathfrak{N}, s_2 \models \varphi$. The case for $n = 0$ gives $\mathfrak{M} \equiv \mathfrak{N}$. $\square$

Observação 3. If *símbolos de função* are present, the previous result is still true, but one needs to consider the isomorphism induced by p between the *subestrutura* of $\mathfrak{M}$ generated by $a_1, \dots, a_n$ and the *subestrutura* of $\mathfrak{N}$ generated by $b_1, \dots, b_n$.

The previous result can be “broken down” into stages by establishing a connection between the number of nested quantifiers in a *fórmula* and how many times the relevant partial isomorphisms can be extended.

Definição 25.18. For any *fórmula* φ , the *quantifier rank* of φ , denoted by $qr(\varphi) \in \mathbb{N}$, is recursively defined as the highest number of nested quantifiers in φ . Two *estruturas* $\mathfrak{M}$ and $\mathfrak{N}$ are *n-equivalent*, written $\mathfrak{M} \equiv_n \mathfrak{N}$, if they agree on all sentences of quantifier rank less than or equal to n .

*mod:bas:pis:
prop:qr-finite* **Proposição 25.19.** Let $\mathcal{L}$ be a finite purely relational *linguagem*, O.s., a *linguagem* containing finitely many *símbolos de predicado* and *símbolos de constante*, and no *símbolos de função*. Then for each $n \in \mathbb{N}$ there are only finitely many first-order *sentenças* in the *linguagem* $\mathcal{L}$ that have quantifier rank no greater than n , up to logical equivalence.

Prova. By induction on n . $\square$

Definição 25.20. Given a *estrutura* $\mathfrak{M}$, let $|\mathfrak{M}|^{<\omega}$ be the set of all finite sequences over $|\mathfrak{M}|$. We use $\mathbf{a}, \mathbf{b}, \mathbf{c}, \dots$ to range over finite sequences of elements. If $\mathbf{a} \in |\mathfrak{M}|^{<\omega}$ and $a \in |\mathfrak{M}|$, then $\mathbf{a}a$ represents the *concatenation* of $\mathbf{a}$ with a .

Definição 25.21. Given *estruturas* $\mathfrak{M}$ and $\mathfrak{N}$, we define relations $I_n \subseteq |\mathfrak{M}|^{<\omega} \times |\mathfrak{N}|^{<\omega}$ between sequences of equal length, by recursion on n as follows:

1. $I_0(\mathbf{a}, \mathbf{b})$ if and only if $\mathbf{a}$ and $\mathbf{b}$ satisfy the same atomic *fórmulas* in $\mathfrak{M}$ and $\mathfrak{N}$; O.s., if $s_1(x_i) = a_i$ and $s_2(x_i) = b_i$ and φ is atomic with all *variáveis* among $x_1, \dots, x_n$, then $\mathfrak{M}, s_1 \models \varphi$ if and only if $\mathfrak{N}, s_2 \models \varphi$.
2. $I_{n+1}(\mathbf{a}, \mathbf{b})$ if and only if for every $a \in |\mathfrak{M}|$ there is a $b \in |\mathfrak{N}|$ such that $I_n(\mathbf{a}a, \mathbf{b}b)$, and vice-versa.

Definição 25.22. Write $\mathfrak{M} \approx_n \mathfrak{N}$ if $I_n(A, A)$ holds of $\mathfrak{M}$ and $\mathfrak{N}$ (where A is the empty sequence).

Teorema 25.23. Let $\mathcal{L}$ be a purely relational *linguagem*. Then $I_n(\mathbf{a}, \mathbf{b})$ implies that for every φ such that $\text{qr}(\varphi) \leq n$, we have $\mathfrak{M}, \mathbf{a} \models \varphi$ if and only if $\mathfrak{N}, \mathbf{b} \models \varphi$ (where again $\mathbf{a}$ satisfies φ if any s such that $s(x_i) = a_i$ satisfies φ). Moreover, if $\mathcal{L}$ is finite, the converse also holds. *mod:bas:pis: thm:b-n-f*

Prova. The proof that $I_n(\mathbf{a}, \mathbf{b})$ implies that $\mathbf{a}$ and $\mathbf{b}$ satisfy the same *fórmulas* of quantifier rank no greater than n is by an easy induction on φ . For the converse we proceed by induction on n , using **Proposição 25.19**, which ensures that for each n there are at most finitely many non-equivalent *fórmulas* of that quantifier rank.

For $n = 0$ the hypothesis that $\mathbf{a}$ and $\mathbf{b}$ satisfy the same quantifier-free *fórmulas* gives that they satisfy the same atomic ones, so that $I_0(\mathbf{a}, \mathbf{b})$.

For the $n + 1$ case, suppose that $\mathbf{a}$ and $\mathbf{b}$ satisfy the same *fórmulas* of quantifier rank no greater than $n + 1$; in order to show that $I_{n+1}(\mathbf{a}, \mathbf{b})$ suffices to show that for each $a \in |\mathfrak{M}|$ there is a $b \in |\mathfrak{N}|$ such that $I_n(\mathbf{aa}, \mathbf{bb})$, and by the inductive hypothesis again suffices to show that for each $a \in |\mathfrak{M}|$ there is a $b \in |\mathfrak{N}|$ such that $\mathbf{aa}$ and $\mathbf{bb}$ satisfy the same *fórmulas* of quantifier rank no greater than n .

Given $a \in |\mathfrak{M}|$, let τ_n^a be set of *fórmulas* $\psi(x, \mathbf{y})$ of rank no greater than n satisfied by $\mathbf{aa}$ in $\mathfrak{M}$; τ_n^a is finite, so we can assume it is a single first-order *fórmula*. It follows that $\mathbf{a}$ satisfies $\exists x \tau_n^a(x, \mathbf{y})$, which has quantifier rank no greater than $n + 1$. By hypothesis $\mathbf{b}$ satisfies the same *fórmula* in $\mathfrak{N}$, so that there is a $b \in |\mathfrak{N}|$ such that $\mathbf{bb}$ satisfies τ_n^a ; in particular, $\mathbf{bb}$ satisfies the same *fórmulas* of quantifier rank no greater than n as $\mathbf{aa}$. Similarly one shows that for every $b \in |\mathfrak{N}|$ there is $a \in |\mathfrak{M}|$ such that $\mathbf{aa}$ and $\mathbf{bb}$ satisfy the same *fórmulas* of quantifier rank no greater than n , which completes the proof. $\square$

Corolário 25.24. If $\mathfrak{M}$ and $\mathfrak{N}$ are purely relational *estruturas* in a finite *linguagem*, then $\mathfrak{M} \approx_n \mathfrak{N}$ if and only if $\mathfrak{M} \equiv_n \mathfrak{N}$. In particular $\mathfrak{M} \equiv \mathfrak{N}$ if and only if for each n , $\mathfrak{M} \approx_n \mathfrak{N}$. *mod:bas:pis: cor:b-n-f*

-NoValue-

content/model-theory/basics/dlo.tex

25.7 Dense Linear Orders

Definição 25.25. A dense linear ordering without endpoints is a *estrutura* $\mathfrak{M}$ for the *linguagem* containing a single 2-place *símbolo de predicado* $<$ satisfying the following sentences:

1. $\forall x \neg x < x$;
2. $\forall x \forall y \forall z (x < y \rightarrow (y < z \rightarrow x < z))$;

3. $\forall x \forall y (x < y \vee x = y \vee y < x)$;
4. $\forall x \exists y x < y$;
5. $\forall x \exists y y < x$;
6. $\forall x \forall y (x < y \rightarrow \exists z (x < z \wedge z < y))$.

mod:bas:dlo: **Teorema 25.26.** *Any two enumerável dense linear orderings without endpoints are isomorphic.*
thm:cantorQ

Prova. Let $\mathfrak{M}_1$ and $\mathfrak{M}_2$ be enumerável dense linear orderings without endpoints, with $<_1 = <^{\mathfrak{M}_1}$ and $<_2 = <^{\mathfrak{M}_2}$, and let $\mathcal{I}$ be the set of all partial isomorphisms between them. $\mathcal{I}$ is not empty since at least $\emptyset \in \mathcal{I}$. We show that $\mathcal{I}$ satisfies the Back-and-Forth property. Then $\mathfrak{M}_1 \simeq_p \mathfrak{M}_2$, and the theorem follows by Teorema 25.16.

To show $\mathcal{I}$ satisfies the Forth property, let $p \in \mathcal{I}$ and let $p(a_i) = b_i$ for $i = 1, \dots, n$, and without loss of generality suppose $a_1 <_1 a_2 <_1 \dots <_1 a_n$. Given $a \in |\mathfrak{M}_1|$, find $b \in |\mathfrak{M}_2|$ as follows:

1. if $a <_1 a_1$ let $b \in |\mathfrak{M}_2|$ be such that $b <_2 b_1$;
2. if $a_n <_1 a$ let $b \in |\mathfrak{M}_2|$ be such that $b_n <_2 b$;
3. if $a_i <_1 a <_1 a_{i+1}$ for some i , then let $b \in |\mathfrak{M}_2|$ be such that $b_i <_2 b <_2 b_{i+1}$.

It is always possible to find a b with the desired property since $\mathfrak{M}_2$ is a dense linear ordering without endpoints. Define $q = p \cup \{\langle a, b \rangle\}$ so that $q \in \mathcal{I}$ is the desired extension of p . This establishes the Forth property. The Back property is similar. So $\mathfrak{M}_1 \simeq_p \mathfrak{M}_2$; by Teorema 25.16, $\mathfrak{M}_1 \simeq \mathfrak{M}_2$. $\square$

Problema 25.5. Complete the proof of Teorema 25.26 by verifying that $\mathcal{I}$ satisfies the Back property.

Observação 4. Let $\mathfrak{S}$ be any enumerável dense linear ordering without endpoints. Then (by Teorema 25.26) $\mathfrak{S} \simeq \mathfrak{Q}$, where $\mathfrak{Q} = (\mathbb{Q}, <)$ is the enumerável dense linear ordering having the set $\mathbb{Q}$ of the rational numbers as its domain. Now consider again the estrutura $\mathfrak{R} = (\mathbb{R}, <)$ from Observação 2. We saw that there is a enumerável estrutura $\mathfrak{S}$ such that $\mathfrak{R} \equiv \mathfrak{S}$. But $\mathfrak{S}$ is a enumerável dense linear ordering without endpoints, and so it is isomorphic (and hence elementarily equivalent) to the estrutura $\mathfrak{Q}$. By transitivity of elementary equivalence, $\mathfrak{R} \equiv \mathfrak{Q}$. (We could have shown this directly by establishing $\mathfrak{R} \simeq_p \mathfrak{Q}$ by the same back-and-forth argument.)

Capítulo 26

Models of Arithmetic

-NoValue-

content/model-theory/models-of-arithmetic/introduction.tex

26.1 Introduction

The *standard model* of arithmetic is the **estrutura** $\mathfrak{N}$ with $|\mathfrak{N}| = \mathbb{N}$ in which $\mathfrak{o}$, $\mathfrak{!}$, $+$, $\times$, and $<$ are interpreted as you would expect. That is, $\mathfrak{o}$ is 0, $\mathfrak{!}$ is the successor function, $+$ is interpreted as addition and $\times$ as multiplication of the numbers in $\mathbb{N}$. Specifically,

$$\begin{aligned}\mathfrak{o}^{\mathfrak{N}} &= 0 \\ \mathfrak{!}^{\mathfrak{N}}(n) &= n + 1 \\ +^{\mathfrak{N}}(n, m) &= n + m \\ \times^{\mathfrak{N}}(n, m) &= nm\end{aligned}$$

Of course, there are structures for $\mathcal{L}_A$ that have domains other than $\mathbb{N}$. For instance, we can take $\mathfrak{M}$ with domain $|\mathfrak{M}| = \{a\}^*$ (the finite sequences of the single symbol a , O.s., $\emptyset$, a , aa , aaa , $\dots$), and interpretations

$$\begin{aligned}\mathfrak{o}^{\mathfrak{M}} &= \emptyset \\ \mathfrak{!}^{\mathfrak{M}}(s) &= s \frown a \\ +^{\mathfrak{M}}(n, m) &= a^{n+m} \\ \times^{\mathfrak{M}}(n, m) &= a^{nm}\end{aligned}$$

These two structures are “essentially the same” in the sense that the only difference is the **membros** of the **domínios** but not how the **membros** of the **domínios** are related among each other by the interpretation functions. We say that the two **estruturas** are *isomorphic*.

It is an easy consequence of the compactness theorem that any theory true in $\mathfrak{N}$ also has models that are not isomorphic to $\mathfrak{N}$. Such structures are called

26.2. STANDARD MODELS OF ARITHMETIC

non-standard. The interesting thing about them is that while the **membros** of a standard model (O.s., $\mathfrak{N}$, but also all **estruturas** isomorphic to it) are exhausted by the values of the standard numerals $\bar{n}$, O.s.,

$$|\mathfrak{N}| = \{\text{Val}^{\mathfrak{N}}(\bar{n}) : n \in \mathbb{N}\}$$

that isn't the case in non-standard models: if $\mathfrak{M}$ is non-standard, then there is at least one $x \in |\mathfrak{M}|$ such that $x \neq \text{Val}^{\mathfrak{M}}(\bar{n})$ for all n .

These non-standard elements are pretty neat: they are “infinite natural numbers.” But their existence also explains, in a sense, the incompleteness phenomena. Consider an example, p.ex., the consistency statement for Peano arithmetic, $\text{Con}_{\mathbf{PA}}$, O.s., $\neg \exists x \text{Prf}_{\mathbf{PA}}(x, \ulcorner \perp \urcorner)$. Since $\mathbf{PA}$ neither proves $\text{Con}_{\mathbf{PA}}$ nor $\neg \text{Con}_{\mathbf{PA}}$, either can be consistently added to $\mathbf{PA}$. Since $\mathbf{PA}$ is consistent, $\mathfrak{N} \models \text{Con}_{\mathbf{PA}}$, and consequently $\mathfrak{N} \not\models \neg \text{Con}_{\mathbf{PA}}$. So $\mathfrak{N}$ is *not* a model of $\mathbf{PA} \cup \{\neg \text{Con}_{\mathbf{PA}}\}$, and all its models must be nonstandard. Models of $\mathbf{PA} \cup \{\neg \text{Con}_{\mathbf{PA}}\}$ must contain some **membro** that serves as the witness that makes $\exists x \text{Prf}_{\mathbf{PA}}(\ulcorner \perp \urcorner)$ true, O.s., a Gödel number of a **derivação** of a contradiction from $\mathbf{PA}$. Such **um membro** can't be standard—since $\mathbf{PA} \vdash \neg \text{Prf}_{\mathbf{PA}}(\bar{n}, \ulcorner \perp \urcorner)$ for every n .

-NoValue-

[content/model-theory/models-of-arithmetic/standard-models.tex](#)

26.2 Standard Models of Arithmetic

mod:mar:stm:
sec

The language of arithmetic $\mathcal{L}_A$ is obviously intended to be about numbers, specifically, about natural numbers. So, “the” standard model $\mathfrak{N}$ is special: it is the model we want to talk about. But in logic, we are often just interested in structural properties, and any two **estruturas** that are isomorphic share those. So we can be a bit more liberal, and consider any **estrutura** that is isomorphic to $\mathfrak{N}$ “standard.”

Definição 26.1. A **estrutura** for $\mathcal{L}_A$ is *standard* if it is isomorphic to $\mathfrak{N}$.

mod:mar:stm:
prop:standard-domain

Proposição 26.2. If a **estrutura** $\mathfrak{M}$ is standard, then its domain is the set of values of the standard numerals, O.s.,

$$|\mathfrak{M}| = \{\text{Val}^{\mathfrak{M}}(\bar{n}) : n \in \mathbb{N}\}$$

Prova. Clearly, every $\text{Val}^{\mathfrak{M}}(\bar{n}) \in |\mathfrak{M}|$. We just have to show that every $x \in |\mathfrak{M}|$ is equal to $\text{Val}^{\mathfrak{M}}(\bar{n})$ for some n . Since $\mathfrak{M}$ is standard, it is isomorphic to $\mathfrak{N}$. Suppose $g: \mathbb{N} \rightarrow |\mathfrak{M}|$ is an isomorphism. Then $g(n) = g(\text{Val}^{\mathfrak{N}}(\bar{n})) = \text{Val}^{\mathfrak{M}}(\bar{n})$. But for every $x \in |\mathfrak{M}|$, there is an $n \in \mathbb{N}$ such that $g(n) = x$, since g is **sobrejetiva**. $\square$

If a structure $\mathfrak{M}$ for $\mathcal{L}_A$ is standard, the elements of its **domínio** can all be named by the standard numerals $\bar{0}, \bar{1}, \bar{2}, \dots$, O.s., the terms o, o', o'' , etc. Of course, this does not mean that the **membros** of $|\mathfrak{M}|$ are the numbers, just that we can pick them out the same way we can pick out the numbers in $|\mathfrak{N}|$.

explanation

Problema 26.1. Show that the converse of [Proposição 26.2](#) is false, O.s., give an example of a [estrutura](#) $\mathfrak{M}$ with $|\mathfrak{M}| = \{\text{Val}^{\mathfrak{M}}(\bar{n}) : n \in \mathbb{N}\}$ that is not isomorphic to $\mathfrak{N}$.

Proposição 26.3. *If $\mathfrak{M} \models \mathbf{Q}$, and $|\mathfrak{M}| = \{\text{Val}^{\mathfrak{M}}(\bar{n}) : n \in \mathbb{N}\}$, then $\mathfrak{M}$ is [mod:mar:stm:](#) [prop:thq-standard](#) standard.*

Prova. We have to show that $\mathfrak{M}$ is isomorphic to $\mathfrak{N}$. Consider the function $g: \mathbb{N} \rightarrow |\mathfrak{M}|$ defined by $g(n) = \text{Val}^{\mathfrak{M}}(\bar{n})$. By the hypothesis, g is [sobrejetiva](#). It is also [injetiva](#): $\mathbf{Q} \vdash \bar{n} \neq \bar{m}$ whenever $n \neq m$. Thus, since $\mathfrak{M} \models \mathbf{Q}$, $\mathfrak{M} \models \bar{n} \neq \bar{m}$, whenever $n \neq m$. Thus, if $n \neq m$, then $\text{Val}^{\mathfrak{M}}(\bar{n}) \neq \text{Val}^{\mathfrak{M}}(\bar{m})$, O.s., $g(n) \neq g(m)$.

We also have to verify that g is an isomorphism.

1. We have $g(o^{\mathfrak{N}}) = g(0)$ since, $o^{\mathfrak{N}} = 0$. By definition of g , $g(0) = \text{Val}^{\mathfrak{M}}(\bar{0})$. But $\bar{0}$ is just o , and the value of a term which happens to be a [símbolo de constante](#) is given by what the [estrutura](#) assigns to that [símbolo de constante](#), O.s., $\text{Val}^{\mathfrak{M}}(o) = o^{\mathfrak{M}}$. So we have $g(o^{\mathfrak{N}}) = o^{\mathfrak{M}}$ as required.
2. $g(r^{\mathfrak{N}}(n)) = g(n+1)$, since r in $\mathfrak{N}$ is the successor function on $\mathbb{N}$. Then, $g(n+1) = \text{Val}^{\mathfrak{M}}(\overline{n+1})$ by definition of g . But $\overline{n+1}$ is the same term as $\bar{n}'$, so $\text{Val}^{\mathfrak{M}}(\overline{n+1}) = \text{Val}^{\mathfrak{M}}(\bar{n}')$. By the definition of the value function, this is $= r^{\mathfrak{M}}(\text{Val}^{\mathfrak{M}}(\bar{n}))$. Since $\text{Val}^{\mathfrak{M}}(\bar{n}) = g(n)$ we get $g(r^{\mathfrak{N}}(n)) = r^{\mathfrak{M}}(g(n))$.
3. $g(+^{\mathfrak{N}}(n, m)) = g(n+m)$, since $+$ in $\mathfrak{N}$ is the addition function on $\mathbb{N}$. Then, $g(n+m) = \text{Val}^{\mathfrak{M}}(\overline{n+m})$ by definition of g . But $\mathbf{Q} \vdash \overline{n+m} = (\bar{n} + \bar{m})$, so $\text{Val}^{\mathfrak{M}}(\overline{n+m}) = \text{Val}^{\mathfrak{M}}(\bar{n} + \bar{m})$. By the definition of the value function, this is $= +^{\mathfrak{M}}(\text{Val}^{\mathfrak{M}}(\bar{n}), \text{Val}^{\mathfrak{M}}(\bar{m}))$. Since $\text{Val}^{\mathfrak{M}}(\bar{n}) = g(n)$ and $\text{Val}^{\mathfrak{M}}(\bar{m}) = g(m)$, we get $g(+^{\mathfrak{N}}(n, m)) = +^{\mathfrak{M}}(g(n), g(m))$.
4. $g(\times^{\mathfrak{N}}(n, m)) = \times^{\mathfrak{M}}(g(n), g(m))$: Exercise.
5. $\langle n, m \rangle \in <^{\mathfrak{N}}$ iff $n < m$. If $n < m$, then $\mathbf{Q} \vdash \bar{n} < \bar{m}$, and also $\mathfrak{M} \models \bar{n} < \bar{m}$. Thus $\langle \text{Val}^{\mathfrak{M}}(\bar{n}), \text{Val}^{\mathfrak{M}}(\bar{m}) \rangle \in <^{\mathfrak{M}}$, O.s., $\langle g(n), g(m) \rangle \in <^{\mathfrak{M}}$. If $n \not< m$, then $\mathbf{Q} \vdash \neg \bar{n} < \bar{m}$, and consequently $\mathfrak{M} \not\models \bar{n} < \bar{m}$. Thus, as before, $\langle g(n), g(m) \rangle \notin <^{\mathfrak{M}}$. Together, we get: $\langle n, m \rangle \in <^{\mathfrak{N}}$ iff $\langle g(n), g(m) \rangle \in <^{\mathfrak{M}}$. $\square$

[explanation](#)

The function g is the most obvious way of defining a mapping from $\mathbb{N}$ to the domain of any other [estrutura](#) $\mathfrak{M}$ for $\mathcal{L}_A$, since every such $\mathfrak{M}$ contains [membros](#) named by $\bar{0}, \bar{1}, \bar{2}$, etc. So it isn't surprising that if $\mathfrak{M}$ makes at least some basic statements about the $\bar{n}$'s true in the same way that $\mathfrak{N}$ does, and g is also bijective, then g will turn into an isomorphism. In fact, if $|\mathfrak{M}|$ contains no [membros](#) other than what the $\bar{n}$'s name, it's the only one.

Proposição 26.4. *If $\mathfrak{M}$ is standard, then g from the proof of [Proposição 26.3](#) [mod:mar:stm:](#) [prop:thq-unique-iso](#) is the only isomorphism from $\mathfrak{N}$ to $\mathfrak{M}$.*

26.3. NON-STANDARD MODELS

Prova. Suppose $h: \mathbb{N} \rightarrow |\mathfrak{M}|$ is an isomorphism between $\mathfrak{N}$ and $\mathfrak{M}$. We show that $g = h$ by induction on n . If $n = 0$, then $g(0) = o^{\mathfrak{M}}$ by definition of g . But since h is an isomorphism, $h(0) = h(o^{\mathfrak{N}}) = o^{\mathfrak{M}}$, so $g(0) = h(0)$.

Now consider the case for $n + 1$. We have

$$\begin{aligned}
 g(n+1) &= \text{Val}^{\mathfrak{M}}(\overline{n+1}) \text{ by definition of } g \\
 &= \text{Val}^{\mathfrak{M}}(\overline{n'}) \text{ since } \overline{n+1} \equiv \overline{n'} \\
 &= \iota^{\mathfrak{M}}(\text{Val}^{\mathfrak{M}}(\overline{n})) \text{ by definition of } \text{Val}^{\mathfrak{M}}(t') \\
 &= \iota^{\mathfrak{M}}(g(n)) \text{ by definition of } g \\
 &= \iota^{\mathfrak{M}}(h(n)) \text{ by induction hypothesis} \\
 &= h(\iota^{\mathfrak{N}}(n)) \text{ since } h \text{ is an isomorphism} \\
 &= h(n+1)
 \end{aligned}$$

□

For any **infinito enumerável** set M , there's **uma bijeção** between $\mathbb{N}$ and M , explanation so every such set M is potentially the **domínio** of a standard model $\mathfrak{M}$. In fact, once you pick an object $z \in M$ and a suitable function s as $o^{\mathfrak{M}}$ and $\iota^{\mathfrak{M}}$, the interpretations of $+$, $\times$, and $<$ is already fixed. Only functions $s: M \rightarrow M \setminus \{z\}$ that are both **injetiva** and **sobrejetiva** are suitable in a standard model as $\iota^{\mathfrak{M}}$. The range of s cannot contain z , since otherwise $\forall x \, o \neq x'$ would be false. That **sentença** is true in $\mathfrak{N}$, and so $\mathfrak{M}$ also has to make it true. The function s has to be **injetiva**, since the successor function $\iota^{\mathfrak{N}}$ in $\mathfrak{N}$ is, and that $\iota^{\mathfrak{N}}$ is **injetiva** is expressed by a **sentença** true in $\mathfrak{N}$. It has to be **sobrejetiva** because otherwise there would be some $x \in M \setminus \{z\}$ not in the domain of s , O.s., the **sentença** $\forall x (x = o \vee \exists y \, y' = x)$ would be false in $\mathfrak{M}$ —but it is true in $\mathfrak{N}$.

-NoValue-

[content/model-theory/models-of-arithmetic/non-standard-models.tex](#)

26.3 Non-Standard Models

We call a **estrutura** for $\mathcal{L}_A$ standard if it is isomorphic to $\mathfrak{N}$. If a **estrutura** explanation isn't isomorphic to $\mathfrak{N}$, it is called non-standard.

Definição 26.5. A **estrutura** $\mathfrak{M}$ for $\mathcal{L}_A$ is *non-standard* if it is not isomorphic to $\mathfrak{N}$. The **membros** $x \in |\mathfrak{M}|$ which are equal to $\text{Val}^{\mathfrak{M}}(\overline{n})$ for some $n \in \mathbb{N}$ are called *standard numbers* (of $\mathfrak{M}$), and those not, *non-standard numbers*.

By **Proposição 26.2**, any standard **estrutura** for $\mathcal{L}_A$ contains only standard **membros**. explanation Consequently, a non-standard **estrutura** must contain at least one non-standard element. In fact, the existence of a non-standard **membro** guarantees that the **estrutura** is non-standard.

Proposição 26.6. *If a **estrutura** $\mathfrak{M}$ for $\mathcal{L}_A$ contains a non-standard number, $\mathfrak{M}$ is non-standard.*

Prova. Suppose not, O.s., suppose $\mathfrak{M}$ standard but contains a non-standard number x . Let $g: \mathbb{N} \rightarrow |\mathfrak{M}|$ be an isomorphism. It is easy to see (by induction on n) that $g(\text{Val}^{\mathfrak{M}}(\bar{n})) = \text{Val}^{\mathfrak{M}}(\bar{n})$. In other words, g maps standard numbers of $\mathfrak{N}$ to standard numbers of $\mathfrak{M}$. If $\mathfrak{M}$ contains a non-standard number, g cannot be *sobrejetiva*, contrary to hypothesis. $\square$

Problema 26.2. Recall that **Q** contains the axioms

$$\forall x \forall y (x' = y' \rightarrow x = y) \quad (Q_1)$$

$$\forall x \ 0 \neq x' \quad (Q_2)$$

$$\forall x (x = 0 \vee \exists y x = y') \quad (Q_3)$$

Give *estruturas* $\mathfrak{M}_1, \mathfrak{M}_2, \mathfrak{M}_3$ such that

1. $\mathfrak{M}_1 \models Q_1, \mathfrak{M}_1 \models Q_2, \mathfrak{M}_1 \not\models Q_3$;
2. $\mathfrak{M}_2 \models Q_1, \mathfrak{M}_2 \not\models Q_2, \mathfrak{M}_2 \models Q_3$; and
3. $\mathfrak{M}_3 \not\models Q_1, \mathfrak{M}_3 \models Q_2, \mathfrak{M}_3 \models Q_3$;

Obviously, you just have to specify $0^{\mathfrak{M}_i}$ and $1^{\mathfrak{M}_i}$ for each.

explanation

It is easy enough to specify non-standard *estruturas* for $\mathcal{L}_A$. For instance, take the structure with *domínio* $\mathbb{Z}$ and interpret all non-logical symbols as usual. Since negative numbers are not values of $\bar{n}$ for any n , this structure is non-standard. Of course, it will not be a *model* of arithmetic in the sense that it makes the same sentences true as $\mathfrak{N}$. For instance, $\forall x x' \neq 0$ is false. However, we can prove that non-standard models of arithmetic exist easily enough, using the compactness theorem.

Proposição 26.7. Let $\mathbf{TA} = \{\varphi : \mathfrak{N} \models \varphi\}$ be the theory of $\mathfrak{N}$. **TA** has a *enumerável* non-standard model.

Prova. Expand $\mathcal{L}_A$ by a new *símbolo de constante* c and consider the set of *sentenças*

$$\Gamma = \mathbf{TA} \cup \{c \neq \bar{0}, c \neq \bar{1}, c \neq \bar{2}, \dots\}$$

Any model $\mathfrak{M}^c$ of Γ would contain *um membro* $x = c^{\mathfrak{M}}$ which is non-standard, since $x \neq \text{Val}^{\mathfrak{M}}(\bar{n})$ for all $n \in \mathbb{N}$. Also, obviously, $\mathfrak{M}^c \models \mathbf{TA}$, since $\mathbf{TA} \subseteq \Gamma$. If we turn $\mathfrak{M}^c$ into a *estrutura* $\mathfrak{M}$ for $\mathcal{L}_A$ simply by forgetting about c , its domain still contains the non-standard x , and also $\mathfrak{M} \models \mathbf{TA}$. The latter is guaranteed since c does not occur in $\mathbf{TA}$. So, it suffices to show that Γ has a model.

We use the compactness theorem to show that Γ has a model. If every finite subset of Γ is satisfiable, so is Γ . Consider any finite subset $\Gamma_0 \subseteq \Gamma$. Γ_0 includes some *sentenças* of $\mathbf{TA}$ and some of the form $c \neq \bar{n}$, but only finitely many. Suppose k is the largest number so that $c \neq \bar{k} \in \Gamma_0$. Define $\mathfrak{N}_k$ by expanding $\mathfrak{N}$ to include the interpretation $c^{\mathfrak{N}_k} = k + 1$. $\mathfrak{N}_k \models \Gamma_0$: if $\varphi \in \mathbf{TA}$, $\mathfrak{N}_k \models \varphi$ since $\mathfrak{N}_k$ is just like $\mathfrak{N}$ in all respects except c , and c does not occur in φ . And $\mathfrak{N}_k \models c \neq \bar{n}$, since $n \leq k$, and $\text{Val}^{\mathfrak{N}_k}(c) = k + 1$. Thus, every finite subset of Γ is satisfiable. $\square$

26.4. MODELS OF $\mathbf{Q}$

-NoValue-

[content/model-theory/models-of-arithmetic/models-of-q.tex](#)

26.4 Models of $\mathbf{Q}$

We know that there are non-standard [estruturas](#) that make the same [sentenças](#) [explanation](#) true as $\mathfrak{N}$ does, O.s., is a model of $\mathbf{TA}$. Since $\mathfrak{N} \models \mathbf{Q}$, any model of $\mathbf{TA}$ is also a model of $\mathbf{Q}$. $\mathbf{Q}$ is much weaker than $\mathbf{TA}$, p.ex., $\mathbf{Q} \not\models \forall x \forall y (x + y) = (y + x)$. Weaker theories are easier to satisfy: they have more models. P.ex., $\mathbf{Q}$ has models which make $\forall x \forall y (x + y) = (y + x)$ false, but those cannot also be models of $\mathbf{TA}$, or $\mathbf{PA}$ for that matter. Models of $\mathbf{Q}$ are also relatively simple: we can specify them explicitly.

[mod:mar:mdq:](#)
[ex:model-K-of-Q](#)

Exemplo 26.8. Consider the [estrutura](#) $\mathfrak{K}$ with domain $|\mathfrak{K}| = \mathbb{N} \cup \{a\}$ and interpretations

$$\begin{aligned} 0^{\mathfrak{K}} &= 0 \\ \iota^{\mathfrak{K}}(x) &= \begin{cases} x + 1 & \text{if } x \in \mathbb{N} \\ a & \text{if } x = a \end{cases} \\ +^{\mathfrak{K}}(x, y) &= \begin{cases} x + y & \text{if } x, y \in \mathbb{N} \\ a & \text{otherwise} \end{cases} \\ \times^{\mathfrak{K}}(x, y) &= \begin{cases} xy & \text{if } x, y \in \mathbb{N} \\ 0 & \text{if } x = 0 \text{ or } y = 0 \\ a & \text{otherwise} \end{cases} \\ <^{\mathfrak{K}} &= \{\langle x, y \rangle : x, y \in \mathbb{N} \text{ and } x < y\} \cup \{\langle x, a \rangle : x \in |\mathfrak{K}|\} \end{aligned}$$

To show that $\mathfrak{K} \models \mathbf{Q}$ we have to verify that all axioms of $\mathbf{Q}$ are true in $\mathfrak{K}$. For convenience, let's write x^* for $\iota^{\mathfrak{K}}(x)$ (the “successor” of x in $\mathfrak{K}$), $x \oplus y$ for $+^{\mathfrak{K}}(x, y)$ (the “sum” of x and y in $\mathfrak{K}$), $x \otimes y$ for $\times^{\mathfrak{K}}(x, y)$ (the “product” of x and y in $\mathfrak{K}$), and $x \odot y$ for $\langle x, y \rangle \in <^{\mathfrak{K}}$. With these abbreviations, we can give the operations in $\mathfrak{K}$ more perspicuously as

x	x^*	$x \oplus y$	0	m	a	$x \otimes y$	0	m	a
n	$n + 1$	0	0	m	a	0	0	0	0
a	a	n	n	$n + m$	a	n	0	nm	a
		a	a	a	a	a	0	a	a

We have $n \odot m$ iff $n < m$ for $n, m \in \mathbb{N}$ and $x \odot a$ for all $x \in |\mathfrak{K}|$.

$\mathfrak{K} \models \forall x \forall y (x' = y' \rightarrow x = y)$ since $*$ is [injetiva](#). $\mathfrak{K} \models \forall x 0 \neq x'$ since 0 is not a $*$ -successor in $\mathfrak{K}$. $\mathfrak{K} \models \forall x (x = 0 \vee \exists y x = y')$ since for every $n > 0$, $n = (n - 1)^*$, and $a = a^*$.

$\mathfrak{K} \models \forall x (x + 0) = x$ since $n \oplus 0 = n + 0 = n$, and $a \oplus 0 = a$ by definition of $\oplus$. $\mathfrak{K} \models \forall x \forall y (x + y') = (x + y)'$ is a bit trickier. If n, m are both standard,

we have:

$$(n \oplus m^*) = (n + (m + 1)) = (n + m) + 1 = (n \oplus m)^*$$

since $\oplus$ and $*$ agree with $+$ and $\prime$ on standard numbers. Now suppose $x \in |\mathfrak{K}|$. Then

$$(x \oplus a^*) = (x \oplus a) = a = a^* = (x \oplus a)^*$$

The remaining case is if $y \in |\mathfrak{K}|$ but $x = a$. Here we also have to distinguish cases according to whether $y = n$ is standard or $y = b$:

$$\begin{aligned} (a \oplus n^*) &= (a \oplus (n + 1)) = a = a^* = (a \oplus n)^* \\ (a \oplus a^*) &= (a \oplus a) = a = a^* = (a \oplus a)^* \end{aligned}$$

This is of course a bit more detailed than needed. For instance, since $a \oplus z = a$ whatever z is, we can immediately conclude $a \oplus a^* = a$. The remaining axioms can be verified the same way.

$\mathfrak{K}$ is thus a model of $\mathbf{Q}$. Its “addition” $\oplus$ is also commutative. But there are other sentences true in $\mathfrak{N}$ but false in $\mathfrak{K}$, and vice versa. For instance, $a \otimes a$, so $\mathfrak{K} \models \exists x x < x$ and $\mathfrak{K} \not\models \forall x \neg x < x$. This shows that $\mathbf{Q} \not\models \forall x \neg x < x$.

Problema 26.3. Prove that $\mathfrak{K}$ from [Exemplo 26.8](#) satisfies the remaining axioms of $\mathbf{Q}$,

$$\forall x (x \times 0) = 0 \quad (Q_6)$$

$$\forall x \forall y (x \times y') = ((x \times y) + x) \quad (Q_7)$$

$$\forall x \forall y (x < y \leftrightarrow \exists z (z' + x) = y) \quad (Q_8)$$

Find a [sentença](#) only involving $\prime$ true in $\mathfrak{N}$ but false in $\mathfrak{K}$.

Exemplo 26.9. Consider the [estrutura](#) $\mathfrak{L}$ with domain $|\mathfrak{L}| = \mathbb{N} \cup \{a, b\}$ and interpretations $\prime^{\mathfrak{L}} = *$, $+^{\mathfrak{L}} = \oplus$ given by [mod:mar:mdq:ex:model-L-of-Q](#)

x	x^*	$x \oplus y$	m	a	b
n	$n + 1$	n	$n + m$	b	a
a	a	a	a	b	a
b	b	b	b	b	a

Since $*$ is [injetiva](#), 0 is not in its range, and every $x \in |\mathfrak{L}|$ other than 0 is, axioms Q_1 – Q_3 are true in $\mathfrak{L}$. For any x , $x \oplus 0 = x$, so Q_4 is true as well. For Q_5 , consider $x \oplus y^*$ and $(x \oplus y)^*$. They are equal if x and y are both standard, since then $*$ and $\oplus$ agree with $\prime$ and $+$. If x is non-standard, and y is standard, we have $x \oplus y^* = x = x^* = (x \oplus y)^*$. If x and y are both non-standard, we have four cases:

$$\begin{aligned} a \oplus a^* &= b = b^* = (a \oplus a)^* \\ b \oplus b^* &= a = a^* = (b \oplus b)^* \\ b \oplus a^* &= b = b^* = (b \oplus y)^* \\ a \oplus b^* &= a = a^* = (a \oplus b)^* \end{aligned}$$

26.5. MODELS OF PA

If x is standard, but y is non-standard, we have

$$\begin{aligned} n \oplus a^* &= n \oplus a = b = b^* = (n \oplus a)^* \\ n \oplus b^* &= n \oplus b = a = a^* = (n \oplus b)^* \end{aligned}$$

So, $\mathfrak{L} \models Q_5$. However, $a \oplus 0 \neq 0 \oplus a$, so $\mathfrak{L} \not\models \forall x \forall y (x + y) = (y + x)$.

Problema 26.4. Expand $\mathfrak{L}$ of [Exemplo 26.9](#) to include $\otimes$ and $\oslash$ that interpret $\times$ and $<$. Show that your structure satisfies the remaining axioms of $\mathbf{Q}$,

$$\forall x (x \times 0) = 0 \quad (Q_6)$$

$$\forall x \forall y (x \times y') = ((x \times y) + x) \quad (Q_7)$$

$$\forall x \forall y (x < y \leftrightarrow \exists z (z' + x) = y) \quad (Q_8)$$

Problema 26.5. In $\mathfrak{L}$ of [Exemplo 26.9](#), $a^* = a$ and $b^* = b$. Is there a model of $\mathbf{Q}$ in which $a^* = b$ and $b^* = a$?

We've explicitly constructed models of $\mathbf{Q}$ in which the non-standard [membros](#) live “beyond” the standard elements. In fact, that much is required by the axioms. A non-standard [membro](#) x cannot be $\oslash 0$, since $\mathbf{Q} \vdash \forall x \neg x < 0$ (see [Lema 35.23](#)). Also, for every n , $\mathbf{Q} \vdash \forall x (x < \bar{n}' \rightarrow (x = \bar{0} \vee x = \bar{1} \vee \dots \vee x = \bar{n}))$ ([Lema 35.24](#)), so we can't have $a \oslash n$ for any $n > 0$.

-NoValue-

[content/model-theory/models-of-arithmetic/models-of-pa.tex](#)

26.5 Models of PA

Any non-standard model of $\mathbf{TA}$ is also one of $\mathbf{PA}$. We know that non-standard models of $\mathbf{TA}$ and hence of $\mathbf{PA}$ exist. We also know that such non-standard models contain non-standard “numbers,” O.s., [membros](#) of the domain that are “beyond” all the standard “numbers.” But how are they arranged? How many are there? We've seen that models of the weaker theory $\mathbf{Q}$ can contain as few as a single non-standard number. But these simple [estruturas](#) are not models of $\mathbf{PA}$ or $\mathbf{TA}$.

The key to understanding the structure of models of $\mathbf{PA}$ or $\mathbf{TA}$ is to see what facts are [deriváveis](#) in these theories. For instance, already $\mathbf{PA}$ proves that $\forall x x \neq x'$ and $\forall x \forall y (x + y) = (y + x)$, so this rules out simple structures (in which these [sentenças](#) are false) as models of $\mathbf{PA}$.

Suppose $\mathfrak{M}$ is a model of $\mathbf{PA}$. Then if $\mathbf{PA} \vdash \varphi$, $\mathfrak{M} \models \varphi$. Let's again use $\mathbf{z}$ for $0^{\mathfrak{M}}$, $*$ for $\neq^{\mathfrak{M}}$, $\oplus$ for $+^{\mathfrak{M}}$, $\otimes$ for $\times^{\mathfrak{M}}$, and $\oslash$ for $<^{\mathfrak{M}}$. Any [sentença](#) φ then states some condition about $\mathbf{z}$, $*$, $\oplus$, $\otimes$, and $\oslash$, and if $\mathfrak{M} \models \varphi$ that condition must be satisfied. For instance, if $\mathfrak{M} \models Q_1$, O.s., $\mathfrak{M} \models \forall x \forall y (x' = y' \rightarrow x = y)$, then $*$ must be [injetiva](#).

Proposição 26.10. In $\mathfrak{M}$, $\oslash$ is a linear strict order, O.s., it satisfies:

1. Not $x \oslash x$ for any $x \in |\mathfrak{M}|$.

2. If $x \otimes y$ and $y \otimes z$ then $x \otimes z$.
3. For any $x \neq y$, $x \otimes y$ or $y \otimes x$

Prova. **PA** proves:

1. $\forall x \neg x < x$
2. $\forall x \forall y \forall z ((x < y \wedge y < z) \rightarrow x < z)$
3. $\forall x \forall y ((x < y \vee y < x) \vee x = y)$ □

Proposição 26.11. $\mathbf{z}$ is the least *membro* of $|\mathfrak{M}|$ in the $\otimes$ -ordering. For any x , $x \otimes x^*$, and x^* is the $\otimes$ -least *membro* with that property. For any x , there is a unique y such that $y^* = x$. (We call y the “predecessor” of x in $\mathfrak{M}$, and denote it by $^*x$.) mod:mar:mpa:
prop:M-discrete

Prova. Exercise. □

Problema 26.6. Find *sentenças* in $\mathcal{L}_A$ *derivável* in **PA** (and hence true in $\mathfrak{N}$) which guarantee the properties of $\mathbf{z}$, $*$, and $\otimes$ in **Proposição 26.11**

Proposição 26.12. All standard *membros* of $\mathfrak{M}$ are less than (according to $\otimes$) all non-standard *membros*.

Prova. We'll use n as short for $\text{Val}^{\mathfrak{M}}(\bar{n})$, a standard *membro* of $\mathfrak{M}$. Already **Q** proves that, for any $n \in \mathbb{N}$, $\forall x (x < \bar{n}' \rightarrow (x = \bar{0} \vee x = \bar{1} \vee \dots \vee x = \bar{n}))$. There are no *membros* that are $\otimes \mathbf{z}$. So if n is standard and x is non-standard, we cannot have $x \otimes n$. By definition, a non-standard element is one that isn't $\text{Val}^{\mathfrak{M}}(\bar{n})$ for any $n \in \mathbb{N}$, so $x \neq n$ as well. Since $\otimes$ is a linear order, we must have $n \otimes x$. □

Proposição 26.13. Every nonstandard *membro* x of $|\mathfrak{M}|$ is an element of the subset

$$\dots *** x \otimes ** x \otimes * x \otimes x \otimes x^* \otimes x^{**} \otimes x^{***} \otimes \dots$$

We call this subset the block of x and write it as $[x]$. It has no least and no greatest *membro*. It can be characterized as the set of those $y \in |\mathfrak{M}|$ such that, for some standard n , $x \oplus n = y$ or $y \oplus n = x$.

Prova. Clearly, such a set $[x]$ always exists since every *membro* y of $|\mathfrak{M}|$ has a unique successor y^* and unique predecessor $^*y$. For successive *membros* y , y^* we have $y \otimes y^*$ and y^* is the $\otimes$ -least *membro* of $|\mathfrak{M}|$ such that y is $\otimes$ -less than it. Since always $^*y \otimes y$ and $y \otimes y^*$, $[x]$ has no least or greatest *membro*. If $y \in [x]$ then $x \in [y]$, for then either $y^{***} = x$ or $x^{***} = y$. If $y^{***} = x$ (with n $*$'s), then $y \oplus n = x$ and conversely, since **PA** $\vdash \forall x x' \dots' = (x + \bar{n})$ (if n is the number of $'$'s). □

26.5. MODELS OF PA

Proposição 26.14. *If $[x] \neq [y]$ and $x \otimes y$, then for any $u \in [x]$ and any $v \in [y]$, $u \otimes v$.*

Prova. Note that $\mathbf{PA} \vdash \forall x \forall y (x < y \rightarrow (x' < y \vee x' = y))$. Thus, if $u \otimes v$, we also have $u \oplus n^* \otimes v$ for any n if $[u] \neq [v]$.

Any $u \in [x]$ is $\otimes y$: $x \otimes y$ by assumption. If $u \otimes x$, $u \otimes y$ by transitivity. And if $x \otimes u$ but $u \in [x]$, we have $u = x \oplus n^*$ for some n , and so $u \otimes y$ by the fact just proved.

Now suppose that $v \in [y]$ is $\otimes y$, O.s., $v \oplus m^* = y$ for some standard m . This rules out $v \otimes x$, otherwise $y = v \oplus m^* \otimes x$. Clearly also, $x \neq v$, otherwise $x \oplus m^* = v \oplus m^* = y$ and we would have $[x] = [y]$. So, $x \otimes v$. But then also $x \oplus n^* \otimes v$ for any n . Hence, if $x \otimes u$ and $u \in [x]$, we have $u \otimes v$. If $u \otimes x$ then $u \otimes v$ by transitivity.

Lastly, if $y \otimes v$, $u \otimes v$ since, as we've shown, $u \otimes y$ and $y \otimes v$. $\square$

Corolário 26.15. *If $[x] \neq [y]$, $[x] \cap [y] = \emptyset$.*

Prova. Suppose $z \in [x]$ and $x \otimes y$. Then $z \otimes u$ for all $u \in [y]$. If $z \in [y]$, we would have $z \otimes z$. Similarly if $y \otimes x$. $\square$

This means that the blocks themselves can be ordered in a way that respects $\otimes$: $[x] \otimes [y]$ iff $x \otimes y$, or, equivalently, if $u \otimes v$ for any $u \in [x]$ and $v \in [y]$. Clearly, the standard block $[0]$ is the least block. It intersects with no non-standard block, and no two non-standard blocks intersect either. Specifically, you cannot “reach” a different block by taking repeated successors or predecessors. explanation

Proposição 26.16. *If x and y are non-standard, then $x \otimes x \oplus y$ and $x \oplus y \notin [x]$.*

Prova. If y is nonstandard, then $y \neq \mathbf{z}$. $\mathbf{PA} \vdash \forall x (y \neq 0 \rightarrow x < (x + y))$. Now suppose $x \oplus y \in [x]$. Since $x \otimes x \oplus y$, we would have $x \oplus n^* = x \oplus y$. But $\mathbf{PA} \vdash \forall x \forall y \forall z ((x + y) = (x + z) \rightarrow y = z)$ (the cancellation law for addition). This would mean $y = n^*$ for some standard n ; but y is assumed to be non-standard. $\square$

Proposição 26.17. *There is no least non-standard block.*

Prova. $\mathbf{PA} \vdash \forall x \exists y ((y + y) = x \vee (y + y)' = x)$, O.s., that every x is divisible by 2 (possibly with remainder 1). If x is non-standard, so is y . By the preceding proposition, $y \otimes y \oplus y$ and $y \oplus y \notin [y]$. Then also $y \otimes (y \oplus y)^*$ and $(y \oplus y)^* \notin [y]$. But $x = y \oplus y$ or $x = (y \oplus y)^*$, so $y \otimes x$ and $y \notin [x]$. $\square$

Proposição 26.18. *There is no largest block.*

Prova. Exercise. $\square$

Problema 26.7. Show that in a non-standard model of $\mathbf{PA}$, there is no largest block.

Proposição 26.19. *The ordering of the blocks is dense. That is, if $x \otimes y$ and $[x] \neq [y]$, then there is a block $[z]$ distinct from both that is between them.* *mod:mar:mpa:
prop:blocks-dense*

Prova. Suppose $x \otimes y$. As before, $x \oplus y$ is divisible by two (possibly with remainder): there is a $z \in |\mathfrak{M}|$ such that either $x \oplus y = z \oplus z$ or $x \oplus y = (z \oplus z)^*$. The element z is the “average” of x and y , and $x \otimes z$ and $z \otimes y$. $\square$

Problema 26.8. Write out a detailed proof of **Proposição 26.19**. Which *sentença* must **PA** *deriva* in order to guarantee the existence of z ? Why is $x \otimes z$ and $z \otimes y$, and why is $[x] \neq [z]$ and $[z] \neq [y]$?

explanation

The non-standard blocks are therefore ordered like the rationals: they form a *infinito enumerável* dense linear ordering without endpoints. One can show that any two such *infinito enumerável* orderings are isomorphic. It follows that for any two *enumerável* non-standard models $\mathfrak{M}_1$ and $\mathfrak{M}_2$ of true arithmetic, their reducts to the language containing $<$ and $=$ only are isomorphic. Indeed, an isomorphism h can be defined as follows: the standard parts of $\mathfrak{M}_1$ and $\mathfrak{M}_2$ are isomorphic to the standard model $\mathfrak{N}$ and hence to each other. The blocks making up the non-standard part are themselves ordered like the rationals and therefore isomorphic; an isomorphism of the blocks can be extended to an isomorphism *within* the blocks by matching up arbitrary elements in each, and then taking the image of the successor of x in $\mathfrak{M}_1$ to be the successor of the image of x in $\mathfrak{M}_2$. Note that it does *not* follow that $\mathfrak{M}_1$ and $\mathfrak{M}_2$ are isomorphic in the full language of arithmetic (indeed, isomorphism is always relative to a *linguagem*), as there are non-isomorphic ways to define addition and multiplication over $|\mathfrak{M}_1|$ and $|\mathfrak{M}_2|$. (This also follows from a famous theorem due to Vaught that the number of countable models of a complete theory cannot be 2.)

-NoValue-

<content/model-theory/models-of-arithmetic/computable-models.tex>

26.6 Computable Models of Arithmetic

explanation

The standard model $\mathfrak{N}$ has two nice features. Its domain is the natural numbers $\mathbb{N}$, O.s., its elements are just the kinds of things we want to talk about using the language of arithmetic, and the standard numeral $\bar{n}$ actually picks out n . The other nice feature is that the interpretations of the non-logical symbols of $\mathcal{L}_A$ are all *computable*. The successor, addition, and multiplication functions which serve as $\iota^{\mathfrak{N}}$, $+\mathfrak{N}$, and $\times^{\mathfrak{N}}$ are computable functions of numbers. (Computable by Turing machines, or definable by primitive recursion, say.) And the less-than relation on $\mathfrak{N}$, O.s., $<^{\mathfrak{N}}$, is decidable.

Non-standard models of arithmetical theories such as **Q** and **PA** must contain non-standard elements. Thus their domains typically include *membros* in addition to $\mathbb{N}$. However, any countable *estrutura* can be built on any *infinito enumerável* set, including $\mathbb{N}$. So there are also non-standard models with domain $\mathbb{N}$. In such models $\mathfrak{M}$, of course, at least some numbers cannot play

26.6. COMPUTABLE MODELS OF ARITHMETIC

the roles they usually play, since some k must be different from $\text{Val}^{\mathfrak{M}}(\bar{n})$ for all $n \in \mathbb{N}$.

Definição 26.20. A estrutura $\mathfrak{M}$ for $\mathcal{L}_A$ is *computable* iff $|\mathfrak{M}| = \mathbb{N}$ and $\iota^{\mathfrak{M}}$, $+\mathfrak{M}$, $\times^{\mathfrak{M}}$ are computable functions and $<^{\mathfrak{M}}$ is a decidable relation.

Exemplo 26.21. Recall the structure $\mathfrak{K}$ from Exemplo 26.8. Its domain was $|\mathfrak{K}| = \mathbb{N} \cup \{a\}$ and interpretations

$$\begin{aligned} o^{\mathfrak{K}} &= 0 \\ \iota^{\mathfrak{K}}(x) &= \begin{cases} x+1 & \text{if } x \in \mathbb{N} \\ a & \text{if } x = a \end{cases} \\ +^{\mathfrak{K}}(x, y) &= \begin{cases} x+y & \text{if } x, y \in \mathbb{N} \\ a & \text{otherwise} \end{cases} \\ \times^{\mathfrak{K}}(x, y) &= \begin{cases} xy & \text{if } x, y \in \mathbb{N} \\ 0 & \text{if } x = 0 \text{ or } y = 0 \\ a & \text{otherwise} \end{cases} \\ <^{\mathfrak{K}} = \{ \langle x, y \rangle : x, y \in \mathbb{N} \text{ and } x < y \} \cup \{ \langle x, a \rangle : n \in |\mathfrak{K}| \} \end{aligned}$$

But $|\mathfrak{K}|$ is *infinito enumerável* and so is equinumerous with $\mathbb{N}$. For instance, $g: \mathbb{N} \rightarrow |\mathfrak{K}|$ with $g(0) = a$ and $g(n) = n+1$ for $n > 0$ is *uma bijeção*. We can turn it into an isomorphism between a new model $\mathfrak{K}'$ of $\mathbf{Q}$ and $\mathfrak{K}$. In $\mathfrak{K}'$, we have to assign different functions and relations to the symbols of $\mathcal{L}_A$, since different *membros* of $\mathbb{N}$ play the roles of standard and non-standard numbers.

Specifically, 0 now plays the role of a , not of the smallest standard number. The smallest standard number is now 1. So we assign $o^{\mathfrak{K}'} = 1$. The successor function is also different now: given a standard number, O.s., an $n > 0$, it still returns $n+1$. But 0 now plays the role of a , which is its own successor. So $\iota^{\mathfrak{K}'}(0) = 0$. For addition and multiplication we likewise have

$$\begin{aligned} +^{\mathfrak{K}'}(x, y) &= \begin{cases} x+y-1 & \text{if } x, y > 0 \\ 0 & \text{otherwise} \end{cases} \\ \times^{\mathfrak{K}'}(x, y) &= \begin{cases} 1 & \text{if } x = 1 \text{ or } y = 1 \\ xy - x - y + 2 & \text{if } x, y > 1 \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

And we have $\langle x, y \rangle \in <^{\mathfrak{K}'}$ iff $x < y$ and $x > 0$ and $y > 0$, or if $y = 0$.

All of these functions are computable functions of natural numbers and $<^{\mathfrak{K}'}$ is a decidable relation on $\mathbb{N}$ —but they are not the same functions as successor, addition, and multiplication on $\mathbb{N}$, and $<^{\mathfrak{K}'}$ is not the same relation as $<$ on $\mathbb{N}$.

Problema 26.9. Give a estrutura $\mathfrak{L}'$ with $|\mathfrak{L}'| = \mathbb{N}$ isomorphic to $\mathfrak{L}$ of Exemplo 26.9.

explanation

Exemplo 26.21 shows that $\mathbf{Q}$ has computable non-standard models with domain $\mathbb{N}$. However, the following result shows that this is not true for models of $\mathbf{PA}$ (and thus also for models of $\mathbf{TA}$).

Teorema 26.22 (Tennenbaum's Theorem). *$\mathfrak{N}$ is the only computable model of $\mathbf{PA}$.*

Capítulo 27

The Interpolation Theorem

-NoValue-

`content/model-theory/interpolation/introduction.tex`

27.1 Introduction

The interpolation theorem is the following result: Suppose $\models \varphi \rightarrow \psi$. Then there is a **sentença** χ such that $\models \varphi \rightarrow \chi$ and $\models \chi \rightarrow \psi$. Moreover, every **símbolo de constante**, **símbolo de função**, and **símbolo de predicado** (other than $=$) in χ occurs both in φ and ψ . The **sentença** χ is called an *interpolant* of φ and ψ .

mod:int:int:
sec

The interpolation theorem is interesting in its own right, but its main importance lies in the fact that it can be used to prove results about definability in a theory, and the conditions under which combining two consistent theories results in a consistent theory. The first result is known as the Beth definability theorem; the second, Robinson's joint consistency theorem.

-NoValue-

`content/model-theory/interpolation/separation.tex`

27.2 Separation of **Sentenças**

A bit of groundwork is needed before we can proceed with the proof of the interpolation theorem. An interpolant for φ and ψ is a **sentença** χ such that $\varphi \models \chi$ and $\chi \models \psi$. By contraposition, the latter is true iff $\neg\psi \models \neg\chi$. A **sentença** χ with this property is said to *separate* φ and $\neg\psi$. So finding an interpolant for φ and ψ amounts to finding a **sentença** that separates φ and $\neg\psi$. As so often,

mod:int:sep:
sec

27.2. SEPARATION OF SENTENÇAS

it will be useful to consider a generalization: a sentence that separates two *sets* of *sentenças*.

Definição 27.1. A sentence χ *separates* sets of sentences Γ and Δ if and only if $\Gamma \models \chi$ and $\Delta \models \neg\chi$. If no such *sentença* exists, then Γ and Δ are *inseparable*.

The inclusion relations between the classes of models of Γ , Δ and χ are represented below:

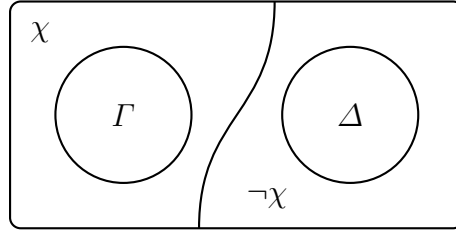


Figura 27.1: χ separates Γ and Δ

mod:int:sep:
fig:sep
mod:int:sep:
lem:sep1

Lema 27.2. Suppose $\mathcal{L}_0$ is the language containing every *símbolo de constante*, *símbolo de função* and *símbolo de predicado* (other than $\doteq$) that occurs in both Γ and Δ , and let $\mathcal{L}'_0$ be obtained by the addition of infinitely many new *símbolos de constante* c_n for $n \geq 0$. Then if Γ and Δ are inseparable in $\mathcal{L}_0$, they are also inseparable in $\mathcal{L}'_0$.

Prova. We proceed indirectly: suppose by way of contradiction that Γ and Δ are separated in $\mathcal{L}'_0$. Then $\Gamma \models \chi[c/x]$ and $\Delta \models \neg\chi[c/x]$ for some $\chi \in \mathcal{L}_0$ (where c is a new *símbolo de constante*—the case where χ contains more than one such new *símbolo de constante* is similar). By compactness, there are *finite* subsets Γ_0 of Γ and Δ_0 of Δ such that $\Gamma_0 \models \chi[c/x]$ and $\Delta_0 \models \neg\chi[c/x]$. Let γ be the conjunction of all *fórmulas* in Γ_0 and δ the conjunction of all *fórmulas* in Δ_0 . Then

$$\gamma \models \chi[c/x], \quad \delta \models \neg\chi[c/x].$$

From the former, by Generalization, we have $\gamma \models \forall x \chi$, and from the latter by contraposition, $\chi[c/x] \models \neg\delta$, whence also $\forall x \chi \models \neg\delta$. Contraposition again gives $\delta \models \neg\forall x \chi$. By monotonicity,

$$\Gamma \models \forall x \chi, \quad \Delta \models \neg\forall x \chi,$$

so that $\forall x \chi$ separates Γ and Δ in $\mathcal{L}_0$. □

mod:int:sep:
lem:sep2

Lema 27.3. Suppose that $\Gamma \cup \{\exists x \sigma\}$ and Δ are inseparable, and c is a new *símbolo de constante* not in Γ , Δ , or σ . Then $\Gamma \cup \{\exists x \sigma, \sigma[c/x]\}$ and Δ are also inseparable.

Prova. Suppose for contradiction that χ separates $\Gamma \cup \{\exists x \sigma, \sigma[c/x]\}$ and Δ , while at the same time $\Gamma \cup \{\exists x \sigma\}$ and Δ are inseparable. We distinguish two cases:

1. c does not occur in χ : in this case $\Gamma \cup \{\exists x \sigma, \neg \chi\}$ is satisfiable (otherwise χ separates $\Gamma \cup \{\exists x \sigma\}$ and Δ). It remains so if $\sigma[c/x]$ is added, so χ does not separate $\Gamma \cup \{\exists x \sigma, \sigma[c/x]\}$ and Δ after all.
2. c does occur in χ so that χ has the form $\chi[c/x]$. Then we have that

$$\Gamma \cup \{\exists x \sigma, \sigma[c/x]\} \models \chi[c/x],$$

whence $\Gamma, \exists x \sigma \models \forall x (\sigma \rightarrow \chi)$ by the Deduction Theorem and Generalization, and finally $\Gamma \cup \{\exists x \sigma\} \models \exists x \chi$. On the other hand, $\Delta \models \neg \chi[c/x]$ and hence by Generalization $\Delta \models \neg \exists x \chi$. So $\Gamma \cup \{\exists x \sigma\}$ and Δ are separable, a contradiction. $\square$

-NoValue-

content/model-theory/interpolation/interpolation-proof.tex

27.3 Craig's Interpolation Theorem

Teorema 27.4 (Craig's Interpolation Theorem). *If $\models \varphi \rightarrow \psi$, then there is a *sentença* χ such that $\models \varphi \rightarrow \chi$ and $\models \chi \rightarrow \psi$, and every *símbolo de constante*, *símbolo de função*, and *símbolo de predicado* (other than $=$) in χ occurs both in φ and ψ . The *sentença* χ is called an interpolant of φ and ψ .*

mod:int:prf:
sec
mod:int:prf:
thm:interpol

Prova. Suppose $\mathcal{L}_1$ is the language of φ and $\mathcal{L}_2$ is the language of ψ . Let $\mathcal{L}_0 = \mathcal{L}_1 \cap \mathcal{L}_2$. For each $i \in \{0, 1, 2\}$, let $\mathcal{L}'_i$ be obtained from $\mathcal{L}_i$ by adding the infinitely many new *símbolos de constante* $c_0, c_1, c_2, \dots$.

If φ is unsatisfiable, $\exists x x \neq x$ is an interpolant. If $\neg \psi$ is unsatisfiable (and hence ψ is valid), $\exists x x = x$ is an interpolant. So we may assume also that both φ and $\neg \psi$ are satisfiable.

In order to prove the contrapositive of the Interpolation Theorem, assume that there is no interpolant for φ and ψ . In other words, assume that $\{\varphi\}$ and $\{\neg \psi\}$ are inseparable in $\mathcal{L}_0$.

Our goal is to extend the pair $(\{\varphi\}, \{\neg \psi\})$ to a maximally inseparable pair (Γ^*, Δ^*) . Let $\varphi_0, \varphi_1, \varphi_2, \dots$ enumerate the *sentenças* of $\mathcal{L}_1$, and $\psi_0, \psi_1, \psi_2, \dots$ enumerate the *sentenças* of $\mathcal{L}_2$. We define two increasing sequences of sets of *sentenças* (Γ_n, Δ_n) , for $n \geq 0$, as follows. Put $\Gamma_0 = \{\varphi\}$ and $\Delta_0 = \{\neg \psi\}$. Assuming (Γ_n, Δ_n) are already defined, define Γ_{n+1} and Δ_{n+1} by:

1. If $\Gamma_n \cup \{\varphi_n\}$ and Δ_n are inseparable in $\mathcal{L}'_0$, put φ_n in Γ_{n+1} . Moreover, if φ_n is an existential *fórmula* $\exists x \sigma$ then pick a new *símbolo de constante* c not occurring in $\Gamma_n, \Delta_n, \varphi_n$ or ψ_n , and put $\sigma[c/x]$ in Γ_{n+1} .

27.3. CRAIG'S INTERPOLATION THEOREM

2. If Γ_{n+1} and $\Delta_n \cup \{\psi_n\}$ are inseparable in $\mathcal{L}'_0$, put ψ_n in Δ_{n+1} . Moreover, if ψ_n is an existential **fórmula** $\exists x \sigma$, then pick a new **símbolo de constante** c not occurring in Γ_{n+1} , Δ_n , φ_n or ψ_n , and put $\sigma[c/x]$ in Δ_{n+1} .

Finally, define:

$$\Gamma^* = \bigcup_{n \geq 0} \Gamma_n, \quad \Delta^* = \bigcup_{n \geq 0} \Delta_n.$$

By simultaneous induction on n we can now prove:

1. Γ_n and Δ_n are inseparable in $\mathcal{L}'_0$;
2. Γ_{n+1} and Δ_n are inseparable in $\mathcal{L}'_0$.

The basis for (1) is given by **Lema 27.2**. For part (2), we need to distinguish three cases:

1. If $\Gamma_0 \cup \{\varphi_0\}$ and Δ_0 are separable, then $\Gamma_1 = \Gamma_0$ and (2) is just (1);
2. If $\Gamma_1 = \Gamma_0 \cup \{\varphi_0\}$, then Γ_1 and Δ_0 are inseparable by construction.
3. It remains to consider the case where φ_0 is existential, so that $\Gamma_1 = \Gamma_0 \cup \{\exists x \sigma, \sigma[c/x]\}$. By construction, $\Gamma_0 \cup \{\exists x \sigma\}$ and Δ_0 are inseparable, so that by **Lema 27.3** also $\Gamma_0 \cup \{\exists x \sigma, \sigma[c/x]\}$ and Δ_0 are inseparable.

This completes the basis of the induction for (1) and (2) above. Now for the inductive step. For (1), if $\Delta_{n+1} = \Delta_n \cup \{\psi_n\}$ then Γ_{n+1} and Δ_{n+1} are inseparable by construction (even when ψ_n is existential, by **Lema 27.3**); if $\Delta_{n+1} = \Delta_n$ (because Γ_{n+1} and $\Delta_n \cup \{\psi_n\}$ are separable), then we use the induction hypothesis on (2). For the inductive step for (2), if $\Gamma_{n+2} = \Gamma_{n+1} \cup \{\varphi_{n+1}\}$ then Γ_{n+2} and Δ_{n+1} are inseparable by construction (even when φ_{n+1} is existential, by **Lema 27.3**); and if $\Gamma_{n+2} = \Gamma_{n+1}$ then we use the inductive case for (1) just proved. This concludes the induction on (1) and (2).

It follows that Γ^* and Δ^* are inseparable; if not, by compactness, there is $n \geq 0$ that separates Γ_n and Δ_n , against (1). In particular, Γ^* and Δ^* are consistent: for if the former or the latter is inconsistent, then they are separated by $\exists x x \neq x$ or $\forall x x = x$, respectively.

We now show that Γ^* is maximally consistent in $\mathcal{L}'_1$ and likewise Δ^* in $\mathcal{L}'_2$. For the former, suppose that $\varphi_n \notin \Gamma^*$ and $\neg\varphi_n \notin \Gamma^*$, for some $n \geq 0$. If $\varphi_n \notin \Gamma^*$ then $\Gamma_n \cup \{\varphi_n\}$ is separable from Δ_n , and so there is $\chi \in \mathcal{L}'_0$ such that both:

$$\Gamma^* \models \varphi_n \rightarrow \chi, \quad \Delta^* \models \neg\chi.$$

Likewise, if $\neg\varphi_n \notin \Gamma^*$, there is $\chi' \in \mathcal{L}'_0$ such that both:

$$\Gamma^* \models \neg\varphi_n \rightarrow \chi', \quad \Delta^* \models \neg\chi'.$$

By propositional logic, $\Gamma^* \models \chi \vee \chi'$ and $\Delta^* \models \neg(\chi \vee \chi')$, so $\chi \vee \chi'$ separates Γ^* and Δ^* . A similar argument establishes that Δ^* is maximal.

Finally, we show that $\Gamma^* \cap \Delta^*$ is maximally consistent in $\mathcal{L}'_0$. It is obviously consistent, since it is the intersection of consistent sets. To show maximality, let $\sigma \in \mathcal{L}'_0$. Now, Γ^* is maximal in $\mathcal{L}'_1 \supseteq \mathcal{L}'_0$, and similarly Δ^* is maximal in $\mathcal{L}'_2 \supseteq \mathcal{L}'_0$. It follows that either $\sigma \in \Gamma^*$ or $\neg\sigma \in \Gamma^*$, and either $\sigma \in \Delta^*$ or $\neg\sigma \in \Delta^*$. If $\sigma \in \Gamma^*$ and $\neg\sigma \in \Delta^*$ then σ would separate Γ^* and Δ^* ; and if $\neg\sigma \in \Gamma^*$ and $\sigma \in \Delta^*$ then Γ^* and Δ^* would be separated by $\neg\sigma$. Hence, either $\sigma \in \Gamma^* \cap \Delta^*$ or $\neg\sigma \in \Gamma^* \cap \Delta^*$, and $\Gamma^* \cap \Delta^*$ is maximal.

Since Γ^* is maximally consistent, it has a model $\mathfrak{M}'_1$ whose **domínio** $|\mathfrak{M}'_1|$ comprises all and only the elements $c^{\mathfrak{M}'_1}$ interpreting the **símbolos de constante**—just like in the proof of the completeness theorem (**Teorema 23.20**). Similarly, Δ^* has a model $\mathfrak{M}'_2$ whose **domínio** $|\mathfrak{M}'_2|$ is given by the interpretations $c^{\mathfrak{M}'_2}$ of the **símbolos de constante**.

Let $\mathfrak{M}_1$ be obtained from $\mathfrak{M}'_1$ by dropping interpretations for **símbolos de constante**, **símbolos de função**, and **símbolos de predicado** in $\mathcal{L}'_1 \setminus \mathcal{L}'_0$, and similarly for $\mathfrak{M}_2$. Then the map $h: M_1 \rightarrow M_2$ defined by $h(c^{\mathfrak{M}'_1}) = c^{\mathfrak{M}'_2}$ is an isomorphism in $\mathcal{L}'_0$, because $\Gamma^* \cap \Delta^*$ is maximally consistent in $\mathcal{L}'_0$, as shown. This follows because any $\mathcal{L}'_0$ -**sentença** either belongs to both Γ^* and Δ^* , or to neither: so $c^{\mathfrak{M}'_1} \in P^{\mathfrak{M}'_1}$ if and only if $P(c) \in \Gamma^*$ if and only if $P(c) \in \Delta^*$ if and only if $c^{\mathfrak{M}'_2} \in P^{\mathfrak{M}'_2}$. The other conditions satisfied by isomorphisms can be established similarly.

Let us now define a model $\mathfrak{M}$ for the **linguagem** $\mathcal{L}_1 \cup \mathcal{L}_2$ as follows:

1. The **domínio** $|\mathfrak{M}|$ is just $|\mathfrak{M}_2|$, O.s., the set of all elements $c^{\mathfrak{M}'_2}$;
2. If a **símbolo de predicado** P is in $\mathcal{L}_2 \setminus \mathcal{L}_1$ then $P^{\mathfrak{M}} = P^{\mathfrak{M}'_2}$;
3. If a predicate P is in $\mathcal{L}_1 \setminus \mathcal{L}_2$ then $P^{\mathfrak{M}} = h(P^{\mathfrak{M}'_2})$, O.s., $\langle c_1^{\mathfrak{M}'_2}, \dots, c_n^{\mathfrak{M}'_2} \rangle \in P^{\mathfrak{M}}$ if and only if $\langle c_1^{\mathfrak{M}'_1}, \dots, c_n^{\mathfrak{M}'_1} \rangle \in P^{\mathfrak{M}'_1}$.
4. If a **símbolo de predicado** P is in $\mathcal{L}_0$ then $P^{\mathfrak{M}} = P^{\mathfrak{M}'_2} = h(P^{\mathfrak{M}'_1})$.
5. **Símbolos de função** of $\mathcal{L}_1 \cup \mathcal{L}_2$, including **símbolos de constante**, are handled similarly.

Finally, one shows by induction on **fórmulas** that $\mathfrak{M}$ agrees with $\mathfrak{M}'_1$ on all **fórmulas** of $\mathcal{L}'_1$ and with $\mathfrak{M}'_2$ on all **fórmulas** of $\mathcal{L}'_2$. In particular, $\mathfrak{M} \models \Gamma^* \cup \Delta^*$, whence $\mathfrak{M} \models \varphi$ and $\mathfrak{M} \models \neg\psi$, and $\not\models \varphi \rightarrow \psi$. This concludes the proof of Craig's Interpolation Theorem. $\square$

-NoValue-

content/model-theory/interpolation/definability.tex

27.4 The Definability Theorem

One important application of the interpolation theorem is Beth's definability theorem. To define an n -place relation R we can give a **fórmula** χ with n free

mod:int:def:
sec

27.4. THE DEFINABILITY THEOREM

variáveis which does not involve R . This would be an *explicit* definition of R in terms of χ . We can then say also that a theory $\Sigma(P)$ in a *linguagem* containing the n -place *símbolo de predicado* P explicitly defines P if it contains (or at least entails) a formalized explicit definition, O.s.,

$$\Sigma(P) \models \forall x_1 \dots \forall x_n (P(x_1, \dots, x_n) \leftrightarrow \chi(x_1, \dots, x_n)).$$

But an explicit definition is only one way of defining—in the sense of determining completely—a relation. A theory may also be such that the interpretation of P is fixed by the interpretation of the rest of the *linguagem* in any model. The definability theorem states that whenever a theory fixes the interpretation of P in this way—whenever it *implicitly defines* P —then it also explicitly defines it.

Definição 27.5. Suppose $\mathcal{L}$ is a *linguagem* not containing the *símbolo de predicado* P . A set $\Sigma(P)$ of *sentenças* of $\mathcal{L} \cup \{P\}$ *explicitly defines* P if and only if there is a *fórmula* $\chi(x_1, \dots, x_n)$ of $\mathcal{L}$ such that

$$\Sigma(P) \models \forall x_1 \dots \forall x_n (P(x_1, \dots, x_n) \leftrightarrow \chi(x_1, \dots, x_n)).$$

Definição 27.6. Suppose $\mathcal{L}$ is a *linguagem* not containing the *símbolos de predicado* P and P' . A set $\Sigma(P)$ of *sentenças* of $\mathcal{L} \cup \{P\}$ *implicitly defines* P if and only if

$$\Sigma(P) \cup \Sigma(P') \models \forall x_1 \dots \forall x_n (P(x_1, \dots, x_n) \leftrightarrow P'(x_1, \dots, x_n)),$$

where $\Sigma(P')$ is the result of uniformly replacing P with P' in $\Sigma(P)$.

In other words, for any model $\mathfrak{M}$ and $R, R' \subseteq |\mathfrak{M}|^n$, if both $(\mathfrak{M}, R) \models \Sigma(P)$ and $(\mathfrak{M}, R') \models \Sigma(P')$, then $R = R'$; where $(\mathfrak{M}, R)$ is the *estrutura* $\mathfrak{M}'$ for the expansion of $\mathcal{L}$ to $\mathcal{L} \cup \{P\}$ such that $P^{\mathfrak{M}'} = R$, and similarly for $(\mathfrak{M}, R')$.

Teorema 27.7 (Beth Definability Theorem). A set $\Sigma(P)$ of $\mathcal{L} \cup \{P\}$ -*fórmulas* *implicitly defines* P if and only if $\Sigma(P)$ *explicitly defines* P .

Prova. If $\Sigma(P)$ explicitly defines P then both

$$\begin{aligned} \Sigma(P) \models & \quad \forall x_1 \dots \forall x_n (P(x_1, \dots, x_n) \leftrightarrow \chi(x_1, \dots, x_n)) \\ \Sigma(P') \models & \quad \forall x_1 \dots \forall x_n (P'(x_1, \dots, x_n) \leftrightarrow \chi(x_1, \dots, x_n)) \end{aligned}$$

and the conclusion follows. For the converse: assume that $\Sigma(P)$ implicitly defines P . First, we add *símbolos de constante* $c_1, \dots, c_n$ to $\mathcal{L}$. Then

$$\Sigma(P) \cup \Sigma(P') \models P(c_1, \dots, c_n) \rightarrow P'(c_1, \dots, c_n).$$

By compactness, there are finite sets $\Delta_0 \subseteq \Sigma(P)$ and $\Delta_1 \subseteq \Sigma(P')$ such that

$$\Delta_0 \cup \Delta_1 \models P(c_1, \dots, c_n) \rightarrow P'(c_1, \dots, c_n).$$

Let $\theta(P)$ be the conjunction of all **sentenças** $\varphi(P)$ such that either $\varphi(P) \in \Delta_0$ or $\varphi(P') \in \Delta_1$ and let $\theta(P')$ be the conjunction of all **sentenças** $\varphi(P')$ such that either $\varphi(P) \in \Delta_0$ or $\varphi(P') \in \Delta_1$. Then $\theta(P) \wedge \theta(P') \models P(c_1, \dots, c_n) \rightarrow P'c_1 \dots c_n$. We can re-arrange this so that each **símbolo de predicado** occurs on one side of $\models$:

$$\theta(P) \wedge P(c_1, \dots, c_n) \models \theta(P') \rightarrow P'(c_1, \dots, c_n).$$

By Craig's Interpolation Theorem there is a **sentença** $\chi(c_1, \dots, c_n)$ not containing P or P' such that:

$$\begin{aligned} \theta(P) \wedge P(c_1, \dots, c_n) &\models \chi(c_1, \dots, c_n); \\ \chi(c_1, \dots, c_n) &\models \theta(P') \rightarrow P'(c_1, \dots, c_n). \end{aligned}$$

From the former of these two entailments we have: $\theta(P) \models P(c_1, \dots, c_n) \rightarrow \chi(c_1, \dots, c_n)$. And from the latter, since an $\mathcal{L} \cup \{P\}$ -model $(\mathfrak{M}, R) \models \varphi(P)$ if and only if the corresponding $\mathcal{L} \cup \{P'\}$ -model $(\mathfrak{M}, R) \models \varphi(P')$, we have $\chi(c_1, \dots, c_n) \models \theta(P') \rightarrow P'(c_1, \dots, c_n)$, from which:

$$\theta(P) \models \chi(c_1, \dots, c_n) \rightarrow P(c_1, \dots, c_n).$$

Putting the two together, $\theta(P) \models P(c_1, \dots, c_n) \leftrightarrow \chi(c_1, \dots, c_n)$, and by monotonicity and generalization also

$$\Sigma(P) \models \forall x_1 \dots \forall x_n (P(x_1, \dots, x_n) \leftrightarrow \chi(x_1, \dots, x_n)). \quad \square$$

Capítulo 28

Lindström's Theorem

-NoValue-

[content/model-theory/lindstrom/introduction.tex](#)

28.1 Introduction

In this chapter we aim to prove Lindström's characterization of first-order logic as the maximal logic for which (given certain further constraints) the Compactness and the Downward Löwenheim–Skolem theorems hold (**Teorema 23.23** and

Teorema 23.32). First, we need a more general characterization of the general class of logics to which the theorem applies. We will restrict ourselves to *relational* languages, O.s., languages which only contain **símbolos de predicado** and individual constants, but no **símbolos de função**.

-NoValue-

content/model-theory/lindstrom/abstract-logics.tex

28.2 Abstract Logics

mod:lin:alg:
sec

Definição 28.1. An *abstract logic* is a pair $\langle L, \models_L \rangle$, where L is a function that assigns to each **linguagem** $\mathcal{L}$ a set $L(\mathcal{L})$ of **sentenças**, and $\models_L$ is a relation between **estruturas** for the **linguagem** $\mathcal{L}$ and **membros** of $L(\mathcal{L})$. In particular, $\langle F, \models \rangle$ is ordinary first-order logic, O.s., F is the function assigning to the **linguagem** $\mathcal{L}$ the set of first-order **sentenças** built from the constants in $\mathcal{L}$, and $\models$ is the satisfaction relation of first-order logic.

Notice that we are still employing the same notion of **estrutura** for a given **linguagem** as for first-order logic, but we do not presuppose that **sentenças** are build up from the basic symbols in $\mathcal{L}$ in the usual way, nor that the relation $\models_L$ is recursively defined in the same way as for first-order logic. So for instance the definition, being completely general, is intended to capture the case where **sentenças** in $\langle L, \models_L \rangle$ contain infinitely long conjunctions or disjunction, or quantifiers other than $\exists$ and $\forall$ (p.ex., “there are infinitely many x such that ...”), or perhaps infinitely long quantifier prefixes. To emphasize that “**sentenças**” in $L(\mathcal{L})$ need not be ordinary **sentenças** of first-order logic, in this chapter we use **variáveis** $\alpha, \beta, \dots$ to range over them, and reserve $\varphi, \psi, \dots$ for ordinary first-order **fórmulas**.

Definição 28.2. Let $\text{Mod}_L(\alpha)$ denote the class $\{\mathfrak{M} : \mathfrak{M} \models_L \alpha\}$. If the **linguagem** needs to be made explicit, we write $\text{Mod}_L^{\mathcal{L}}(\alpha)$. Two **estruturas** $\mathfrak{M}$ and $\mathfrak{N}$ for $\mathcal{L}$ are *elementarily equivalent* in $\langle L, \models_L \rangle$, written $\mathfrak{M} \equiv_L \mathfrak{N}$, if the same **sentenças** from $L(\mathcal{L})$ are true in each.

Definição 28.3. An abstract logic $\langle L, \models_L \rangle$ for the **linguagem** $\mathcal{L}$ is *normal* if it satisfies the following properties:

1. (*L-Monotonicity*) For **linguagens** $\mathcal{L}$ and $\mathcal{L}'$, if $\mathcal{L} \subseteq \mathcal{L}'$, then $L(\mathcal{L}) \subseteq L(\mathcal{L}')$.
2. (*Expansion Property*) For each $\alpha \in L(\mathcal{L})$ there is a *finite* subset $\mathcal{L}'$ of $\mathcal{L}$ such that the relation $\mathfrak{M} \models_L \alpha$ depends only on the reduct of $\mathfrak{M}$ to $\mathcal{L}'$; O.s., if $\mathfrak{M}$ and $\mathfrak{N}$ have the same reduct to $\mathcal{L}'$ then $\mathfrak{M} \models_L \alpha$ if and only if $\mathfrak{N} \models_L \alpha$.
3. (*Isomorphism Property*) If $\mathfrak{M} \models_L \alpha$ and $\mathfrak{M} \simeq \mathfrak{N}$ then also $\mathfrak{N} \models_L \alpha$.

4. (*Renaming Property*) The relation $\models_L$ is preserved under renaming: if the **linguagem** $\mathcal{L}'$ is obtained from $\mathcal{L}$ by replacing each symbol P by a symbol P' of the same arity and each constant c by a distinct constant c' , then for each **estrutura** $\mathfrak{M}$ and **sentença** α , $\mathfrak{M} \models_L \alpha$ if and only if $\mathfrak{M}' \models_L \alpha'$, where $\mathfrak{M}'$ is the $\mathcal{L}'$ -**estrutura** corresponding to $\mathcal{L}$ and $\alpha' \in L(\mathcal{L}')$.
5. (*Boolean Property*) The abstract logic $\langle L, \models_L \rangle$ is closed under the Boolean connectives in the sense that for each $\alpha \in L(\mathcal{L})$ there is a $\beta \in L(\mathcal{L})$ such that $\mathfrak{M} \models_L \beta$ if and only if $\mathfrak{M} \not\models_L \alpha$, and for each α and β there is a γ such that $\text{Mod}_L(\gamma) = \text{Mod}_L(\alpha) \cap \text{Mod}_L(\beta)$. Similarly for atomic **fórmulas** and the other connectives.
6. (*Quantifier Property*) For each constant c in $\mathcal{L}$ and $\alpha \in L(\mathcal{L})$ there is a $\beta \in L(\mathcal{L})$ such that

$$\text{Mod}_L^{\mathcal{L}'}(\beta) = \{\mathfrak{M} : (\mathfrak{M}, a) \in \text{Mod}_L^{\mathcal{L}}(\alpha) \text{ for some } a \in |\mathfrak{M}|\},$$

where $\mathcal{L}' = \mathcal{L} \setminus \{c\}$ and $(\mathfrak{M}, a)$ is the expansion of $\mathfrak{M}$ to $\mathcal{L}$ assigning a to c .

7. (*Relativization Property*) Given a **sentença** $\alpha \in L(\mathcal{L})$ and symbols $R, c_1, \dots, c_n$ not in $\mathcal{L}$, there is a **sentença** $\beta \in L(\mathcal{L} \cup \{R, c_1, \dots, c_n\})$ called the *relativization* of α to $R(x, c_1, \dots, c_n)$, such that for each **estrutura** $\mathfrak{M}$:

$$(\mathfrak{M}, X, b_1, \dots, b_n) \models_L \beta \text{ if and only if } \mathfrak{N} \models_L \alpha,$$

where $\mathfrak{N}$ is the substructure of $\mathfrak{M}$ with **domínio** $|\mathfrak{N}| = \{a \in |\mathfrak{M}| : R^{\mathfrak{M}}(a, b_1, \dots, b_n)\}$ (see **Observação 1**), and $(\mathfrak{M}, X, b_1, \dots, b_n)$ is the expansion of $\mathfrak{M}$ interpreting $R, c_1, \dots, c_n$ by $X, b_1, \dots, b_n$, respectively (with $X \subseteq M^{n+1}$).

Definição 28.4. Given two abstract logics $\langle L_1, \models_{L_1} \rangle$ and $\langle L_2, \models_{L_2} \rangle$ we say that the latter is *at least as expressive* as the former, written $\langle L_1, \models_{L_1} \rangle \leq \langle L_2, \models_{L_2} \rangle$, if for each **linguagem** $\mathcal{L}$ and **sentença** $\alpha \in L_1(\mathcal{L})$ there is a **sentença** $\beta \in L_2(\mathcal{L})$ such that $\text{Mod}_{L_1}^{\mathcal{L}}(\alpha) = \text{Mod}_{L_2}^{\mathcal{L}}(\beta)$. The logics $\langle L_1, \models_{L_1} \rangle$ and $\langle L_2, \models_{L_2} \rangle$ are *equivalent* if $\langle L_1, \models_{L_1} \rangle \leq \langle L_2, \models_{L_2} \rangle$ and $\langle L_2, \models_{L_2} \rangle \leq \langle L_1, \models_{L_1} \rangle$.

Observação 5. First-order logic, O.s., the abstract logic $\langle F, \models \rangle$, is normal. In fact, the above properties are mostly straightforward for first-order logic. We just remark that the expansion property comes down to extensionality, and that the relativization of a **sentença** α to $R(x, c_1, \dots, c_n)$ is obtained by replacing each **subfórmula** $\forall x \beta$ by $\forall x (R(x, c_1, \dots, c_n) \rightarrow \beta)$. Moreover, if $\langle L, \models_L \rangle$ is normal, then $\langle F, \models \rangle \leq \langle L, \models_L \rangle$, as can be shown by induction on first-order **fórmulas**. Accordingly, with no loss in generality, we can assume that every first-order **sentença** belongs to every normal logic.

-NoValue-

28.3 Compactness and Löwenheim–Skolem Properties

mod:lin:lsp:
sec We now give the obvious extensions of compactness and Löwenheim–Skolem to the case of abstract logics.

Definição 28.5. An abstract logic $\langle L, \models_L \rangle$ has the *Compactness Property* if each set Γ of $L(\mathcal{L})$ -*sentenças* is satisfiable whenever each finite $\Gamma_0 \subseteq \Gamma$ is satisfiable.

Definição 28.6. $\langle L, \models_L \rangle$ has the *Downward Löwenheim–Skolem property* if any satisfiable Γ has a *enumerável* model.

The notion of partial isomorphism from **Definição 25.15** is purely “algebraic” (O.s., given without reference to the *sentenças* of the language but only to the constants provided by the *linguagem* $\mathcal{L}$ of the *estruturas*), and hence it applies to the case of abstract logics. In case of first-order logic, we know from **Teorema 25.17** that if two *estruturas* are partially isomorphic then they are elementarily equivalent. That proof does not carry over to abstract logics, for induction on *fórmulas* need not be available for arbitrary $\alpha \in L(\mathcal{L})$, but the theorem is true nonetheless, provided the Löwenheim–Skolem property holds.

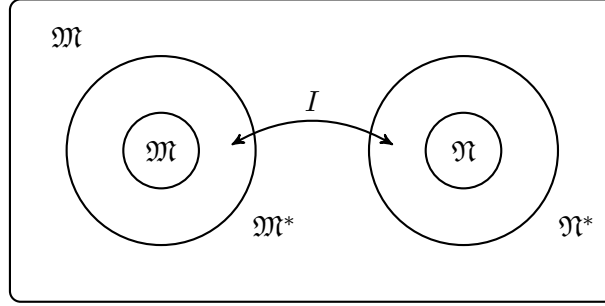
mod:lin:lsp:
thm:abstract-p-isom **Teorema 28.7.** Suppose $\langle L, \models_L \rangle$ is a normal logic with the Löwenheim–Skolem property. Then any two *estruturas* that are partially isomorphic are elementarily equivalent in $\langle L, \models_L \rangle$.

Prova. Suppose $\mathfrak{M} \simeq_p \mathfrak{N}$, but for some α also $\mathfrak{M} \models_L \alpha$ while $\mathfrak{N} \not\models_L \alpha$. By the Isomorphism Property we can assume that $|\mathfrak{M}|$ and $|\mathfrak{N}|$ are disjoint, and by the Expansion Property we can assume that $\alpha \in L(\mathcal{L})$ for a finite *linguagem* $\mathcal{L}$. Let $\mathcal{I}$ be a set of partial isomorphisms between $\mathfrak{M}$ and $\mathfrak{N}$, and with no loss of generality also assume that if $p \in \mathcal{I}$ and $q \subseteq p$ then also $q \in \mathcal{I}$.

$|\mathfrak{M}|^{<\omega}$ is the set of finite sequences of *membros* of $|\mathfrak{M}|$. Let S be the ternary relation over $|\mathfrak{M}|^{<\omega}$ representing concatenation, O.s., if $\mathbf{a}, \mathbf{b}, \mathbf{c} \in |\mathfrak{M}|^{<\omega}$ then $S(\mathbf{a}, \mathbf{b}, \mathbf{c})$ holds if and only if $\mathbf{c}$ is the concatenation of $\mathbf{a}$ and $\mathbf{b}$; and let T be the ternary relation such that $T(\mathbf{a}, \mathbf{b}, \mathbf{c})$ holds for $\mathbf{b} \in M$ and $\mathbf{a}, \mathbf{c} \in |\mathfrak{M}|^{<\omega}$ if and only if $\mathbf{a} = a_1, \dots, a_n$ and $\mathbf{c} = a_1, \dots, a_n, b$. Pick new 3-place *símbolos de predicado* P and Q and form the *estrutura* $\mathfrak{M}^*$ having the universe $|\mathfrak{M}| \cup |\mathfrak{M}|^{<\omega}$, having $\mathfrak{M}$ as a substructure, and interpreting P and Q by the concatenation relations S and T (so $\mathfrak{M}^*$ is in the *linguagem* $\mathcal{L} \cup \{P, Q\}$).

Define $|\mathfrak{N}|^{<\omega}$, S' , T' , P' , Q' and $\mathfrak{N}^*$ analogously. Since by hypothesis $\mathfrak{M} \simeq_p \mathfrak{N}$, there is a relation I between $|\mathfrak{M}|^{<\omega}$ and $|\mathfrak{N}|^{<\omega}$ such that $I(\mathbf{a}, \mathbf{b})$ holds if and only if $\mathbf{a}$ and $\mathbf{b}$ are isomorphic and satisfy the back-and-forth condition of **Definição 25.15**. Now, let $\mathfrak{M}$ be the *estrutura* whose *domínio* is the union of the *domínios* of $\mathfrak{M}^*$ and $\mathfrak{N}^*$, having $\mathfrak{M}^*$ and $\mathfrak{N}^*$ as *subestruturas*, in the *linguagem* with one extra binary *símbolo de predicado* R interpreted by the relation I and *símbolos de predicado* denoting the *domínios* $|\mathfrak{M}|^*$ and $|\mathfrak{N}|^*$.

The crucial observation is that in the *linguagem* of the *estrutura* $\mathfrak{M}$ there is a *first-order sentença* θ_1 true in $\mathfrak{M}$ saying that $\mathfrak{M} \models_L \alpha$ and $\mathfrak{N} \not\models_L \alpha$

Figura 28.1: The *estrutura* $\mathfrak{M}$ with the internal partial isomorphism.

(this requires the Relativization Property), as well as a *first-order sentença* θ_2 true in $\mathfrak{M}$ saying that $\mathfrak{M} \simeq_p \mathfrak{N}$ via the partial isomorphism I . By the Löwenheim–Skolem Property, θ_1 and θ_2 are jointly true in a *enumerável* model $\mathfrak{M}_0$ containing partially isomorphic substructures $\mathfrak{M}_0$ and $\mathfrak{N}_0$ such that $\mathfrak{M}_0 \models_L \alpha$ and $\mathfrak{N}_0 \not\models_L \alpha$. But *enumerável* partially isomorphic *estruturas* are in fact isomorphic by [Teorema 25.16](#), contradicting the Isomorphism Property of normal abstract logics. $\square$

-NoValue-

content/model-theory/lindstrom/lindstrom-proof.tex

28.4 Lindström's Theorem

Lema 28.8. *Suppose $\alpha \in L(\mathcal{L})$, with $\mathcal{L}$ finite, and assume also that there is an $n \in \mathbb{N}$ such that for any two *estruturas* $\mathfrak{M}$ and $\mathfrak{N}$, if $\mathfrak{M} \equiv_n \mathfrak{N}$ and $\mathfrak{M} \models_L \alpha$ then also $\mathfrak{N} \models_L \alpha$. Then α is equivalent to a first-order *sentença*, O.s., there is a first-order θ such that $\text{Mod}_L(\alpha) = \text{Mod}_L(\theta)$.*

mod:lin:prf:
sec
mod:lin:prf:
lem:lindstrom

Prova. Let n be such that any two n -equivalent *estruturas* $\mathfrak{M}$ and $\mathfrak{N}$ agree on the value assigned to α . Recall [Proposição 25.19](#): there are only finitely many first-order *sentenças* in a finite *linguagem* that have quantifier rank no greater than n , up to logical equivalence. Now, for each fixed *estrutura* $\mathfrak{M}$ let $\theta_{\mathfrak{M}}$ be the conjunction of all first-order *sentenças* α true in $\mathfrak{M}$ with $\text{qr}(\alpha) \leq n$ (this conjunction is finite), so that $\mathfrak{N} \models \theta_{\mathfrak{M}}$ if and only if $\mathfrak{N} \equiv_n \mathfrak{M}$. Then put $\theta = \bigvee \{\theta_{\mathfrak{M}} : \mathfrak{M} \models_L \alpha\}$; this disjunction is also finite (up to logical equivalence).

The conclusion $\text{Mod}_L(\alpha) = \text{Mod}_L(\theta)$ follows. In fact, if $\mathfrak{N} \models_L \theta$ then for some $\mathfrak{M} \models_L \alpha$ we have $\mathfrak{N} \models \theta_{\mathfrak{M}}$, whence also $\mathfrak{N} \models_L \alpha$ (by the hypothesis of the lemma). Conversely, if $\mathfrak{N} \models_L \alpha$ then $\theta_{\mathfrak{N}}$ is a disjunct in θ , and since $\mathfrak{N} \models \theta_{\mathfrak{N}}$, also $\mathfrak{N} \models_L \theta$. $\square$

28.4. LINDSTRÖM'S THEOREM

*mod:lin:prf:
thm:lindstrom*

Teorema 28.9 (Lindström's Theorem). *Suppose $\langle L, \models_L \rangle$ has the Compactness and the Löwenheim–Skolem Properties. Then $\langle L, \models_L \rangle \leq \langle F, \models \rangle$ (so $\langle L, \models_L \rangle$ is equivalent to first-order logic).*

Prova. By **Lema 28.8**, it suffices to show that for any $\alpha \in L(\mathcal{L})$, with $\mathcal{L}$ finite, there is $n \in \mathbb{N}$ such that for any two **estruturas** $\mathfrak{M}$ and $\mathfrak{N}$: if $\mathfrak{M} \equiv_n \mathfrak{N}$ then $\mathfrak{M}$ and $\mathfrak{N}$ agree on α . For then α is equivalent to a first-order **sentença**, from which $\langle L, \models_L \rangle \leq \langle F, \models \rangle$ follows. Since we are working in a finite, purely relational **linguagem**, by **Teorema 25.23** we can replace the statement that $\mathfrak{M} \equiv_n \mathfrak{N}$ by the corresponding algebraic statement that $I_n(\emptyset, \emptyset)$.

Given α , suppose towards a contradiction that for each n there are **estruturas** $\mathfrak{M}_n$ and $\mathfrak{N}_n$ such that $I_n(\emptyset, \emptyset)$, but (say) $\mathfrak{M}_n \models_L \alpha$ whereas $\mathfrak{N}_n \not\models_L \alpha$. By the Isomorphism Property we can assume that all the $\mathfrak{M}_n$'s interpret the constants of the language by the same objects; furthermore, since there are only finitely many atomic **sentenças** in the language, we may also assume that they satisfy the same atomic **sentenças** (we can take a subsequence of the $\mathfrak{M}$'s otherwise). Let $\mathfrak{M}$ be the union of all the $\mathfrak{M}_n$'s, O.s., the unique minimal **estrutura** having each $\mathfrak{M}_n$ as a substructure. As in the proof of **Teorema 28.7**, let $\mathfrak{M}^*$ be the extension of $\mathfrak{M}$ with **domínio** $|\mathfrak{M}| \cup |\mathfrak{M}|^{<\omega}$, in the expanded **linguagem** comprising the concatenation predicates P and Q .

Similarly, define $\mathfrak{N}_n$, $\mathfrak{N}$ and $\mathfrak{N}^*$. Now let $\mathfrak{M}$ be the **estrutura** whose **domínio** comprises the **domínios** of $\mathfrak{M}^*$ and $\mathfrak{N}^*$ as well as the natural numbers $\mathbb{N}$ along with their natural ordering $\leq$, in the **linguagem** with extra predicates representing the **domínios** $|\mathfrak{M}|$, $|\mathfrak{N}|$, $|\mathfrak{M}|^{<\omega}$ and $|\mathfrak{N}|^{<\omega}$ as well as predicates coding the domains of $\mathfrak{M}_n$ and $\mathfrak{N}_n$ in the sense that:

$$\begin{aligned} |\mathfrak{M}_n| &= \{a \in |\mathfrak{M}| : R(a, n)\}; & |\mathfrak{N}_n| &= \{a \in |\mathfrak{N}| : S(a, n)\}; \\ |\mathfrak{M}|_n^{<\omega} &= \{a \in |\mathfrak{M}|^{<\omega} : R(a, n)\}; & |\mathfrak{N}|_n^{<\omega} &= \{a \in |\mathfrak{N}|^{<\omega} : S(a, n)\}. \end{aligned}$$

The **estrutura** $\mathfrak{M}$ also has a ternary relation J such that $J(n, \mathbf{a}, \mathbf{b})$ holds if and only if $I_n(\mathbf{a}, \mathbf{b})$.

Now there is a **sentença** θ in the **linguagem** $\mathcal{L}$ augmented by R, S, J , etc., saying that $\leq$ is a discrete linear ordering with first but no last element and such that $\mathfrak{M}_n \models \alpha$, $\mathfrak{N}_n \not\models \alpha$, and for each n in the ordering, $J(n, \mathbf{a}, \mathbf{b})$ holds if and only if $I_n(\mathbf{a}, \mathbf{b})$.

Using the Compactness Property, we can find a model $\mathfrak{M}^*$ of θ in which the ordering contains a non-standard element n^* . In particular then $\mathfrak{M}^*$ will contain sub**estruturas** $\mathfrak{M}_{n^*}$ and $\mathfrak{N}_{n^*}$ such that $\mathfrak{M}_{n^*} \models_L \alpha$ and $\mathfrak{N}_{n^*} \not\models_L \alpha$. But now we can define a set $\mathcal{I}$ of pairs of k -tuples from $|\mathfrak{M}_{n^*}|$ and $|\mathfrak{N}_{n^*}|$ by putting $\langle \mathbf{a}, \mathbf{b} \rangle \in \mathcal{I}$ if and only if $J(n^* - k, \mathbf{a}, \mathbf{b})$, where k is the length of $\mathbf{a}$ and $\mathbf{b}$. Since n^* is non-standard, for each standard k we have that $n^* - k > 0$, and the set $\mathcal{I}$ witnesses the fact that $\mathfrak{M}_{n^*} \simeq_p \mathfrak{N}_{n^*}$. But by **Teorema 28.7**, $\mathfrak{M}_{n^*}$ is L -equivalent to $\mathfrak{N}_{n^*}$, a contradiction. $\square$

Parte V

Computability

This part is based on Jeremy Avigad's notes on computability theory. Only the chapter on recursive functions contains exercises yet, and everything could stand to be expanded with motivation, examples, details, and exercises.

Capítulo 29

Recursive Functions

These are Jeremy Avigad's notes on recursive functions, revised and expanded by Richard Zach. This chapter does contain some exercises, and can be included independently to provide the basis for a discussion of arithmetization of syntax.

-NoValue- [content/computability/recursive-functions/introduction.tex](#)

29.1 Introduction

In order to develop a mathematical theory of computability, one has to, first of all, develop a *model* of computability. We now think of computability as the kind of thing that computers do, and computers work with symbols. But at the beginning of the development of theories of computability, the paradigmatic example of computation was *numerical* computation. Mathematicians were always interested in number-theoretic functions, O.s., functions $f: \mathbb{N}^n \rightarrow \mathbb{N}$ that can be computed. So it is not surprising that at the beginning of the theory of computability, it was such functions that were studied. The most familiar

[cmp:rec:int:
sec](#)

29.2. PRIMITIVE RECURSION

examples of computable numerical functions, such as addition, multiplication, exponentiation (of natural numbers) share an interesting feature: they can be defined *recursively*. It is thus quite natural to attempt a general definition of *computable function* on the basis of recursive definitions. Among the many possible ways to define number-theoretic functions recursively, one particularly simple pattern of definition here becomes central: so-called *primitive recursion*.

In addition to computable functions, we might be interested in computable sets and relations. A set is computable if we can compute the answer to whether or not a given number is **um membro** of the set, and a relation is computable iff we can compute whether or not a tuple $\langle n_1, \dots, n_k \rangle$ is **um membro** of the relation. By considering the *characteristic function* of a set or relation, discussion of computable sets and relations can be subsumed under that of computable functions. Thus we can define primitive recursive relations as well, p.ex., the relation “ n evenly divides m ” is a primitive recursive relation.

Primitive recursive functions—those that can be defined using just primitive recursion—are not, however, the only computable number-theoretic functions. Many generalizations of primitive recursion have been considered, but the most powerful and widely-accepted additional way of computing functions is by unbounded search. This leads to the definition of *partial recursive functions*, and a related definition to *general recursive functions*. General recursive functions are computable and total, and the definition characterizes exactly the partial recursive functions that happen to be total. Recursive functions can simulate every other model of computation (Turing machines, lambda calculus, etc.) and so represent one of the many accepted models of computation.

-NoValue-

[content/computability/recursive-functions/primitive-recursion.tex](#)

29.2 Primitive Recursion

cmp:rec:pre: A characteristic of the natural numbers is that every natural number can be
sec reached from 0 by applying the successor operation $+1$ finitely many times—any natural number is either 0 or the successor of $\dots$ the successor of 0. One way to specify a function $h: \mathbb{N} \rightarrow \mathbb{N}$ that makes use of this fact is this: (a) specify what the value of h is for argument 0, and (b) also specify how to, given the value of $h(x)$, compute the value of $h(x+1)$. For (a) tells us directly what $h(0)$ is, so h is defined for 0. Now, using the instruction given by (b) for $x = 0$, we can compute $h(1) = h(0+1)$ from $h(0)$. Using the same instructions for $x = 1$, we compute $h(2) = h(1+1)$ from $h(1)$, and so on. For every natural number x , we'll eventually reach the step where we define $h(x)$ from $h(x+1)$, and so $h(x)$ is defined for all $x \in \mathbb{N}$.

For instance, suppose we specify $h: \mathbb{N} \rightarrow \mathbb{N}$ by the following two equations:

$$\begin{aligned}h(0) &= 1 \\h(x+1) &= 2 \cdot h(x)\end{aligned}$$

If we already know how to multiply, then these equations give us the information required for (a) and (b) above. By successively applying the second equation, we get that

$$\begin{aligned}h(1) &= 2 \cdot h(0) = 2, \\h(2) &= 2 \cdot h(1) = 2 \cdot 2, \\h(3) &= 2 \cdot h(2) = 2 \cdot 2 \cdot 2, \\&\vdots\end{aligned}$$

We see that the function h we have specified is $h(x) = 2^x$.

The characteristic feature of the natural numbers guarantees that there is only one function h that meets these two criteria. A pair of equations like these is called a *definition by primitive recursion* of the function h . It is so-called because we define h “recursively,” O.s., the definition, specifically the second equation, involves h itself on the right-hand-side. It is “primitive” because in defining $h(x+1)$ we only use the value $h(x)$, O.s., the immediately preceding value. This is the simplest way of defining a function on $\mathbb{N}$ recursively.

We can define even more fundamental functions like addition and multiplication by primitive recursion. In these cases, however, the functions in question are 2-place. We fix one of the argument places, and use the other for the recursion. E.g, to define $\text{add}(x, y)$ we can fix x and define the value first for $y = 0$ and then for $y + 1$ in terms of y . Since x is fixed, it will appear on the left and on the right side of the defining equations.

$$\begin{aligned}\text{add}(x, 0) &= x \\ \text{add}(x, y + 1) &= \text{add}(x, y) + 1\end{aligned}$$

These equations specify the value of add for all x and y . To find $\text{add}(2, 3)$, for instance, we apply the defining equations for $x = 2$, using the first to find $\text{add}(2, 0) = 2$, then using the second to successively find $\text{add}(2, 1) = 2 + 1 = 3$, $\text{add}(2, 2) = 3 + 1 = 4$, $\text{add}(2, 3) = 4 + 1 = 5$.

In the definition of add we used $+$ on the right-hand-side of the second equation, but only to add 1. In other words, we used the successor function $\text{succ}(z) = z + 1$ and applied it to the previous value $\text{add}(x, y)$ to define $\text{add}(x, y + 1)$. So we can think of the recursive definition as given in terms of a single function which we apply to the previous value. However, it doesn't hurt—and sometimes is necessary—to allow the function to depend not just on the previous value but also on x and y . Consider:

$$\begin{aligned}\text{mult}(x, 0) &= 0 \\ \text{mult}(x, y + 1) &= \text{add}(\text{mult}(x, y), x)\end{aligned}$$

This is a primitive recursive definition of a function mult by applying the function add to both the preceding value $\text{mult}(x, y)$ and the first argument x . It also defines the function $\text{mult}(x, y)$ for all arguments x and y . For instance, $\text{mult}(2, 3)$ is determined by successively computing $\text{mult}(2, 0)$, $\text{mult}(2, 1)$,

29.3. COMPOSITION

mult(2, 2), and mult(2, 3):

$$\begin{aligned}\text{mult}(2, 0) &= 0 \\ \text{mult}(2, 1) &= \text{mult}(2, 0 + 1) = \text{add}(\text{mult}(2, 0), 2) = \text{add}(0, 2) = 2 \\ \text{mult}(2, 2) &= \text{mult}(2, 1 + 1) = \text{add}(\text{mult}(2, 1), 2) = \text{add}(2, 2) = 4 \\ \text{mult}(2, 3) &= \text{mult}(2, 2 + 1) = \text{add}(\text{mult}(2, 2), 2) = \text{add}(4, 2) = 6\end{aligned}$$

The general pattern then is this: to give a primitive recursive definition of a function $h(x_0, \dots, x_{k-1}, y)$, we provide two equations. The first defines the value of $h(x_0, \dots, x_{k-1}, 0)$ without reference to h . The second defines the value of $h(x_0, \dots, x_{k-1}, y + 1)$ in terms of $h(x_0, \dots, x_{k-1}, y)$, the other arguments $x_0, \dots, x_{k-1}$, and y . Only the immediately preceding value of h may be used in that second equation. If we think of the operations given by the right-hand-sides of these two equations as themselves being functions f and g , then the general pattern to define a new function h by primitive recursion is this:

$$\begin{aligned}h(x_0, \dots, x_{k-1}, 0) &= f(x_0, \dots, x_{k-1}) \\ h(x_0, \dots, x_{k-1}, y + 1) &= g(x_0, \dots, x_{k-1}, y, h(x_0, \dots, x_{k-1}, y))\end{aligned}$$

In the case of add, we have $k = 1$ and $f(x_0) = x_0$ (the identity function), and $g(x_0, y, z) = z + 1$ (the 3-place function that returns the successor of its third argument):

$$\begin{aligned}\text{add}(x_0, 0) &= f(x_0) = x_0 \\ \text{add}(x_0, y + 1) &= g(x_0, y, \text{add}(x_0, y)) = \text{succ}(\text{add}(x_0, y))\end{aligned}$$

In the case of mult, we have $f(x_0) = 0$ (the constant function always returning 0) and $g(x_0, y, z) = \text{add}(z, x_0)$ (the 3-place function that returns the sum of its last and first argument):

$$\begin{aligned}\text{mult}(x_0, 0) &= f(x_0) = 0 \\ \text{mult}(x_0, y + 1) &= g(x_0, y, \text{mult}(x_0, y)) = \text{add}(\text{mult}(x_0, y), x_0)\end{aligned}$$

-NoValue- _____
[content/computability/recursive-functions/composition.tex](#)

29.3 Composition

cmp:rec:com: sec If f and g are two one-place functions of natural numbers, we can compose them: $h(x) = f(g(x))$. The new function $h(x)$ is then defined by *composition* from the functions f and g . We'd like to generalize this to functions of more than one argument.

Here's one way of doing this: suppose f is a k -place function, and $g_0, \dots, g_{k-1}$ are k functions which are all n -place. Then we can define a new n -place function h as follows:

$$h(x_0, \dots, x_{n-1}) = f(g_0(x_0, \dots, x_{n-1}), \dots, g_{k-1}(x_0, \dots, x_{n-1}))$$

If f and all g_i are computable, so is h : To compute $h(x_0, \dots, x_{n-1})$, first compute the values $y_i = g_i(x_0, \dots, x_{n-1})$ for each $i = 0, \dots, k-1$. Then feed these values into f to compute $h(x_0, \dots, x_{k-1}) = f(y_0, \dots, y_{k-1})$.

This may seem like an overly restrictive characterization of what happens when we compute a new function using some existing ones. For one thing, sometimes we do not use all the arguments of a function, as when we defined $g(x, y, z) = \text{succ}(z)$ for use in the primitive recursive definition of add. Suppose we are allowed use of the following functions:

$$P_i^n(x_0, \dots, x_{n-1}) = x_i$$

The functions P_i^k are called *projection* functions: P_i^n is an n -place function. Then g can be defined by

$$g(x, y, z) = \text{succ}(P_2^3(x, y, z)).$$

Here the role of f is played by the 1-place function succ , so $k = 1$. And we have one 3-place function P_2^3 which plays the role of g_0 . The result is a 3-place function that returns the successor of the third argument.

The projection functions also allow us to define new functions by reordering or identifying arguments. For instance, the function $h(x) = \text{add}(x, x)$ can be defined by

$$h(x_0) = \text{add}(P_0^1(x_0), P_0^1(x_0)).$$

Here $k = 2$, $n = 1$, the role of $f(y_0, y_1)$ is played by add , and the roles of $g_0(x_0)$ and $g_1(x_0)$ are both played by $P_0^1(x_0)$, the one-place projection function (aka the identity function).

If $f(y_0, y_1)$ is a function we already have, we can define the function $h(x_0, x_1) = f(x_1, x_0)$ by

$$h(x_0, x_1) = f(P_1^2(x_0, x_1), P_0^2(x_0, x_1)).$$

Here $k = 2$, $n = 2$, and the roles of g_0 and g_1 are played by P_1^2 and P_0^2 , respectively.

You may also worry that $g_0, \dots, g_{k-1}$ are all required to have the same arity n . (Remember that the *arity* of a function is the number of arguments; an n -place function has arity n .) But adding the projection functions provides the desired flexibility. For example, suppose f and g are 3-place functions and h is the 2-place function defined by

$$h(x, y) = f(x, g(x, x, y), y).$$

The definition of h can be rewritten with the projection functions, as

$$h(x, y) = f(P_0^2(x, y), g(P_0^2(x, y), P_0^2(x, y), P_1^2(x, y)), P_1^2(x, y)).$$

Then h is the composition of f with P_0^2 , l , and P_1^2 , where

$$l(x, y) = g(P_0^2(x, y), P_0^2(x, y), P_1^2(x, y)),$$

O.s., l is the composition of g with P_0^2 , P_0^2 , and P_1^2 .

-NoValue-

`content/computability/recursive-functions/pr-functions.tex`

29.4 Primitive Recursion Functions

cmp:rec:prf: sec Let us record again how we can define new functions from existing ones using primitive recursion and composition.

cmp:rec:prf: defn:primitive-recursion **Definição 29.1.** Suppose f is a k -place function ($k \geq 1$) and g is a $(k+2)$ -place function. The function defined by *primitive recursion from f and g* is the $(k+1)$ -place function h defined by the equations

$$\begin{aligned} h(x_0, \dots, x_{k-1}, 0) &= f(x_0, \dots, x_{k-1}) \\ h(x_0, \dots, x_{k-1}, y+1) &= g(x_0, \dots, x_{k-1}, y, h(x_0, \dots, x_{k-1}, y)) \end{aligned}$$

cmp:rec:prf: defn:composition **Definição 29.2.** Suppose f is a k -place function, and $g_0, \dots, g_{k-1}$ are k functions which are all n -place. The function defined by *composition from f and $g_0, \dots, g_{k-1}$* is the n -place function h defined by

$$h(x_0, \dots, x_{n-1}) = f(g_0(x_0, \dots, x_{n-1}), \dots, g_{k-1}(x_0, \dots, x_{n-1})).$$

In addition to *succ* and the projection functions

$$P_i^n(x_0, \dots, x_{n-1}) = x_i,$$

for each natural number n and $i < n$, we will include among the primitive recursive functions the function $\text{zero}(x) = 0$.

Definição 29.3. The set of primitive recursive functions is the set of functions from $\mathbb{N}^n$ to $\mathbb{N}$, defined inductively by the following clauses:

1. zero is primitive recursive.
2. *succ* is primitive recursive.
3. Each projection function P_i^n is primitive recursive.
4. If f is a k -place primitive recursive function and $g_0, \dots, g_{k-1}$ are n -place primitive recursive functions, then the composition of f with $g_0, \dots, g_{k-1}$ is primitive recursive.
5. If f is a k -place primitive recursive function and g is a $k+2$ -place primitive recursive function, then the function defined by primitive recursion from f and g is primitive recursive.

Put more concisely, the set of primitive recursive functions is the smallest explanation set containing zero, *succ*, and the projection functions P_j^n , and which is closed under composition and primitive recursion.

Another way of describing the set of primitive recursive functions is by defining it in terms of “stages.” Let S_0 denote the set of starting functions: zero, *succ*, and the projections. These are the primitive recursive functions of stage 0. Once a stage S_i has been defined, let S_{i+1} be the set of all functions

you get by applying a single instance of composition or primitive recursion to functions already in S_i . Then

$$S = \bigcup_{i \in \mathbb{N}} S_i$$

is the set of all primitive recursive functions

Let us verify that add is a primitive recursive function.

Proposição 29.4. *The addition function $\text{add}(x, y) = x + y$ is primitive recursive.*

Prova. We already have a primitive recursive definition of add in terms of two functions f and g which matches the format of [Definição 29.1](#):

$$\begin{aligned} \text{add}(x_0, 0) &= f(x_0) = x_0 \\ \text{add}(x_0, y + 1) &= g(x_0, y, \text{add}(x_0, y)) = \text{succ}(\text{add}(x_0, y)) \end{aligned}$$

So add is primitive recursive provided f and g are as well. $f(x_0) = x_0 = P_0^1(x_0)$, and the projection functions count as primitive recursive, so f is primitive recursive. The function g is the three-place function $g(x_0, y, z)$ defined by

$$g(x_0, y, z) = \text{succ}(z).$$

This does not yet tell us that g is primitive recursive, since g and succ are not quite the same function: succ is one-place, and g has to be three-place. But we can define g “officially” by composition as

$$g(x_0, y, z) = \text{succ}(P_2^3(x_0, y, z))$$

Since succ and P_2^3 count as primitive recursive functions, g does as well, since it can be defined by composition from primitive recursive functions. $\square$

Proposição 29.5. *The multiplication function $\text{mult}(x, y) = x \cdot y$ is primitive recursive.*

*cmp:rec:prf:
prop:mult-pr*

Prova. Exercise. $\square$

Problema 29.1. Prove [Proposição 29.5](#) by showing that the primitive recursive definition of mult can be put into the form required by [Definição 29.1](#) and showing that the corresponding functions f and g are primitive recursive.

Exemplo 29.6. Here’s our very first example of a primitive recursive definition:

$$\begin{aligned} h(0) &= 1 \\ h(y + 1) &= 2 \cdot h(y). \end{aligned}$$

29.5. PRIMITIVE RECURSION NOTATIONS

This function cannot fit into the form required by [Definição 29.1](#), since $k = 0$. The definition also involves the constants 1 and 2. To get around the first problem, let's introduce a dummy argument and define the function h' :

$$\begin{aligned} h'(x_0, 0) &= f(x_0) = 1 \\ h'(x_0, y + 1) &= g(x_0, y, h'(x_0, y)) = 2 \cdot h'(x_0, y). \end{aligned}$$

The function $f(x_0) = 1$ can be defined from `succ` and `zero` by composition: $f(x_0) = \text{succ}(\text{zero}(x_0))$. The function g can be defined by composition from $g'(z) = 2 \cdot z$ and projections:

$$g(x_0, y, z) = g'(P_2^3(x_0, y, z))$$

and g' in turn can be defined by composition as

$$g'(z) = \text{mult}(g''(z), P_0^1(z))$$

and

$$g''(z) = \text{succ}(f(z)),$$

where f is as above: $f(z) = \text{succ}(\text{zero}(z))$. Now that we have h' , we can use composition again to let $h(y) = h'(P_0^1(y), P_0^1(y))$. This shows that h can be defined from the basic functions using a sequence of compositions and primitive recursions, so h is primitive recursive.

-NoValue- [content/computability/recursive-functions/notation-pr-functions.tex](#)

29.5 Primitive Recursion Notations

cmp:rec:not:
sec One advantage to having the precise inductive description of the primitive recursive functions is that we can be systematic in describing them. For example, we can assign a “notation” to each such function, as follows. Use symbols `zero`, `succ`, and P_i^n for zero, successor, and the projections. Now suppose h is defined by composition from a k -place function f and n -place functions $g_0, \dots, g_{k-1}$, and we have assigned notations $F, G_0, \dots, G_{k-1}$ to the latter functions. Then, using a new symbol $\text{Comp}_{k,n}$, we can denote the function h by $\text{Comp}_{k,n}[F, G_0, \dots, G_{k-1}]$.

For functions defined by primitive recursion, we can use analogous notations. Suppose the $(k + 1)$ -ary function h is defined by primitive recursion from the k -ary function f and the $(k + 2)$ -ary function g , and the notations assigned to f and g are F and G , respectively. Then the notation assigned to h is $\text{Rec}_k[F, G]$.

Recall that the addition function is defined by primitive recursion as

$$\begin{aligned} \text{add}(x_0, 0) &= P_0^1(x_0) = x_0 \\ \text{add}(x_0, y + 1) &= \text{succ}(P_2^3(x_0, y, \text{add}(x_0, y))) = \text{add}(x_0, y) + 1 \end{aligned}$$

Here the role of f is played by P_0^1 , and the role of g is played by $\text{succ}(P_2^3(x_0, y, z))$, which is assigned the notation $\text{Comp}_{1,3}[\text{succ}, P_2^3]$ as it is the result of defining a function by composition from the 1-ary function succ and the 3-ary function P_2^3 . With this setup, we can denote the addition function by

$$\text{Rec}_1[P_0^1, \text{Comp}_{1,3}[\text{succ}, P_2^3]].$$

Having these notations sometimes proves useful, p.ex., when enumerating primitive recursive functions.

Problema 29.2. Give the complete primitive recursive notation for mult .

-NoValue- _____
`content/computability/recursive-functions/pr-functions-computable.tex`

29.6 Primitive Recursive Functions are Computable

Suppose a function h is defined by primitive recursion

`cmp:rec:cmp:`
`sec`

$$\begin{aligned} h(\vec{x}, 0) &= f(\vec{x}) \\ h(\vec{x}, y + 1) &= g(\vec{x}, y, h(\vec{x}, y)) \end{aligned}$$

and suppose the functions f and g are computable. (We use $\vec{x}$ to abbreviate $x_0, \dots, x_{k-1}$.) Then $h(\vec{x}, 0)$ can obviously be computed, since it is just $f(\vec{x})$ which we assume is computable. $h(\vec{x}, 1)$ can then also be computed, since $1 = 0 + 1$ and so $h(\vec{x}, 1)$ is just

$$h(\vec{x}, 1) = g(\vec{x}, 0, h(\vec{x}, 0)) = g(\vec{x}, 0, f(\vec{x})).$$

We can go on in this way and compute

$$\begin{aligned} h(\vec{x}, 2) &= g(\vec{x}, 1, h(\vec{x}, 1)) = g(\vec{x}, 1, g(\vec{x}, 0, f(\vec{x}))) \\ h(\vec{x}, 3) &= g(\vec{x}, 2, h(\vec{x}, 2)) = g(\vec{x}, 2, g(\vec{x}, 1, g(\vec{x}, 0, f(\vec{x})))) \\ h(\vec{x}, 4) &= g(\vec{x}, 3, h(\vec{x}, 3)) = g(\vec{x}, 3, g(\vec{x}, 2, g(\vec{x}, 1, g(\vec{x}, 0, f(\vec{x})))) \\ &\vdots \end{aligned}$$

Thus, to compute $h(\vec{x}, y)$ in general, successively compute $h(\vec{x}, 0)$, $h(\vec{x}, 1)$, $\dots$, until we reach $h(\vec{x}, y)$.

Thus, a primitive recursive definition yields a new computable function if the functions f and g are computable. Composition of functions also results in a computable function if the functions f and g_i are computable.

Since the basic functions zero , succ , and P_i^n are computable, and composition and primitive recursion yield computable functions from computable functions, this means that every primitive recursive function is computable.

-NoValue- _____
`content/computability/recursive-functions/examples.tex`

29.7 Examples of Primitive Recursive Functions

cmp:rec:exa:
sec

We already have some examples of primitive recursive functions: the addition and multiplication functions `add` and `mult`. The identity function $\text{id}(x) = x$ is primitive recursive, since it is just P_0^1 . The constant functions $\text{const}_n(x) = n$ are primitive recursive since they can be defined from zero and `succ` by successive composition. This is useful when we want to use constants in primitive recursive definitions, p.ex., if we want to define the function $f(x) = 2 \cdot x$ can obtain it by composition from $\text{const}_2(x)$ and multiplication as $f(x) = \text{mult}(\text{const}_2(x), P_0^1(x))$. We'll make use of this trick from now on.

Proposição 29.7. *The exponentiation function $\exp(x, y) = x^y$ is primitive recursive.*

Prova. We can define `exp` primitive recursively as

$$\begin{aligned}\exp(x, 0) &= 1 \\ \exp(x, y + 1) &= \text{mult}(x, \exp(x, y)).\end{aligned}$$

Strictly speaking, this is not a recursive definition from primitive recursive functions. Officially, though, we have:

$$\begin{aligned}\exp(x, 0) &= f(x) \\ \exp(x, y + 1) &= g(x, y, \exp(x, y)).\end{aligned}$$

where

$$\begin{aligned}f(x) &= \text{succ}(\text{zero}(x)) = 1 \\ g(x, y, z) &= \text{mult}(P_0^3(x, y, z), P_2^3(x, y, z)) = x \cdot z\end{aligned}$$

and so f and g are defined from primitive recursive functions by composition. $\square$

Proposição 29.8. *The predecessor function $\text{pred}(y)$ defined by*

$$\text{pred}(y) = \begin{cases} 0 & \text{if } y = 0 \\ y - 1 & \text{otherwise} \end{cases}$$

is primitive recursive.

Prova. Note that

$$\begin{aligned}\text{pred}(0) &= 0 \text{ and} \\ \text{pred}(y + 1) &= y.\end{aligned}$$

This is almost a primitive recursive definition. It does not, strictly speaking, fit into the pattern of definition by primitive recursion, since that pattern requires

at least one extra argument x . It is also odd in that it does not actually use $\text{pred}(y)$ in the definition of $\text{pred}(y+1)$. But we can first define $\text{pred}'(x, y)$ by

$$\begin{aligned}\text{pred}'(x, 0) &= \text{zero}(x) = 0, \\ \text{pred}'(x, y+1) &= P_1^3(x, y, \text{pred}'(x, y)) = y.\end{aligned}$$

and then define pred from it by composition, p.ex., as $\text{pred}(x) = \text{pred}'(\text{zero}(x), P_0^1(x))$. $\square$

Proposição 29.9. *The factorial function $\text{fac}(x) = x! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot x$ is primitive recursive.*

Prova. The obvious primitive recursive definition is

$$\begin{aligned}\text{fac}(0) &= 1 \\ \text{fac}(y+1) &= \text{fac}(y) \cdot (y+1).\end{aligned}$$

Officially, we have to first define a two-place function h

$$\begin{aligned}h(x, 0) &= \text{const}_1(x) \\ h(x, y+1) &= g(x, y, h(x, y))\end{aligned}$$

where $g(x, y, z) = \text{mult}(P_2^3(x, y, z), \text{succ}(P_1^3(x, y, z)))$ and then let

$$\text{fac}(y) = h(P_0^1(y), P_0^1(y)) = h(y, y).$$

From now on we'll be a bit more laissez-faire and not give the official definitions by composition and primitive recursion. $\square$

Proposição 29.10. *Truncated subtraction, $x \dot{-} y$, defined by*

$$x \dot{-} y = \begin{cases} 0 & \text{if } x < y \\ x - y & \text{otherwise} \end{cases}$$

is primitive recursive.

Prova. We have:

$$\begin{aligned}x \dot{-} 0 &= x \\ x \dot{-} (y+1) &= \text{pred}(x \dot{-} y)\end{aligned} \quad \square$$

Proposição 29.11. *The distance between x and y , $|x - y|$, is primitive recursive.*

Prova. We have $|x - y| = (x \dot{-} y) + (y \dot{-} x)$, so the distance can be defined by composition from $+$ and $\dot{-}$, which are primitive recursive. $\square$

29.7. EXAMPLES OF PRIMITIVE RECURSIVE FUNCTIONS

Proposição 29.12. *The maximum of x and y , $\max(x, y)$, is primitive recursive.*

Prova. We can define $\max(x, y)$ by composition from $+$ and $\dot{-}$ by

$$\max(x, y) = x + (y \dot{-} x).$$

If x is the maximum, O.s., $x \geq y$, then $y \dot{-} x = 0$, so $x + (y \dot{-} x) = x + 0 = x$. If y is the maximum, then $y \dot{-} x = y - x$, and so $x + (y \dot{-} x) = x + (y - x) = y$. $\square$

cmp:rec:exa:
prop:min-pr **Proposição 29.13.** *The minimum of x and y , $\min(x, y)$, is primitive recursive.*

Prova. Exercise. $\square$

Problema 29.3. Prove **Proposição 29.13**.

Problema 29.4. Show that

$$f(x, y) = 2^{\left(2^{\left(\dots^{2^x}\right)}\right)} y \text{ 's}$$

is primitive recursive.

Problema 29.5. Show that integer division $d(x, y) = \lfloor x/y \rfloor$ (O.s., division, where you disregard everything after the decimal point) is primitive recursive. When $y = 0$, we stipulate $d(x, y) = 0$. Give an explicit definition of d using primitive recursion and composition.

Proposição 29.14. *The set of primitive recursive functions is closed under the following two operations:*

1. *Finite sums: if $f(\vec{x}, z)$ is primitive recursive, then so is the function*

$$g(\vec{x}, y) = \sum_{z=0}^y f(\vec{x}, z).$$

2. *Finite products: if $f(\vec{x}, z)$ is primitive recursive, then so is the function*

$$h(\vec{x}, y) = \prod_{z=0}^y f(\vec{x}, z).$$

Prova. For example, finite sums are defined recursively by the equations

$$\begin{aligned} g(\vec{x}, 0) &= f(\vec{x}, 0) \\ g(\vec{x}, y + 1) &= g(\vec{x}, y) + f(\vec{x}, y + 1). \end{aligned}$$

$\square$

-NoValue-

content/computability/recursive-functions/pr-relations.tex

29.8 Primitive Recursive Relations

Definição 29.15. A relation $R(\vec{x})$ is said to be primitive recursive if its characteristic function,

$$\chi_R(\vec{x}) = \begin{cases} 1 & \text{if } R(\vec{x}) \\ 0 & \text{otherwise} \end{cases}$$

is primitive recursive.

In other words, when one speaks of a primitive recursive relation $R(\vec{x})$, one is referring to a relation of the form $\chi_R(\vec{x}) = 1$, where χ_R is a primitive recursive function which, on any input, returns either 1 or 0. For example, the relation $\text{IsZero}(x)$, which holds if and only if $x = 0$, corresponds to the function χ_{IsZero} , defined using primitive recursion by

$$\begin{aligned} \chi_{\text{IsZero}}(0) &= 1, \\ \chi_{\text{IsZero}}(x+1) &= 0. \end{aligned}$$

It should be clear that one can compose relations with other primitive recursive functions. So the following are also primitive recursive:

1. The equality relation, $x = y$, defined by $\text{IsZero}(|x - y|)$
2. The less-than relation, $x \leq y$, defined by $\text{IsZero}(x \dot{-} y)$

Proposição 29.16. *The set of primitive recursive relations is closed under Boolean operations, that is, if $P(\vec{x})$ and $Q(\vec{x})$ are primitive recursive, so are*

1. $\neg P(\vec{x})$
2. $P(\vec{x}) \wedge Q(\vec{x})$
3. $P(\vec{x}) \vee Q(\vec{x})$
4. $P(\vec{x}) \rightarrow Q(\vec{x})$

Prova. Suppose $P(\vec{x})$ and $Q(\vec{x})$ are primitive recursive, O.s., their characteristic functions χ_P and χ_Q are. We have to show that the characteristic functions of $\neg P(\vec{x})$, etc., are also primitive recursive.

$$\chi_{\neg P}(\vec{x}) = \begin{cases} 0 & \text{if } \chi_P(\vec{x}) = 1 \\ 1 & \text{otherwise} \end{cases}$$

We can define $\chi_{\neg P}(\vec{x})$ as $1 \dot{-} \chi_P(\vec{x})$.

$$\chi_{P \wedge Q}(\vec{x}) = \begin{cases} 1 & \text{if } \chi_P(\vec{x}) = \chi_Q(\vec{x}) = 1 \\ 0 & \text{otherwise} \end{cases}$$

We can define $\chi_{P \wedge Q}(\vec{x})$ as $\chi_P(\vec{x}) \cdot \chi_Q(\vec{x})$ or as $\min(\chi_P(\vec{x}), \chi_Q(\vec{x}))$. Similarly,

$$\begin{aligned} \chi_{P \vee Q}(\vec{x}) &= \max(\chi_P(\vec{x}), \chi_Q(\vec{x})) \text{ and} \\ \chi_{P \rightarrow Q}(\vec{x}) &= \max(1 \dot{-} \chi_P(\vec{x}), \chi_Q(\vec{x})). \end{aligned}$$

□

29.8. PRIMITIVE RECURSIVE RELATIONS

Proposição 29.17. *The set of primitive recursive relations is closed under bounded quantification, O.s., if $R(\vec{x}, z)$ is a primitive recursive relation, then so are the relations*

$$(\forall z < y) R(\vec{x}, z) \text{ and } (\exists z < y) R(\vec{x}, z).$$

$(\forall z < y) R(\vec{x}, z)$ holds of $\vec{x}$ and y if and only if $R(\vec{x}, z)$ holds for every z less than y , and similarly for $(\exists z < y) R(\vec{x}, z)$.

Prova. By convention, we take $(\forall z < 0) R(\vec{x}, z)$ to be true (for the trivial reason that there are no z less than 0) and $(\exists z < 0) R(\vec{x}, z)$ to be false. A bounded universal quantifier functions just like a finite product or iterated minimum, O.s., if $P(\vec{x}, y) \Leftrightarrow (\forall z < y) R(\vec{x}, z)$ then $\chi_P(\vec{x}, y)$ can be defined by

$$\begin{aligned} \chi_P(\vec{x}, 0) &= 1 \\ \chi_P(\vec{x}, y+1) &= \min(\chi_P(\vec{x}, y), \chi_R(\vec{x}, y)). \end{aligned}$$

Bounded existential quantification can similarly be defined using max. Alternatively, it can be defined from bounded universal quantification, using the equivalence $(\exists z < y) R(\vec{x}, z) \Leftrightarrow \neg(\forall z < y) \neg R(\vec{x}, z)$. Note that, for example, a bounded quantifier of the form $(\exists x \leq y) \dots x \dots$ is equivalent to $(\exists x < y+1) \dots x \dots$. $\square$

Problema 29.6. Show that the three place relation $x \equiv y \pmod n$ (congruence modulo n) is primitive recursive.

Another useful primitive recursive function is the conditional function, $\text{cond}(x, y, z)$, defined by

$$\text{cond}(x, y, z) = \begin{cases} y & \text{if } x = 0 \\ z & \text{otherwise.} \end{cases}$$

This is defined recursively by

$$\begin{aligned} \text{cond}(0, y, z) &= y, \\ \text{cond}(x+1, y, z) &= z. \end{aligned}$$

One can use this to justify definitions of primitive recursive functions by cases from primitive recursive relations:

Proposição 29.18. *If $g_0(\vec{x}), \dots, g_m(\vec{x})$ are primitive recursive functions, and $R_0(\vec{x}), \dots, R_{m-1}(\vec{x})$ are primitive recursive relations, then the function f defined by*

$$f(\vec{x}) = \begin{cases} g_0(\vec{x}) & \text{if } R_0(\vec{x}) \\ g_1(\vec{x}) & \text{if } R_1(\vec{x}) \text{ and not } R_0(\vec{x}) \\ \vdots & \\ g_{m-1}(\vec{x}) & \text{if } R_{m-1}(\vec{x}) \text{ and none of the previous hold} \\ g_m(\vec{x}) & \text{otherwise} \end{cases}$$

is also primitive recursive.

Prova. When $m = 1$, this is just the function defined by

$$f(\vec{x}) = \text{cond}(\chi_{\neg R_0}(\vec{x}), g_0(\vec{x}), g_1(\vec{x})).$$

For m greater than 1, one can just compose definitions of this form. $\square$

-NoValue-

content/computability/recursive-functions/bounded-minimization.tex

29.9 Bounded Minimization

explanation

It is often useful to define a function as the least number satisfying some property or relation P . If P is decidable, we can compute this function simply by trying out all the possible numbers, 0, 1, 2, ..., until we find the least one satisfying P . This kind of unbounded search takes us out of the realm of primitive recursive functions. However, if we're only interested in the least number *less than some independently given bound*, we stay primitive recursive. In other words, and a bit more generally, suppose we have a primitive recursive relation $R(x, z)$. Consider the function that maps x and y to the least $z < y$ such that $R(x, z)$. It, too, can be computed, by testing whether $R(x, 0)$, $R(x, 1)$, ..., $R(x, y - 1)$. But why is it primitive recursive?

cmp:rec:bmi:
sec

Proposição 29.19. *If $R(\vec{x}, z)$ is primitive recursive, so is the function $m_R(\vec{x}, y)$ which returns the least z less than y such that $R(\vec{x}, z)$ holds, if there is one, and y otherwise. We will write the function m_R as*

$$(\min z < y) R(\vec{x}, z),$$

Prova. Note that there can be no $z < 0$ such that $R(\vec{x}, z)$ since there is no $z < 0$ at all. So $m_R(\vec{x}, 0) = 0$.

In case the bound is of the form $y + 1$ we have three cases:

1. There is a $z < y$ such that $R(\vec{x}, z)$, in which case $m_R(\vec{x}, y + 1) = m_R(\vec{x}, y)$.
2. There is no such $z < y$ but $R(\vec{x}, y)$ holds, then $m_R(\vec{x}, y + 1) = y$.
3. There is no $z < y + 1$ such that $R(\vec{x}, z)$, then $m_R(\vec{x}, y + 1) = y + 1$.

So we can define $m_R(\vec{x}, 0)$ by primitive recursion as follows:

$$m_R(\vec{x}, 0) = 0$$

$$m_R(\vec{x}, y + 1) = \begin{cases} m_R(\vec{x}, y) & \text{if } m_R(\vec{x}, y) \neq y \\ y & \text{if } m_R(\vec{x}, y) = y \text{ and } R(\vec{x}, y) \\ y + 1 & \text{otherwise.} \end{cases}$$

Note that there is a $z < y$ such that $R(\vec{x}, z)$ iff $m_R(\vec{x}, y) \neq y$. $\square$

Problem 29.7. Suppose $R(\vec{x}, z)$ is primitive recursive. Define the function $m'_R(\vec{x}, y)$ which returns the least z less than y such that $R(\vec{x}, z)$ holds, if there is one, and 0 otherwise, by primitive recursion from χ_R .

-NoValue-

content/computability/recursive-functions/primes.tex

29.10 Primes

cmp:rec:pri:
sec

Bounded quantification and bounded minimization provide us with a good deal of machinery to show that natural functions and relations are primitive recursive. For example, consider the relation “ x divides y ”, written $x \mid y$. The relation $x \mid y$ holds if division of y by x is possible without remainder, O.s., if y is an integer multiple of x . (If it doesn’t hold, O.s., the remainder when dividing x by y is > 0 , we write $x \nmid y$.) In other words, $x \mid y$ iff for some z , $x \cdot z = y$. Obviously, any such z , if it exists, must be $\leq y$. So, we have that $x \mid y$ iff for some $z \leq y$, $x \cdot z = y$. We can define the relation $x \mid y$ by bounded existential quantification from $=$ and multiplication by

$$x \mid y \Leftrightarrow (\exists z \leq y) (x \cdot z) = y.$$

We’ve thus shown that $x \mid y$ is primitive recursive.

A natural number x is *prime* if it is neither 0 nor 1 and is only divisible by 1 and itself. In other words, prime numbers are such that, whenever $y \mid x$, either $y = 1$ or $y = x$. To test if x is prime, we only have to check if $y \mid x$ for all $y \leq x$, since if $y > x$, then automatically $y \nmid x$. So, the relation $\text{Prime}(x)$, which holds iff x is prime, can be defined by

$$\text{Prime}(x) \Leftrightarrow x \geq 2 \wedge (\forall y \leq x) (y \mid x \rightarrow y = 1 \vee y = x)$$

and is thus primitive recursive.

The primes are 2, 3, 5, 7, 11, etc. Consider the function $p(x)$ which returns the x th prime in that sequence, O.s., $p(0) = 2$, $p(1) = 3$, $p(2) = 5$, etc. (For convenience we will often write $p(x)$ as p_x ($p_0 = 2$, $p_1 = 3$, etc.))

If we had a function $\text{nextPrime}(x)$, which returns the first prime number larger than x , p can be easily defined using primitive recursion:

$$\begin{aligned} p(0) &= 2 \\ p(x+1) &= \text{nextPrime}(p(x)) \end{aligned}$$

Since $\text{nextPrime}(x)$ is the least y such that $y > x$ and y is prime, it can be easily computed by unbounded search. But it can also be defined by bounded minimization, thanks to a result due to Euclid: there is always a prime number between x and $x! + 1$.

$$\text{nextPrime}(x) = (\min y \leq x! + 1) (y > x \wedge \text{Prime}(y)).$$

This shows, that $\text{nextPrime}(x)$ and hence $p(x)$ are (not just computable but) primitive recursive.

(If you're curious, here's a quick proof of Euclid's theorem. Suppose p_n is the largest prime $\leq x$ and consider the product $p = p_0 \cdot p_1 \cdot \dots \cdot p_n$ of all primes $\leq x$. Either $p + 1$ is prime or there is a prime between x and $p + 1$. Why? Suppose $p + 1$ is not prime. Then some prime number $q \mid p + 1$ where $q < p + 1$. None of the primes $\leq x$ divide $p + 1$. (By definition of p , each of the primes $p_i \leq x$ divides p , O.s., with remainder 0. So, each of the primes $p_i \leq x$ divides $p + 1$ with remainder 1, and so $p_i \nmid p + 1$.) Hence, q is a prime $> x$ and $< p + 1$. And $p \leq x!$, so there is a prime $> x$ and $\leq x! + 1$.)

Problema 29.8. Define integer division $d(x, y)$ using bounded minimization.

-NoValue-
[content/computability/recursive-functions/sequences.tex](#)

29.11 Sequences

The set of primitive recursive functions is remarkably robust. But we will be able to do even more once we have developed a adequate means of handling *sequences*. We will identify finite sequences of natural numbers with natural numbers in the following way: the sequence $\langle a_0, a_1, a_2, \dots, a_k \rangle$ corresponds to the number

$$p_0^{a_0+1} \cdot p_1^{a_1+1} \cdot p_2^{a_2+1} \cdot \dots \cdot p_k^{a_k+1}.$$

We add one to the exponents to guarantee that, for example, the sequences $\langle 2, 7, 3 \rangle$ and $\langle 2, 7, 3, 0, 0 \rangle$ have distinct numeric codes. We can take both 0 and 1 to code the empty sequence; for concreteness, let Λ denote 0.

The reason that this coding of sequences works is the so-called Fundamental Theorem of Arithmetic: every natural number $n \geq 2$ can be written in one and only one way in the form

$$n = p_0^{a_0} \cdot p_1^{a_1} \cdot \dots \cdot p_k^{a_k}$$

with $a_k \geq 1$. This guarantees that the mapping $\langle \rangle(a_0, \dots, a_k) = \langle a_0, \dots, a_k \rangle$ is injective: different sequences are mapped to different numbers; to each number only at most one sequence corresponds.

We'll now show that the operations of determining the length of a sequence, determining its i th element, appending an element to a sequence, and concatenating two sequences, are all primitive recursive.

Proposição 29.20. *The function $\text{len}(s)$, which returns the length of the sequence s , is primitive recursive.*

Prova. Let $R(i, s)$ be the relation defined by

$$R(i, s) \text{ iff } p_i \mid s \wedge p_{i+1} \nmid s.$$

29.11. SEQUENCES

R is clearly primitive recursive. Whenever s is the code of a non-empty sequence, O.s.,

$$s = p_0^{a_0+1} \cdot \dots \cdot p_k^{a_k+1},$$

$R(i, s)$ holds if p_i is the largest prime such that $p_i \mid s$, O.s., $i = k$. The length of s thus is $i + 1$ iff p_i is the largest prime that divides s , so we can let

$$\text{len}(s) = \begin{cases} 0 & \text{if } s = 0 \text{ or } s = 1 \\ 1 + (\min i < s) R(i, s) & \text{otherwise} \end{cases}$$

We can use bounded minimization here, since there is only one i that satisfies $R(i, s)$ when s is a code of a sequence, and if i exists it is less than s itself. $\square$

Proposição 29.21. *The function $\text{append}(s, a)$, which returns the result of appending a to the sequence s , is primitive recursive.*

Prova. append can be defined by:

$$\text{append}(s, a) = \begin{cases} 2^{a+1} & \text{if } s = 0 \text{ or } s = 1 \\ s \cdot p_{\text{len}(s)}^{a+1} & \text{otherwise.} \end{cases} \quad \square$$

Proposição 29.22. *The function $\text{element}(s, i)$, which returns the i th element of s (where the initial element is called the 0th), or 0 if i is greater than or equal to the length of s , is primitive recursive.*

Prova. Note that a is the i th element of s iff p_i^{a+1} is the largest power of p_i that divides s , O.s., $p_i^{a+1} \mid s$ but $p_i^{a+2} \nmid s$. So:

$$\text{element}(s, i) = \begin{cases} 0 & \text{if } i \geq \text{len}(s) \\ (\min a < s) (p_i^{a+2} \nmid s) & \text{otherwise.} \end{cases} \quad \square$$

Instead of using the official names for the functions defined above, we introduce a more compact notation. We will use $(s)_i$ instead of $\text{element}(s, i)$, and $\langle s_0, \dots, s_k \rangle$ to abbreviate

$$\text{append}(\text{append}(\dots \text{append}(A, s_0) \dots), s_k).$$

Note that if s has length k , the elements of s are $(s)_0, \dots, (s)_{k-1}$.

Proposição 29.23. *The function $\text{concat}(s, t)$, which concatenates two sequences, is primitive recursive.*

Prova. We want a function concat with the property that

$$\text{concat}(\langle a_0, \dots, a_k \rangle, \langle b_0, \dots, b_l \rangle) = \langle a_0, \dots, a_k, b_0, \dots, b_l \rangle.$$

We'll use a “helper” function $\text{hconcat}(s, t, n)$ which concatenates the first n symbols of t to s . This function can be defined by primitive recursion as follows:

$$\begin{aligned}\text{hconcat}(s, t, 0) &= s \\ \text{hconcat}(s, t, n+1) &= \text{append}(\text{hconcat}(s, t, n), (t)_n)\end{aligned}$$

Then we can define concat by

$$\text{concat}(s, t) = \text{hconcat}(s, t, \text{len}(t)). \quad \square$$

We will write $s \frown t$ instead of $\text{concat}(s, t)$.

It will be useful for us to be able to bound the numeric code of a sequence in terms of its length and its largest element. Suppose s is a sequence of length k , each element of which is less than or equal to some number x . Then s has at most k prime factors, each at most p_{k-1} , and each raised to at most $x+1$ in the prime factorization of s . In other words, if we define

$$\text{sequenceBound}(x, k) = p_{k-1}^{k \cdot (x+1)},$$

then the numeric code of the sequence s described above is at most $\text{sequenceBound}(x, k)$.

Having such a bound on sequences gives us a way of defining new functions using bounded search. For example, we can define concat using bounded search. All we need to do is write down a primitive recursive *specification* of the object (number of the concatenated sequence) we are looking for, and a bound on how far to look. The following works:

$$\begin{aligned}\text{concat}(s, t) &= (\min v < \text{sequenceBound}(s+t, \text{len}(s) + \text{len}(t))) \\ &\quad (\text{len}(v) = \text{len}(s) + \text{len}(t) \wedge \\ &\quad (\forall i < \text{len}(s)) ((v)_i = (s)_i) \wedge \\ &\quad (\forall j < \text{len}(t)) ((v)_{\text{len}(s)+j} = (t)_j))\end{aligned}$$

Problema 29.9. Show that there is a primitive recursive function $\text{sconcat}(s)$ with the property that

$$\text{sconcat}(\langle s_0, \dots, s_k \rangle) = s_0 \frown \dots \frown s_k.$$

Problema 29.10. Show that there is a primitive recursive function $\text{tail}(s)$ with the property that

$$\begin{aligned}\text{tail}(\Lambda) &= 0 \text{ and} \\ \text{tail}(\langle s_0, \dots, s_k \rangle) &= \langle s_1, \dots, s_k \rangle.\end{aligned}$$

Proposição 29.24. *The function $\text{subseq}(s, i, n)$ which returns the subsequence of s of length n beginning at the i th element, is primitive recursive.* [cmp:rec:seq](#)
[prop:subseq](#)

Prova. Exercise. $\square$

Problema 29.11. Prove [Proposição 29.24](#).

-NoValue-

[content/computability/recursive-functions/trees.tex](#)

29.12 Trees

[cmp:rec:tre:](#)
[sec](#)

Sometimes it is useful to represent trees as natural numbers, just like we can represent sequences by numbers and properties of and operations on them by primitive recursive relations and functions on their codes. We'll use sequences and their codes to do this. A tree can be either a single node (possibly with a label) or else a node (possibly with a label) connected to a number of subtrees. The node is called the *root* of the tree, and the subtrees it is connected to its *immediate subtrees*.

We code trees recursively as a sequence $\langle k, d_1, \dots, d_k \rangle$, where k is the number of immediate subtrees and $d_1, \dots, d_k$ the codes of the immediate subtrees. If the nodes have labels, they can be included after the immediate subtrees. So a tree consisting just of a single node with label l would be coded by $\langle 0, l \rangle$, and a tree consisting of a root (labelled l_1) connected to two single nodes (labelled l_2, l_3) would be coded by $\langle 2, \langle 0, l_2 \rangle, \langle 0, l_3 \rangle, l_1 \rangle$.

[cmp:rec:tre:](#)
[prop:subtreeseq](#)

Proposição 29.25. *The function $\text{SubtreeSeq}(t)$, which returns the code of a sequence the elements of which are the codes of all subtrees of the tree with code t , is primitive recursive.*

Prova. First note that $\text{ISubtrees}(t) = \text{subseq}(t, 1, (t)_0)$ is primitive recursive and returns the codes of the immediate subtrees of a tree t . Now we can define a helper function $\text{hSubtreeSeq}(t, n)$ which computes the sequence of all subtrees which are n nodes removed from the root. The sequence of subtrees of t which is 0 nodes removed from the root—in other words, begins at the root of t —is the sequence consisting just of t . To obtain a sequence of all level $n + 1$ subtrees of t , we concatenate the level n subtrees with a sequence consisting of all immediate subtrees of the level n subtrees. To get a list of all these, note that if $f(x)$ is a primitive recursive function returning codes of sequences, then $g_f(s, k) = f((s)_0) \frown \dots \frown f((s)_k)$ is also primitive recursive:

$$\begin{aligned} g(s, 0) &= f((s)_0) \\ g(s, k + 1) &= g(s, k) \frown f((s)_{k+1}) \end{aligned}$$

For instance, if s is a sequence of trees, then $h(s) = g_{\text{ISubtrees}}(s, \text{len}(s))$ gives the sequence of the immediate subtrees of the elements of s . We can use it to define hSubtreeSeq by

$$\begin{aligned} \text{hSubtreeSeq}(t, 0) &= \langle t \rangle \\ \text{hSubtreeSeq}(t, n + 1) &= \text{hSubtreeSeq}(t, n) \frown h(\text{hSubtreeSeq}(t, n)). \end{aligned}$$

The maximum level of subtrees in a tree coded by t , O.s., the maximum distance between the root and a leaf node, is bounded by the code t . So a sequence of codes of all subtrees of the tree coded by t is given by $\text{hSubtreeSeq}(t, t)$. $\square$

Problema 29.12. The definition of hSubtreeSeq in the proof of [Proposição 29.25](#) in general includes repetitions. Give an alternative definition which guarantees that the code of a subtree occurs only once in the resulting list.

-NoValue-

[content/computability/recursive-functions/other-recursions.tex](#)

29.13 Other Recursions

Using pairing and sequencing, we can justify more exotic (and useful) forms of primitive recursion. For example, it is often useful to define two functions simultaneously, such as in the following definition: cmp:rec:ore:
sec

$$\begin{aligned} h_0(\vec{x}, 0) &= f_0(\vec{x}) \\ h_1(\vec{x}, 0) &= f_1(\vec{x}) \\ h_0(\vec{x}, y + 1) &= g_0(\vec{x}, y, h_0(\vec{x}, y), h_1(\vec{x}, y)) \\ h_1(\vec{x}, y + 1) &= g_1(\vec{x}, y, h_0(\vec{x}, y), h_1(\vec{x}, y)) \end{aligned}$$

This is an instance of *simultaneous recursion*. Another useful way of defining functions is to give the value of $h(\vec{x}, y + 1)$ in terms of *all* the values $h(\vec{x}, 0), \dots, h(\vec{x}, y)$, as in the following definition:

$$\begin{aligned} h(\vec{x}, 0) &= f(\vec{x}) \\ h(\vec{x}, y + 1) &= g(\vec{x}, y, \langle h(\vec{x}, 0), \dots, h(\vec{x}, y) \rangle). \end{aligned}$$

The following schema captures this idea more succinctly:

$$h(\vec{x}, y) = g(\vec{x}, y, \langle h(\vec{x}, 0), \dots, h(\vec{x}, y - 1) \rangle)$$

with the understanding that the last argument to g is just the empty sequence when y is 0. In either formulation, the idea is that in computing the “successor step,” the function h can make use of the entire sequence of values computed so far. This is known as a *course-of-values* recursion. For a particular example, it can be used to justify the following type of definition:

$$h(\vec{x}, y) = \begin{cases} g(\vec{x}, y, h(\vec{x}, k(\vec{x}, y))) & \text{if } k(\vec{x}, y) < y \\ f(\vec{x}) & \text{otherwise} \end{cases}$$

In other words, the value of h at y can be computed in terms of the value of h at *any* previous value, given by k .

29.14. NON-PRIMITIVE RECURSIVE FUNCTIONS

Problema 29.13. Define the remainder function $r(x, y)$ by course-of-values recursion. (If x, y are natural numbers and $y > 0$, $r(x, y)$ is the number less than y such that $x = z \times y + r(x, y)$ for some z . For definiteness, let's say that if $y = 0$, $r(x, 0) = 0$.)

You should think about how to obtain these functions using ordinary primitive recursion. One final version of primitive recursion is more flexible in that one is allowed to change the *parameters* (side values) along the way:

$$\begin{aligned} h(\vec{x}, 0) &= f(\vec{x}) \\ h(\vec{x}, y + 1) &= g(\vec{x}, y, h(k(\vec{x}), y)) \end{aligned}$$

This, too, can be simulated with ordinary primitive recursion. (Doing so is tricky. For a hint, try unwinding the computation by hand.)

-NoValue- _____
 content/computability/recursive-functions/non-pr-functions.tex

29.14 Non-Primitive Recursive Functions

cmp:rec:npr:
 sec

The primitive recursive functions do not exhaust the intuitively computable functions. It should be intuitively clear that we can make a list of all the unary primitive recursive functions, $f_0, f_1, f_2, \dots$ such that we can effectively compute the value of f_x on input y ; in other words, the function $g(x, y)$, defined by

$$g(x, y) = f_x(y)$$

is computable. But then so is the function

$$\begin{aligned} h(x) &= g(x, x) + 1 \\ &= f_x(x) + 1. \end{aligned}$$

For each primitive recursive function f_i , the value of h and f_i differ at i . So h is computable, but not primitive recursive; and one can say the same about g . This is an “effective” version of Cantor’s diagonalization argument.

One can provide more explicit examples of computable functions that are not primitive recursive. For example, let the notation $g^n(x)$ denote $g(g(\dots g(x)))$, with n g ’s in all; and define a sequence $g_0, g_1, \dots$ of functions by

$$\begin{aligned} g_0(x) &= x + 1 \\ g_{n+1}(x) &= g_n^x(x) \end{aligned}$$

You can confirm that each function g_n is primitive recursive. Each successive function grows much faster than the one before; $g_1(x)$ is equal to $2x$, $g_2(x)$ is equal to $2^x \cdot x$, and $g_3(x)$ grows roughly like an exponential stack of x 2’s. The Ackermann–Péter function is essentially the function $G(x) = g_x(x)$, and one can show that this grows faster than any primitive recursive function.

Let us return to the issue of enumerating the primitive recursive functions. Remember that we have assigned symbolic notations to each primitive recursive function; so it suffices to enumerate notations. We can assign a natural number $\#(F)$ to each notation F , recursively, as follows:

$$\begin{aligned}\#(0) &= \langle 0 \rangle \\ \#(S) &= \langle 1 \rangle \\ \#(P_i^n) &= \langle 2, n, i \rangle \\ \#(\text{Comp}_{k,l}[H, G_0, \dots, G_{k-1}]) &= \langle 3, k, l, \#(H), \#(G_0), \dots, \#(G_{k-1}) \rangle \\ \#(\text{Rec}_l[G, H]) &= \langle 4, l, \#(G), \#(H) \rangle\end{aligned}$$

Here we are using the fact that every sequence of numbers can be viewed as a natural number, using the codes from the last section. The upshot is that every code is assigned a natural number. Of course, some sequences (and hence some numbers) do not correspond to notations; but we can let f_i be the unary primitive recursive function with notation coded as i , if i codes such a notation; and the constant 0 function otherwise. The net result is that we have an explicit way of enumerating the unary primitive recursive functions.

(In fact, some functions, like the constant zero function, will appear more than once on the list. This is not just an artifact of our coding, but also a result of the fact that the constant zero function has more than one notation. We will later see that one can not computably avoid these repetitions; for example, there is no computable function that decides whether or not a given notation represents the constant zero function.)

We can now take the function $g(x, y)$ to be given by $f_x(y)$, where f_x refers to the enumeration we have just described. How do we know that $g(x, y)$ is computable? Intuitively, this is clear: to compute $g(x, y)$, first “unpack” x , and see if it is a notation for a unary function. If it is, compute the value of that function on input y .

digression

You may already be convinced that (with some work!) one can write a program (say, in Java or C++) that does this; and now we can appeal to the Church–Turing thesis, which says that anything that, intuitively, is computable can be computed by a Turing machine.

Of course, a more direct way to show that $g(x, y)$ is computable is to describe a Turing machine that computes it, explicitly. This would, in particular, avoid the Church–Turing thesis and appeals to intuition. Soon we will have built up enough machinery to show that $g(x, y)$ is computable, appealing to a model of computation that can be *simulated* on a Turing machine: namely, the recursive functions.

-NoValue-

[content/computability/recursive-functions/partial-functions.tex](#)

29.15 Partial Recursive Functions

cmp:rec:par:
sec

29.15. PARTIAL RECURSIVE FUNCTIONS

To motivate the definition of the recursive functions, note that our proof that there are computable functions that are not primitive recursive actually establishes much more. The argument was simple: all we used was the fact that it is possible to enumerate functions $f_0, f_1, \dots$ such that, as a function of x and y , $f_x(y)$ is computable. So the argument applies to *any class of functions that can be enumerated in such a way*. This puts us in a bind: we would like to describe the computable functions explicitly; but any explicit description of a collection of computable functions cannot be exhaustive!

The way out is to allow *partial* functions to come into play. We will see that it *is* possible to enumerate the partial computable functions. In fact, we already pretty much know that this is the case, since it is possible to enumerate Turing machines in a systematic way. We will come back to our diagonal argument later, and explore why it does not go through when partial functions are included.

The question is now this: what do we need to add to the primitive recursive functions to obtain all the partial recursive functions? We need to do two things:

1. Modify our definition of the primitive recursive functions to allow for partial functions as well.
2. Add something to the definition, so that some new partial functions are included.

The first is easy. As before, we will start with zero, successor, and projections, and close under composition and primitive recursion. The only difference is that we have to modify the definitions of composition and primitive recursion to allow for the possibility that some of the terms in the definition are not defined. If f and g are partial functions, we will write $f(x) \downarrow$ to mean that f is defined at x , O.s., x is in the domain of f ; and $f(x) \uparrow$ to mean the opposite, O.s., that f is not defined at x . We will use $f(x) \simeq g(x)$ to mean that either $f(x)$ and $g(x)$ are both undefined, or they are both defined and equal. We will use these notations for more complicated terms as well. We will adopt the convention that if h and $g_0, \dots, g_k$ all are partial functions, then

$$h(g_0(\vec{x}), \dots, g_k(\vec{x}))$$

is defined if and only if each g_i is defined at $\vec{x}$, and h is defined at $g_0(\vec{x}), \dots, g_k(\vec{x})$. With this understanding, the definitions of composition and primitive recursion for partial functions is just as above, except that we have to replace “=” by “ $\simeq$ ”.

What we will add to the definition of the primitive recursive functions to obtain partial functions is the *unbounded search operator*. If $f(x, \vec{z})$ is any partial function on the natural numbers, define $\mu x f(x, \vec{z})$ to be

the least x such that $f(0, \vec{z}), f(1, \vec{z}), \dots, f(x, \vec{z})$ are all defined, and $f(x, \vec{z}) = 0$, if such an x exists

with the understanding that $\mu x f(x, \vec{z})$ is undefined otherwise. This defines $\mu x f(x, \vec{z})$ uniquely.

explanation

Note that our definition makes no reference to Turing machines, or algorithms, or any specific computational model. But like composition and primitive recursion, there is an operational, computational intuition behind unbounded search. When it comes to the computability of a partial function, arguments where the function is undefined correspond to inputs for which the computation does not halt. The procedure for computing $\mu x f(x, \vec{z})$ will amount to this: compute $f(0, \vec{z})$, $f(1, \vec{z})$, $f(2, \vec{z})$ until a value of 0 is returned. If any of the intermediate computations do not halt, however, neither does the computation of $\mu x f(x, \vec{z})$.

If $R(x, \vec{z})$ is any relation, $\mu x R(x, \vec{z})$ is defined to be $\mu x (1 - \chi_R(x, \vec{z}))$. In other words, $\mu x R(x, \vec{z})$ returns the least value of x such that $R(x, \vec{z})$ holds. So, if $f(x, \vec{z})$ is a total function, $\mu x f(x, \vec{z})$ is the same as $\mu x (f(x, \vec{z}) = 0)$. But note that our original definition is more general, since it allows for the possibility that $f(x, \vec{z})$ is not everywhere defined (whereas, in contrast, the characteristic function of a relation is always total).

Definição 29.26. The set of *partial recursive functions* is the smallest set of partial functions from the natural numbers to the natural numbers (of various arities) containing zero, successor, and projections, and closed under composition, primitive recursion, and unbounded search.

Of course, some of the partial recursive functions will happen to be total, O.s., defined for every argument.

Definição 29.27. The set of *recursive functions* is the set of partial recursive functions that are total.

cmp:rec:par:
defn:recursive-fn

A recursive function is sometimes called “total recursive” to emphasize that it is defined everywhere.

-NoValue-

[content/computability/recursive-functions/normal-form.tex](#)

29.16 The Normal Form Theorem

Teorema 29.28 (Kleene’s Normal Form Theorem). *There is a primitive recursive relation $T(e, x, s)$ and a primitive recursive function $U(s)$, with the following property: if f is any partial recursive function, then for some e ,*

cmp:rec:nft:
sec
cmp:rec:nft:
thm:kleene-nf

$$f(x) \simeq U(\mu s T(e, x, s))$$

for every x .

explanation

The proof of the normal form theorem is involved, but the basic idea is simple. Every partial recursive function has an *index* e , intuitively, a number

29.17. THE HALTING PROBLEM

coding its program or definition. If $f(x) \downarrow$, the computation can be recorded systematically and coded by some number s , and the fact that s codes the computation of f on input x can be checked primitive recursively using only x and the definition e . Consequently, the relation T , “the function with index e has a computation for input x , and s codes this computation,” is primitive recursive. Given the full record of the computation s , the “upshot” of s is the value of $f(x)$, and it can be obtained from s primitive recursively as well.

The normal form theorem shows that only a single unbounded search is required for the definition of any partial recursive function. Basically, we can search through all numbers until we find one that codes a computation of the function with index e for input x . We can use the numbers e as “names” of partial recursive functions, and write φ_e for the function f defined by the equation in the theorem. Note that any partial recursive function can have more than one index—in fact, every partial recursive function has infinitely many indices.

-NoValue-

[content/computability/recursive-functions/halting-problem.tex](#)

29.17 The Halting Problem

cmp:rec:hlt:
sec

The *halting problem* in general is the problem of deciding, given the specification e (p.ex., program) of a computable function and a number n , whether the computation of the function on input n halts, O.s., produces a result. Famously, Alan Turing proved that this problem itself cannot be solved by a computable function, O.s., the function

$$h(e, n) = \begin{cases} 1 & \text{if computation } e \text{ halts on input } n \\ 0 & \text{otherwise,} \end{cases}$$

is not computable.

In the context of partial recursive functions, the role of the specification of a program may be played by the index e given in Kleene’s normal form theorem. If f is a partial recursive function, any e for which the equation in the normal form theorem holds, is an index of f . Given a number e , the normal form theorem states that

$$\varphi_e(x) \simeq U(\mu s T(e, x, s))$$

is partial recursive, and for every partial recursive $f: \mathbb{N} \rightarrow \mathbb{N}$, there is an $e \in \mathbb{N}$ such that $\varphi_e(x) \simeq f(x)$ for all $x \in \mathbb{N}$. In fact, for each such f there is not just one, but infinitely many such e . The *halting function* h is defined by

$$h(e, x) = \begin{cases} 1 & \text{if } \varphi_e(x) \downarrow \\ 0 & \text{otherwise.} \end{cases}$$

Note that $h(e, x) = 0$ if $\varphi_e(x) \uparrow$, but also when e is not the index of a partial recursive function at all.

Teorema 29.29. *The halting function h is not partial recursive.*

*cmp:rec:hlt:
thm:halting-problem*

Prova. If h were partial recursive, we could define

$$d(y) = \begin{cases} 1 & \text{if } h(y, y) = 0 \\ \mu x \, x \neq y & \text{otherwise.} \end{cases}$$

Since no number x satisfies $x \neq x$, there is no $\mu x \, x \neq x$, and so $d(y) \uparrow$ iff $h(y, y) \neq 0$. From this definition it follows that

1. $d(y) \downarrow$ iff $\varphi_y(y) \uparrow$ or y is not the index of a partial recursive function.
2. $d(y) \uparrow$ iff $\varphi_y(y) \downarrow$.

If h were partial recursive, then d would be partial recursive as well. Thus, by the Kleene normal form theorem, it has an index e_d . Consider the value of $h(e_d, e_d)$. There are two possible cases, 0 and 1.

1. If $h(e_d, e_d) = 1$ then $\varphi_{e_d}(e_d) \downarrow$. But $\varphi_{e_d} \simeq d$, and $d(e_d)$ is defined iff $h(e_d, e_d) = 0$. So $h(e_d, e_d) \neq 1$.
2. If $h(e_d, e_d) = 0$ then either e_d is not the index of a partial recursive function, or it is and $\varphi_{e_d}(e_d) \uparrow$. But again, $\varphi_{e_d} \simeq d$, and $d(e_d)$ is undefined iff $\varphi_{e_d}(e_d) \downarrow$.

The upshot is that e_d cannot, after all, be the index of a partial recursive function. But if h were partial recursive, d would be too, and so our definition of e_d as an index of it would be admissible. We must conclude that h cannot be partial recursive. $\square$

-NoValue-

[content/computability/recursive-functions/general-recursive-functions.tex](#)

29.18 General Recursive Functions

There is another way to obtain a set of total functions. Say a total function $f(x, \vec{z})$ is *regular* if for every sequence of natural numbers $\vec{z}$, there is an x such that $f(x, \vec{z}) = 0$. In other words, the regular functions are exactly those functions to which one can apply unbounded search, and end up with a total function. One can, conservatively, restrict unbounded search to regular functions:

*cmp:rec:gen:
sec*

Definição 29.30. The set of *general recursive functions* is the smallest set of functions from the natural numbers to the natural numbers (of various arities) containing zero, successor, and projections, and closed under composition, primitive recursion, and unbounded search applied to *regular* functions.

*cmp:rec:gen:
defn:general-recursive*

Clearly every general recursive function is total. The difference between [Definição 29.30](#) and [Definição 29.27](#) is that in the latter one is allowed to use partial recursive functions along the way; the only requirement is that the function you end up with at the end is total. So the word “general,” a historic relic, is a misnomer; on the surface, [Definição 29.30](#) is *less* general than [Definição 29.27](#). But, fortunately, the difference is illusory; though the definitions are different, the set of general recursive functions and the set of recursive functions are one and the same.

Capítulo 30

Computability Theory

Material in this chapter should be reviewed and expanded. In particular, there are no exercises yet.

-NoValue- [content/computability/computability-theory/introduction.tex](#)

30.1 Introduction

[cmp:thy:int:
sec](#)

The branch of logic known as *computability theory* deals with issues having to do with the computability, or relative computability, of functions and sets. It is evidence of Kleene’s influence that the subject used to be known as *recursion theory*, and today, both names are commonly used.

Let us call a function $f: \mathbb{N} \rightarrow \mathbb{N}$ *partial computable* if it can be computed in some model of computation. If f is total we will simply say that f is *computable*. A relation R with computable characteristic function χ_R is also called computable. If f and g are partial functions, we will write $f(x) \downarrow$ to mean that f is defined at x , O.s., x is in the domain of f ; and $f(x) \uparrow$ to mean the opposite, O.s., that f is not defined at x . We will use $f(x) \simeq g(x)$ to mean that either $f(x)$ and $g(x)$ are both undefined, or they are both defined and equal.

One can explore the theory of computability without having to refer to a specific model of computation. To do this, one shows that there is a universal

partial computable function $\text{Un}(k, x)$. This allows us to enumerate the partial computable functions. We will adopt the notation φ_k to denote the k -th unary partial computable function, defined by $\varphi_k(x) \simeq \text{Un}(k, x)$. (Kleene used $\{k\}$ for this purpose, but this notation has not been used as much recently.) Slightly more generally, we can uniformly enumerate the partial computable functions of arbitrary arities, and we will use φ_k^n to denote the k -th n -ary partial recursive function.

If $f(\vec{x}, y)$ is a total or partial function, then $\mu y f(\vec{x}, y)$ is the function of $\vec{x}$ that returns the least y such that $f(\vec{x}, y) = 0$, assuming that all of $f(\vec{x}, 0), \dots, f(\vec{x}, y - 1)$ are defined; if there is no such y , $\mu y f(\vec{x}, y)$ is undefined. If $R(\vec{x}, y)$ is a relation, $\mu y R(\vec{x}, y)$ is defined to be the least y such that $R(\vec{x}, y)$ is true; in other words, the least y such that $1 \dot{-} \chi_R(\vec{x}, y) = 0$.

To show that a function is computable, there are two ways one can proceed:

1. Rigorously: describe a Turing machine or partial recursive function explicitly, and show that it computes the function you have in mind;
2. Informally: describe an algorithm that computes it, and appeal to Church's thesis.

There is no fine line between the two; a detailed description of an algorithm should provide enough information so that it is relatively clear how one could, in principle, design the right Turing machine or sequence of partial recursive definitions. Fully rigorous definitions are unlikely to be informative, and we will try to find a happy medium between these two approaches; in short, we will try to find intuitive yet rigorous proofs that the precise definitions could be obtained.

-NoValue-

[content/computability/computability-theory/coding-computations.tex](#)

30.2 Coding Computations

In every model of computation, it is possible to do the following:

[cmp:thy:cod:
sec](#)

1. Describe the *definitions* of computable functions in a systematic way. For instance, you can think of Turing machine specifications, recursive definitions, or programs in a programming language as providing these definitions.
2. Describe the complete record of the computation of a function given by some definition for a given input. For instance, a Turing machine computation can be described by the sequence of configurations (state of the machine, contents of the tape) for each step of computation.
3. Test whether a putative record of a computation is in fact the record of how a computable function with a given definition would be computed for a given input (on which the function is defined, O.s., the computation halts).

30.3. THE NORMAL FORM THEOREM

4. Extract from such a description of the complete record of a computation the value of the function for a given input. For instance, the contents of the tape in the very last step of a halting Turing machine computation is the value.

Using coding, it is possible to assign to each description of a computable function a numerical *index* in such a way that the instructions can be recovered from the index in a computable way. Similarly, the complete record of a computation can be coded by a single number as well. The resulting arithmetical relation “ s codes the record of computation of the function with index e for input x ” and the function “output of computation sequence with code s ” are then computable; in fact, they are primitive recursive.

This fundamental fact is very powerful, and allows us to prove a number of striking and important results about computability, independently of the model of computation chosen.

-NoValue-

`content/computability/computability-theory/normal-form.tex`

30.3 The Normal Form Theorem

`cmp:thy:nfm:`
`sec`

Suppose we can describe definitions of computable functions, and test if some putative description of the complete record of the computation of that function on some input is correct. Then it stands to reason that independently of the model of computation, we can determine the value of any computable function f on any input x as follows:

1. Search through all possible descriptions of records of computation.
2. Test if a given record is the record of a computation of $f(x)$.
3. Extract the value of $f(x)$ from the correct record if we have found it.

That this is in fact true is the content of Kleene’s normal form theorem.

`cmp:thy:nfm:`
`thm:normal-form`

Teorema 30.1 (Kleene’s Normal Form Theorem). *There is a primitive recursive relation $T(e, x, s)$ and a primitive recursive function $U(s)$, with the following property: if f is any partial computable function, then for some e ,*

$$f(x) \simeq U(\mu s T(e, x, s))$$

for every x .

Esboço de prova. For any model of computation one can rigorously define a description of the computable function f and code such description using a natural number e . One can also rigorously define a notion of “computation sequence” which records the process of computing the function with index e for input x . Such a computation sequence can likewise be coded as a number s . This can be done in such a way that

1. the relation $T(e, x, s)$, which holds iff a number s codes the computation sequence of the function with index e on input x , and
2. the function $U(s)$ which maps a computation sequence coded by s to the end result of that computation

are both computable. In fact, the relation T and the function U are primitive recursive. $\square$

explanation

In order to give a rigorous proof of the Normal Form Theorem, we would have to fix a model of computation and carry out the coding of descriptions of computable functions and of computation sequences in detail, and verify that the relation T and function U are primitive recursive. For most applications, it suffices that T and U are computable and that U is total.

It is probably best to remember the proof of the normal form theorem in slogan form: $\mu s T(e, x, s)$ searches for a computation sequence of the function with index e on input x , and U returns the output of the computation sequence if one can be found.

If the model of computation is the partial recursive functions (which is what Kleene originally used), it shows that only a single use of unbounded search, O.s., a single $\mu y f(\vec{x}, y)$ operator is necessary for the definition of any function. In this sense it shows that any partial recursive function has a normal form.

T and U can be used to define the enumeration $\varphi_0, \varphi_1, \varphi_2, \dots$. From now on, we will assume that we have fixed a suitable choice of T and U , and take the equation

$$\varphi_e(x) \simeq U(\mu s T(e, x, s))$$

to be the *definition* of φ_e .

Here is another useful fact:

Teorema 30.2. *Every partial computable function has infinitely many indices.*

Again, this is intuitively clear. Given any (description of) a computable function, one can come up with a different description which computes the same function (input-output pair) but does so, p.ex., by first doing something that has no effect on the computation (say, test if $0 = 0$, or count to 5, etc.). The index of the altered description will always be different from the original index. Both are indices of the same function, just computed slightly differently.

-NoValue-

[content/computability/computability-theory/s-m-n.tex](#)

30.4 The s - m - n Theorem

explanation

The next theorem is known as the “ s - m - n theorem,” for a reason that will be clear in a moment. The hard part is understanding just what the theorem says; once you understand the statement, it will seem fairly obvious.

cmp:thy:smn:
sec

30.5. THE UNIVERSAL PARTIAL COMPUTABLE FUNCTION

cmp:thy:smn:
thm:s-m-n

Teorema 30.3. *For each pair of natural numbers n and m , there is a primitive recursive function s_n^m such that for every sequence $e, a_0, \dots, a_{m-1}, y_0, \dots, y_{n-1}$, we have*

$$\varphi_{s_n^m(e, a_0, \dots, a_{m-1})}^n(y_0, \dots, y_{n-1}) \simeq \varphi_e^{m+n}(a_0, \dots, a_{m-1}, y_0, \dots, y_{n-1}).$$

It is helpful to think of s_n^m as acting on *programs*. That is, s_n^m takes a program e for an $(m+n)$ -ary function, as well as fixed inputs $a_0, \dots, a_{m-1}$; and it returns a program $s_n^m(x, a_0, \dots, a_{m-1})$ for the n -ary function of the remaining arguments. It you think of x as the description of a Turing machine, then $s_n^m(e, a_0, \dots, a_{m-1})$ is the Turing machine that, on input $y_0, \dots, y_{n-1}$, prepends $a_0, \dots, a_{m-1}$ to the input string, and runs e . Each s_n^m is then just a primitive recursive function that finds a code for the appropriate Turing machine.

-NoValue-

[content/computability/computability-theory/universal-part-function.tex](#)

30.5 The Universal Partial Computable Function

cmp:thy:uni:
sec
cmp:thy:uni:
thm:univ-comp

Teorema 30.4. *There is a universal partial computable function $\text{Un}(e, x)$. In other words, there is a function $\text{Un}(e, x)$ such that:*

1. $\text{Un}(e, x)$ is partial computable.
2. If $f(x)$ is any partial computable function, then there is a natural number e such that $f(x) \simeq \text{Un}(e, x)$ for every x .

Prova. Let $\text{Un}(e, x) \simeq U(\mu s T(e, x, s))$, where U and T are as in Kleene's normal form theorem ([Teorema 30.1](#)). □

This is just a precise way of saying that we have an effective enumeration of the partial computable functions; the idea is that if we write f_e for the function defined by $f_e(x) = \text{Un}(e, x)$, then the sequence $f_0, f_1, f_2, \dots$ includes all the partial computable functions, with the property that $f_e(x)$ can be computed “uniformly” in e and x . For simplicity, we are using a binary function that is universal for unary functions, but by coding sequences of numbers we can easily generalize this to more arguments. For example, note that if $f(x, y, z)$ is a 3-place partial recursive function, then the function $g(x) \simeq f((x)_0, (x)_1, (x)_2)$ is a unary recursive function.

-NoValue-

[content/computability/computability-theory/no-universal-function.tex](#)

30.6 No Universal Computable Function

Although there is a partial computable function that is total for the partial computable functions, there is no total computable function that is universal for the total computable functions.

cmp:thy:nou:
sec

Teorema 30.5. *There is no universal computable function. In other words, any function $\text{Un}'(k, x)$ which is such that if $f(x)$ is a total computable function, then there is a natural number k such that $f(x) = \text{Un}'(k, x)$ for every x , is not computable.*

cmp:thy:nou:
thm:no-univ

Prova. The proof is a simple diagonalization: if $\text{Un}'(k, x)$ were total and computable, then

$$d(x) = \text{Un}'(x, x) + 1$$

would also be total and computable. However, by definition, $d(k)$ is not equal to $\text{Un}'(k, k)$. Hence, for every k , the values of $d(x)$ and $\text{Un}'(k, x)$ differ for at least one x , namely $x = k$. $\square$

explanation

Teorema 30.4 above shows that we can get around this diagonalization argument, but only at the expense of allowing the universal function to be partial. That is, Un is universal for the total computable functions, it just isn't total. The diagonalization argument doesn't work in the partial case.

Problema 30.1. To understand why the diagonalization argument in the proof of **Teorema 30.5** does not work in the partial case, consider the function $f(x) \simeq \text{Un}(x, x) + 1$. Is it partial computable? If so, it has an index e , O.s., $f(x) \simeq \text{Un}(e, x)$. What can you say about $f(e)$?

-NoValue- _____
content/computability/computability-theory/halting-problem.tex

30.7 The Halting Problem

By construction, the universal partial computable function $\text{Un}(e, x)$ is defined if and only if the computation of the function coded by e produces a value for input x . It is natural to ask if we can decide whether this is the case. In fact, it is not. For the Turing machine model of computation, this means that whether a given Turing machine halts on a given input is computationally undecidable. The following theorem is therefore known as the “undecidability of the halting problem.” We will provide two proofs below. The first continues the thread of our previous discussion, while the second is more direct.

cmp:thy:hlt:
sec

Teorema 30.6. *Let*

cmp:thy:hlt:
thm:halting-problem

$$h(e, x) = \begin{cases} 1 & \text{if } \text{Un}(e, x) \text{ is defined} \\ 0 & \text{otherwise.} \end{cases}$$

Then h is not computable.

30.7. THE HALTING PROBLEM

Prova. Suppose h is computable. We show that this would let us define a universal computable function. Define

$$\text{Un}'(e, x) = \begin{cases} \text{Un}(e, x) & \text{if } h(e, x) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

But now $\text{Un}'(e, x)$ is a total function, and is computable if h is. For instance, we could define g using primitive recursion, by

$$\begin{aligned} g(0, e, x) &\simeq 0 \\ g(y + 1, e, x) &\simeq \text{Un}(e, x); \end{aligned}$$

then

$$\text{Un}'(e, x) \simeq g(h(e, x), e, x).$$

Since $\text{Un}'(e, x)$ agrees with $\text{Un}(e, x)$ wherever the latter is defined, Un' is universal for those partial computable functions that happen to be total. But this contradicts [Teorema 30.5](#). $\square$

Prova. Suppose $h(e, x)$ were computable. Define the function g by

$$g(x) = \begin{cases} 0 & \text{if } h(x, x) = 0 \\ \uparrow & \text{otherwise.} \end{cases}$$

The function g is partial computable. For example, one can define it as $\mu y \, h(x, x) = 0$. So, for some e , $g(x) \simeq \text{Un}(e, x)$ for every x . Is g defined at e ? If it is, then, by the definition of g , $h(e, e) = 0$ (h can only take the value 0 if it is defined). By the definition of h , this means that $\text{Un}(e, e)$ is undefined. By our assumption that $g(x) \simeq \text{Un}(e, x)$ for every x , we have that $g(e)$ is undefined, a contradiction. On the other hand, if $g(e)$ is undefined, then $h(e, e) \neq 0$, and so $h(e, e) = 1$. It follows that $\text{Un}(e, e)$ is defined. But since $g(x) \simeq \text{Un}(e, x)$, then $g(e)$ would also be defined. Again, a contradiction. $\square$

We can describe this argument in terms of Turing machines. Suppose there [explanation](#) were a Turing machine H that takes as input a description of a Turing machine E and an input x , and decides whether or not E halts on input x . Then we could build another Turing machine G which takes a single input x , runs H to decide if the machine M_x with index x halts on input x , and does the opposite. In other words, if H reports that M_x halts on input x , G goes into an infinite loop, and if H reports that M_x doesn't halt on input x , then G just halts. Does G halt on its own index as input? The argument above shows that it does if and only if it doesn't—a contradiction. So our supposition that there is a such Turing machine H must be false.

-NoValue-

[content/computability/computability-theory/russells-paradox.tex](#)

30.8 Comparison with Russell's Paradox

It is instructive to compare and contrast the arguments in this section with Russell's paradox: cmp:thy:rus:sec

1. Russell's paradox: let $S = \{x : x \notin x\}$. Then $S \in S$ if and only if $X \notin S$, a contradiction.

Conclusion: There is no such set S . Assuming the existence of a “set of all sets” is inconsistent with the other axioms of set theory.

2. A modification of Russell's paradox: let F be the “function” from the set of all functions to $\{0, 1\}$, defined by

$$F(f) = \begin{cases} 1 & \text{if } f \text{ is in the domain of } f, \text{ and } f(f) = 0 \\ 0 & \text{otherwise} \end{cases}$$

A similar argument shows that $F(F) = 0$ if and only if $F(F) = 1$, a contradiction.

Conclusion: F is not a function. The “set of all functions” is too big to be the domain of a function.

3. The diagonalization argument: let $f_0, f_1, \dots$ be the enumeration of the partial computable functions, and let $G: \mathbb{N} \rightarrow \{0, 1\}$ be defined by

$$G(x) = \begin{cases} 1 & \text{if } f_x(x) \downarrow = 0 \\ 0 & \text{otherwise} \end{cases}$$

If G is computable, then it is the function f_k for some k . But then $G(k) = 1$ if and only if $G(k) = 0$, a contradiction.

Conclusion: G is not computable. Note that according to the axioms of set theory, G is still a function; there is no paradox here, just a clarification.

That talk of partial functions, computable functions, partial computable functions, and so on can be confusing. The set of all partial functions from $\mathbb{N}$ to $\mathbb{N}$ is a big collection of objects. Some of them are total, some of them are computable, some are both total and computable, and some are neither. Keep in mind that when we say “function,” by default, we mean a total function. Thus we have:

1. computable functions
2. partial computable functions that are not total
3. functions that are not computable
4. partial functions that are neither total nor computable

30.9. COMPUTABLE SETS

To sort this out, it might help to draw a big square representing all the partial functions from $\mathbb{N}$ to $\mathbb{N}$, and then mark off two overlapping regions, corresponding to the total functions and the computable partial functions, respectively. It is a good exercise to see if you can describe an object in each of the resulting regions in the diagram.

-NoValue-

`content/computability/computability-theory/computable-sets.tex`

30.9 Computable Sets

cmp:thy:cps:
sec We can extend the notion of computability from computable functions to computable sets:

Definição 30.7. Let S be a set of natural numbers. Then S is *computable* iff its characteristic function χ_S is. In other words, S is computable iff the function

$$\chi_S(x) = \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{otherwise} \end{cases}$$

is computable. Similarly, a relation $R(x_0, \dots, x_{k-1})$ is computable if and only if its characteristic function is.

Computable sets and relations are also called *decidable*.

Notice that we now have a number of notions of computability: for partial functions, for functions, and for sets. Do not get them confused! The Turing machine computing a partial function returns the output of the function, for input values at which the function is defined; the Turing machine computing a set returns either 1 or 0, after deciding whether or not the input value is in the set or not.

explanation

-NoValue-

`content/computability/computability-theory/ce-sets.tex`

30.10 Computably Enumerable Sets

cmp:thy:ces:
sec **Definição 30.8.** A set is *computably enumerable* if it is empty or the range of a computable function.

Observações históricas Computably enumerable sets are also called *recursively enumerable* instead. This is the original terminology, and today both are commonly used, as well as the abbreviations “c.e.” and “r.e.”

You should think about what the definition means, and why the terminology is appropriate. The idea is that if S is the range of the computable function f , then

explanation

$$S = \{f(0), f(1), f(2), \dots\},$$

and so f can be seen as “enumerating” the elements of S . Note that according to the definition, f need not be an increasing function, O.s., the enumeration need not be in increasing order. In fact, f need not even be injective, O.s., repetitions in the enumeration $f(0), f(1), f(2), \dots$ of S are allowed. For instance, the constant function $f(x) = 0$ enumerates the set $\{0\}$.

Any computable set is computably enumerable. To see this, suppose S is computable. If S is empty, then by definition it is computably enumerable. Otherwise, let a be any element of S . Define f by

$$f(x) = \begin{cases} x & \text{if } \chi_S(x) = 1 \\ a & \text{otherwise.} \end{cases}$$

Then f is a computable function, and S is the range of f .

-NoValue-

[content/computability/computability-theory/equiv-ce-defs.tex](#)

30.11 Equivalent Definitions of Computably Enumerable Sets

The following gives a number of important equivalent statements of what it means to be computably enumerable.

[cmp:thy:eqc:sec](#)

Teorema 30.9. *Let S be a set of natural numbers. Then the following are equivalent:*

[cmp:thy:eqc:thm:ce-equiv](#)

1. S is computably enumerable.
2. S is the range of a partial computable function.
3. S is empty or the range of a primitive recursive function.
4. S is the domain of a partial computable function.

[cmp:thy:eqc:case:ce](#)

[cmp:thy:eqc:case:ran-pc](#)

[cmp:thy:eqc:case:ran-prim](#)

[cmp:thy:eqc:case:ce-domain](#)

[explanation](#)

The first three clauses say that we can equivalently take any non-empty computably enumerable set to be enumerated by either a computable function, a partial computable function, or a primitive recursive function. The fourth clause tells us that if S is computably enumerable, then for some index e ,

$$S = \{x : \varphi_e(x) \downarrow\}.$$

In other words, S is the set of inputs on for which the computation of φ_e halts. For that reason, computably enumerable sets are sometimes called *semi-decidable*: if a number is in the set, you eventually get a “yes,” but if it isn’t, you never get a “no”!

30.11. DEFINITIONS OF C. E. SETS

Prova. Since every primitive recursive function is computable and every computable function is partial computable, (3) implies (1) and (1) implies (2). (Note that if S is empty, S is the range of the partial computable function that is nowhere defined.) If we show that (2) implies (3), we will have shown the first three clauses equivalent.

So, suppose S is the range of the partial computable function φ_e . If S is empty, we are done. Otherwise, let a be any element of S . By Kleene's normal form theorem, we can write

$$\varphi_e(x) = U(\mu s T(e, x, s)).$$

In particular, $\varphi_e(x) \downarrow$ and $= y$ if and only if there is an s such that $T(e, x, s)$ and $U(s) = y$. Define $f(z)$ by

$$f(z) = \begin{cases} U((z)_1) & \text{if } T(e, (z)_0, (z)_1) \\ a & \text{otherwise.} \end{cases}$$

Then f is primitive recursive, because T and U are. Expressed in terms of Turing machines, if z codes a pair $\langle (z)_0, (z)_1 \rangle$ such that $(z)_1$ is a halting computation of machine M_e on input $(z)_0$, then f returns the output of the computation; otherwise, it returns a . We need to show that S is the range of f , O.s., for any natural number y , $y \in S$ if and only if it is in the range of f . In the forwards direction, suppose $y \in S$. Then y is in the range of φ_e , so for some x and s , $T(e, x, s)$ holds and $U(s) = y$. But then $y = f(\langle x, s \rangle)$. Conversely, suppose y is in the range of f . Then either $y = a$, or for some z , $T(e, (z)_0, (z)_1)$ and $U((z)_1) = y$. Since, in the latter case, $\varphi_e(x) \downarrow = y$, either way, y is in S .

(The notation $\varphi_e(x) \downarrow = y$ means " $\varphi_e(x)$ is defined and equal to y ." We could just as well use $\varphi_e(x) = y$, but the extra arrow is sometimes helpful in reminding us that we are dealing with a partial function.)

To finish up the proof of Teorema 30.9, it suffices to show that (1) and (4) are equivalent. First, let us show that (1) implies (4). Suppose S is the range of a computable function f , O.s.,

$$S = \{y : \text{for some } x, f(x) = y\}.$$

Let

$$g(y) = \mu x (f(x) = y).$$

Then g is a partial computable function, and $g(y)$ is defined if and only if for some x , $f(x) = y$. In other words, the domain of g is the range of f . Expressed in terms of Turing machines: given a Turing machine F that enumerates the elements of S , let G be the Turing machine that semi-decides S by searching through the outputs of F to see if a given element is in the set, halts if it is and keeps searching forever if it isn't.

Finally, to show (4) implies (1), suppose that S is the domain of the partial computable function φ_e , O.s.,

$$S = \{x : \varphi_e(x) \downarrow\}.$$

If S is empty, we are done; otherwise, let a be any element of S . Define f by

$$f(z) = \begin{cases} (z)_0 & \text{if } T(e, (z)_0, (z)_1) \\ a & \text{otherwise.} \end{cases}$$

Then, as above, a number x is in the range of f if and only if $\varphi_e(x) \downarrow$, O.s., if and only if $x \in S$. Expressed in terms of Turing machines: given a machine M_e that semi-decides S , enumerate the elements of S by running through all possible Turing machine computations, and returning the inputs that correspond to halting computations. $\square$

Clause (4) of **Teorema 30.9** provides us with a convenient way of enumerating the computably enumerable sets: for each e , let W_e denote the domain of φ_e , O.s.,

$$W_e = \{x : \varphi_e(x) \downarrow\}.$$

Then if A is any computably enumerable set, $A = W_e$, for some e .

The following provides yet another characterization of the computably enumerable sets.

Teorema 30.10. *A set S is computably enumerable if and only if there is a computable relation $R(x, y)$ such that* *cmp:thy:eqc:
thm:exists-char*

$$S = \{x : \exists y R(x, y)\}.$$

Prova. In the forward direction, suppose S is computably enumerable. Then for some e , $S = W_e$. For this value of e we can write S as

$$S = \{x : \exists y T(e, x, y)\}.$$

In the reverse direction, suppose $S = \{x : \exists y R(x, y)\}$. Define f by

$$f(x) \simeq \mu y R(x, y).$$

Then f is partial computable, and S is the domain of f . $\square$

-NoValue-

 content/computability/computability-theory/non-comp-set.tex

30.12 There Are Non-Computable Sets

We saw above that every computable set is computably enumerable. Is the converse true? The following shows that, in general, it is not. *cmp:thy:nep:
sec*

Teorema 30.11. *Let K_0 be the set $\{\langle e, x \rangle : \varphi_e(x) \downarrow\}$. Then K_0 is computably enumerable but not computable.* *cmp:thy:nep:
thm:K-0*

30.13. UNION AND INTERSECTION OF C.E. SETS

Prova. To see that K_0 is computably enumerable, note that it is the domain of the function f defined by

$$f(z) = \mu y (\text{len}(z) = 2 \wedge T((z)_0, (z)_1, y)).$$

For, if $\varphi_e(x)$ is defined, $f(\langle e, x \rangle)$ finds a halting computation sequence; if $\varphi_e(x)$ is undefined, so is $f(\langle e, x \rangle)$; and if z doesn't even code a pair, then $f(z)$ is also undefined.

The fact that K_0 is not computable is just the undecidability of the halting problem, **Teorema 30.6**. $\square$

The set K_0 is the set of pairs $\langle e, x \rangle$ such that $\varphi_e(x) \downarrow$, O.s., $\langle e, x \rangle \in K_0$ iff φ_e is defined (halts) on input x , so it is also called the “halting set.” The set $K = \{e : \varphi_e(e) \downarrow\}$ is the “self-halting set.” It is often used as a canonical undecidable set.

cmp:thy:nep: thm:K **Teorema 30.12.** *The self-halting set $K = \{e : \varphi_e(e) \downarrow\}$ is c.e. but not decidable.*

Prova. Suppose K is decidable, O.s., its characteristic function χ_K is computable. Let

$$d(e) = \begin{cases} 1 & \text{if } \chi_K(e) = 0 \\ \uparrow & \text{otherwise.} \end{cases}$$

Let k be the index of d , O.s., $d \simeq \varphi_k$. Then $d(k) \simeq \varphi_k(k)$. This contradicts the fact that $d(k) \downarrow$ iff $\varphi_k(k) \uparrow$, which follows from the definition of d .

K is the domain of $f(x) = \mu y T(x, x, y)$ and so is c.e. $\square$

-NoValue-

content/computability/computability-theory/ce-closed-cup-cap.tex

30.13 Computably Enumerable Sets are Closed under Union and Intersection

cmp:thy:clo: sec The following theorem gives some closure properties on the set of computably enumerable sets.

Teorema 30.13. *Suppose A and B are computably enumerable. Then so are $A \cap B$ and $A \cup B$.*

Prova. **Teorema 30.9** allows us to use various characterizations of the computably enumerable sets. By way of illustration, we will provide a few different proofs.

For the first proof, suppose A is enumerated by a computable function f , and B is enumerated by a computable function g . Let

$$\begin{aligned} h(x) &= \mu y (f(y) = x \vee g(y) = x) \text{ and} \\ j(x) &= \mu y (f((y)_0) = x \wedge g((y)_1) = x). \end{aligned}$$

Then $A \cup B$ is the domain of h , and $A \cap B$ is the domain of j .

explanation Here is what is going on, in computational terms: given procedures that enumerate A and B , we can semi-decide if an element x is in $A \cup B$ by looking for x in either enumeration; and we can semi-decide if an element x is in $A \cap B$ for looking for x in both enumerations at the same time.

For the second proof, suppose again that A is enumerated by f and B is enumerated by g . Let

$$k(x) = \begin{cases} f(x/2) & \text{if } x \text{ is even} \\ g((x-1)/2) & \text{if } x \text{ is odd.} \end{cases}$$

Then k enumerates $A \cup B$; the idea is that k just alternates between the enumerations offered by f and g . Enumerating $A \cap B$ is trickier. If $A \cap B$ is empty, it is trivially computably enumerable. Otherwise, let c be any element of $A \cap B$, and define l by

$$l(x) = \begin{cases} f((x)_0) & \text{if } f((x)_0) = g((x)_1) \\ c & \text{otherwise.} \end{cases}$$

In computational terms, l runs through pairs of elements in the enumerations of f and g , and outputs every match it finds; otherwise, it just stalls by outputting c .

For the last proof, suppose A is the *domain* of the partial function $m(x)$ and B is the domain of the partial function $n(x)$. Then $A \cap B$ is the domain of the partial function $m(x) + n(x)$.

explanation In computational terms, if A is the set of values for which m halts and B is the set of values for which n halts, $A \cap B$ is the set of values for which both procedures halt.

Expressing $A \cup B$ as a set of halting values is more difficult, because one has to simulate m and n in parallel. Let d be an index for m and let e be an index for n ; in other words, $m = \varphi_d$ and $n = \varphi_e$. Then $A \cup B$ is the domain of the function

$$p(x) = \mu y (T(d, x, y) \vee T(e, x, y)).$$

explanation In computational terms, on input x , p searches for either a halting computation for m or a halting computation for n , and halts if it finds either one. $\square$

-NoValue-

<content/computability/computability-theory/complement-ce.tex>

30.14 Computably Enumerable Sets not Closed under Complement

Suppose A is computably enumerable. Is the complement of A , $\overline{A} = \mathbb{N} \setminus A$, always computably enumerable as well? The following theorem and corollary show that the answer is “no.”

[cmp:thy:cmp:sec](#)

30.15. REDUCIBILITY

cmp:thy:cmp: thm:ce-comp **Teorema 30.14.** *Let A be any set of natural numbers. Then A is computable if and only if both A and $\overline{A}$ are computably enumerable.*

Prova. The forwards direction is easy: if A is computable, then $\overline{A}$ is computable as well ($\chi_A = 1 - \chi_{\overline{A}}$), and so both are computably enumerable.

In the other direction, suppose A and $\overline{A}$ are both computably enumerable. Let A be the domain of φ_d , and let $\overline{A}$ be the domain of φ_e . Define h by

$$h(x) = \mu s (T(d, x, s) \vee T(e, x, s)).$$

In other words, on input x , h searches for either a halting computation of φ_d or a halting computation of φ_e . Now, if $x \in A$, it will succeed in the first case, and if $x \in \overline{A}$, it will succeed in the second case. So, h is a total computable function. But now we have that for every x , $x \in A$ if and only if $T(e, x, h(x))$, O.s., if φ_e is the one that is defined. Since $T(e, x, h(x))$ is a computable relation, A is computable. $\square$

It is easier to understand what is going on in informal computational terms: *explanation* to decide A , on input x search for halting computations of φ_e and φ_f . One of them is bound to halt; if it is φ_e , then x is in A , and otherwise, x is in $\overline{A}$.

cmp:thy:cmp: cor:comp-k **Corolário 30.15.** *$\overline{K_0}$ is not computably enumerable.*

Prova. We know that K_0 is computably enumerable, but not computable. If $\overline{K_0}$ were computably enumerable, then K_0 would be computable by **Teorema 30.14**, contradicting **Teorema 30.11**. $\square$

-NoValue- <content/computability/computability-theory/reducibility.tex>

30.15 Reducibility

cmp:thy:red: sec We now know that there is at least one set, K_0 , that is computably enumerable *explanation* but not computable. It should be clear that there are others. The method of reducibility provides a powerful method of showing that other sets have these properties, without constantly having to return to first principles.

Generally speaking, a “reduction” of a set A to a set B is a method of transforming answers to whether or not **membros** are in B into answers as to whether or not **membros** are in A . We will focus on a notion called “many-one reducibility,” but there are many other notions of reducibility available, with varying properties. Notions of reducibility are also central to the study of computational complexity, where efficiency issues have to be considered as well. For example, a set is said to be “NP-complete” if it is in NP and every NP problem can be reduced to it, using a notion of reduction that is similar to the one described below, only with the added requirement that the reduction can be computed in polynomial time.

We have already used the notion of reduction notion implicitly. Define the set K by

$$K = \{x : \varphi_x(x) \downarrow\},$$

O.s., $K = \{x : x \in W_x\}$. Our proof that the halting problem is unsolvable (Teorema 30.6) shows most directly that K is not computable. Recall that K_0 is the set

$$K_0 = \{\langle e, x \rangle : \varphi_e(x) \downarrow\},$$

O.s., $K_0 = \{\langle x, e \rangle : x \in W_e\}$. It is easy to extend any proof of the uncomputability of K to the uncomputability of K_0 : if K_0 were computable, we could decide whether or not **um membro** x is in K simply by asking whether or not the pair $\langle x, x \rangle$ is in K_0 . The function f which maps x to $\langle x, x \rangle$ is an example of a *reduction* of K to K_0 .

Definição 30.16. Let A and B be sets of natural numbers. A computable function $f: \mathbb{N} \rightarrow \mathbb{N}$ is a *many-one reduction* of A to B iff, for every natural number x ,

$$x \in A \quad \text{if and only if} \quad f(x) \in B.$$

If such a reduction f exists, we say that A is *many-one reducible* to B , written $A \leq_m B$. If A is many-one reducible to B and vice-versa, then A and B are said to be *many-one equivalent*, written $A \equiv_m B$.

If the function f in the definition above happens to be injective, A is said to be *one-one reducible* to B . Most of the reductions described below meet this stronger requirement, but we will not use this fact.

digression

It is true, but by no means obvious, that one-one reducibility really is a stronger requirement than many-one reducibility. In other words, there are infinite sets A and B such that A is many-one reducible to B but not one-one reducible to B .

-NoValue-

<content/computability/computability-theory/prop-reduce.tex>

30.16 Properties of Reducibility

We write $A \leq_m B$ if A reduces to B , and this notation suggests that if $A \leq_m B$, then A is “no harder than” B and that B is “as hard or harder than” A , and that $\leq_m$, like the usual $\leq$ on numbers, orders sets (or decision problems) by their complexity. The following two propositions support this intuition. The first one says that $\leq_m$ is transitive.

cmp:thy:ppr:sec

Proposição 30.17. If $A \leq_m B$ and $B \leq_m C$, then $A \leq_m C$.

cmp:thy:ppr:prop:trans-red

Prova. Composing a reduction of A to B with a reduction of B to C yields a reduction of A to C . $\square$

30.16. PROPERTIES OF REDUCIBILITY

Problema 30.2. Prove [Proposição 30.17](#) by showing that if f and g are many-one reductions of A to B and B to C , respectively, then $g \circ f$ is a many-one reduction of A to C .

cmp:thy:ppr:
prop:reduce **Proposição 30.18.** Let A and B be any sets, and suppose $A \leq_m B$.

1. If B is computably enumerable, so is A .
2. If B is computable, so is A .

Prova. Let f be a many-one reduction from A to B . For the first claim, just check that if B is the domain of a partial function g , then A is the domain of $g \circ f$:

$$\begin{aligned} x \in A &\text{ iff } f(x) \in B \\ &\text{ iff } g(f(x)) \downarrow. \end{aligned}$$

For the second claim, remember that if B is computable then B and $\overline{B}$ are computably enumerable ([Teorema 30.14](#)). It is not hard to check that f is also a many-one reduction of $\overline{A}$ to $\overline{B}$, so, by the first part of this proof, A and $\overline{A}$ are [computavelmente enumerável](#). So A is computable as well by [Teorema 30.14](#). (Alternatively, you can check that $\chi_A = \chi_B \circ f$; so if χ_B is computable, then so is χ_A .) $\square$

Problema 30.3. Suppose f is a many-one reduction of A to B . Show that f is also a many-one reduction of $\overline{A}$ to $\overline{B}$.

Problema 30.4. Show that if $f: A \rightarrow B$ is a many-one reduction, then $\chi_A = \chi_B \circ f$.

A more general notion of reducibility called *Turing reducibility* is useful in other contexts, especially for proving undecidability results. Note that by [Corolário 30.15](#), the complement of K_0 is not reducible to K_0 , since it is not computably enumerable. But, intuitively, if you knew the answers to questions about K_0 , you would know the answer to questions about its complement as well. A set A is said to be Turing reducible to B if one can determine answers to questions in A using a computable procedure that can ask questions about B . This is more liberal than many-one reducibility, in which (1) you are only allowed to ask one question about B , and (2) a “yes” answer has to translate to a “yes” answer to the question about A , and similarly for “no.” It is still the case that if A is Turing reducible to B and B is computable then A is computable as well (though, as we have seen, the analogous statement does not hold for computable enumerability). digression

You should think about the various notions of reducibility we have discussed, and understand the distinctions between them. We will, however, only deal with many-one reducibility in this chapter. Incidentally, both types of reducibility discussed in the last paragraph have analogues in computational

complexity, with the added requirement that the Turing machines run in polynomial time: the complexity version of many-one reducibility is known as *Karp reducibility*, while the complexity version of Turing reducibility is known as *Cook reducibility*.

-NoValue-

[content/computability/computability-theory/complete-ce-sets.tex](#)

30.17 Complete Computably Enumerable Sets

Definição 30.19. A set A is a *complete computavelmente enumerável set* (under many-one reducibility) if

[cmp:thy:cce:sec](#)

1. A is computably enumerable, and
2. for any other computably enumerable set B , $B \leq_m A$.

In other words, complete computably enumerable sets are the “hardest” computably enumerable sets possible. They allow one to answer questions about *any* computably enumerable set.

Teorema 30.20. K , K_0 , and K_1 are all complete computably enumerable sets.

Prova. To see that K_0 is complete, let B be any computably enumerable set. Then for some index e ,

$$B = W_e = \{x : \varphi_e(x) \downarrow\}.$$

Let f be the function $f(x) = \langle e, x \rangle$. Then for every natural number x , $x \in B$ if and only if $f(x) \in K_0$. In other words, f reduces B to K_0 .

To see that K_1 is complete, note that in the proof of [Proposição 30.21](#) we reduced K_0 to it. So, by [Proposição 30.17](#), any computably enumerable set can be reduced to K_1 as well.

K can be reduced to K_0 in much the same way. □

Problema 30.5. Give a reduction of K to K_0 .

[digression](#)

So, it turns out that all the examples of computably enumerable sets that we have considered so far are either computable, or complete. This should seem strange! Are there any examples of computably enumerable sets that are neither computable nor complete? The answer is yes, but it wasn’t until the middle of the 1950s that this was established by Friedberg and Muchnik, independently.

-NoValue-

[content/computability/computability-theory/k-1.tex](#)

30.18 An Example of Reducibility

Let us consider an application of **Proposição 30.18**.

cmp:thy:k1:
sec

cmp:thy:k1:
prop:k1 **Proposição 30.21.** *Let*

$$K_1 = \{e : \varphi_e(0) \downarrow\}.$$

Then K_1 is computably enumerable but not computable.

Prova. Since $K_1 = \{e : \exists s T(e, 0, s)\}$, K_1 is computably enumerable by **Teorema 30.10**.

To show that K_1 is not computable, let us show that K_0 is reducible to it.

This is a little bit tricky, since using K_1 we can only ask questions about computations that start with a particular input, 0. Suppose you have a smart friend who can answer questions of this type (friends like this are known as “oracles”). Then suppose someone comes up to you and asks you whether or not $\langle e, x \rangle$ is in K_0 , that is, whether or not machine e halts on input x . One thing you can do is build another machine, e_x , that, for *any* input, ignores that input and instead runs e on input x . Then clearly the question as to whether machine e halts on input x is equivalent to the question as to whether machine e_x halts on input 0 (or any other input). So, then you ask your friend whether this new machine, e_x , halts on input 0; your friend’s answer to the modified question provides the answer to the original one. This provides the desired reduction of K_0 to K_1 .

explanation

Using the universal partial computable function, let f be the 3-ary function defined by

$$f(x, y, z) \simeq \varphi_x(y).$$

Note that f ignores its third input entirely. Pick an index e such that $f = \varphi_e^3$; so we have

$$\varphi_e^3(x, y, z) \simeq \varphi_x(y).$$

By the s - m - n theorem, there is a function $s(e, x, y)$ such that, for every z ,

$$\begin{aligned} \varphi_{s(e, x, y)}(z) &\simeq \varphi_e^3(x, y, z) \\ &\simeq \varphi_x(y). \end{aligned}$$

In terms of the informal argument above, $s(e, x, y)$ is an index for the machine that, for any input z , ignores that input and computes $\varphi_x(y)$.

explanation

In particular, we have

$$\varphi_{s(e, x, y)}(0) \downarrow \quad \text{if and only if} \quad \varphi_x(y) \downarrow.$$

In other words, $\langle x, y \rangle \in K_0$ if and only if $s(e, x, y) \in K_1$. So the function g defined by

$$g(w) = s(e, (w)_0, (w)_1)$$

is a reduction of K_0 to K_1 . □

-NoValue-

content/computability/computability-theory/total.tex

30.19 Totality is Undecidable

Let us consider one more example of using the *s-m-n* theorem to show that something is noncomputable. Let Tot be the set of indices of total computable functions, O.s., cmp:thy:tot:
sec

$$\text{Tot} = \{x : \text{for every } y, \varphi_x(y) \downarrow\}.$$

Proposição 30.22. *Tot is not computable.* cmp:thy:tot:
prop:total

Prova. To see that Tot is not computable, it suffices to show that *K* is reducible to it. Let $h(x, y)$ be defined by

$$h(x, y) \simeq \begin{cases} 0 & \text{if } x \in K \\ \uparrow & \text{otherwise} \end{cases}$$

Note that $h(x, y)$ does not depend on y at all. It should not be hard to see that h is partial computable: on input x, y , we compute h by first simulating the function φ_x on input x ; if this computation halts, $h(x, y)$ outputs 0 and halts. So $h(x, y)$ is just $\text{zero}(\mu s T(x, x, s))$, where zero is the constant zero function.

Using the *s-m-n* theorem, there is a primitive recursive function $k(x)$ such that for every x and y ,

$$\varphi_{k(x)}(y) = \begin{cases} 0 & \text{if } x \in K \\ \uparrow & \text{otherwise} \end{cases}$$

So $\varphi_{k(x)}$ is total if $x \in K$, and undefined otherwise. Thus, k is a reduction of *K* to Tot. □

digression It turns out that Tot is not even computably enumerable—its complexity lies further up on the “arithmetical hierarchy.” But we will not worry about this strengthening here.

-NoValue-

<content/computability/computability-theory/rice-theorem.tex>

30.20 Rice’s Theorem

If you think about it, you will see that the specifics of Tot do not play into the proof of **Proposição 30.22**. We designed $h(x, y)$ to act like the constant function $j(y) = 0$ exactly when x is in *K*; but we could just as well have made it act like any other partial computable function under those circumstances. This observation lets us state a more general theorem, which says, roughly, that no nontrivial property of computable functions is decidable. cmp:thy:rce:
sec

Keep in mind that $\varphi_0, \varphi_1, \varphi_2, \dots$ is our standard enumeration of the partial computable functions.

Teorema 30.23 (Rice’s Theorem). *Let C be any set of partial computable functions, and let $A = \{n : \varphi_n \in C\}$. If A is computable, then either C is empty or C is the set of all the partial computable functions.*

30.20. RICE'S THEOREM

An *index set* is a set A with the property that if n and m are indices which “compute” the same function, then either both n and m are in A , or neither is. It is not hard to see that the set A in the theorem has this property. Conversely, if A is an index set and C is the set of functions computed by these indices, then $A = \{n : \varphi_n \in C\}$.

With this terminology, Rice’s theorem is equivalent to saying that no non-trivial index set is decidable. To understand what the theorem says, it is helpful to emphasize the distinction between *programs* (say, in your favorite programming language) and the functions they compute. There are certainly questions about programs (indices), which are syntactic objects, that are computable: does this program have more than 150 symbols? Does it have more than 22 lines? Does it have a “while” statement? Does the string “hello world” ever appear as the argument to a “print” statement? Rice’s theorem says that no nontrivial question about the program’s *behavior* is computable. This includes questions like these: does the program halt on input 0? Does it ever halt? Does it ever output an even number?

explanation

Prova do teorema de Rice. Suppose C is neither empty nor the set of all the partial computable functions, and let A be the set of indices of functions in C . We will show that if A were computable, we could solve the halting problem; so A is not computable.

Without loss of generality, we can assume that the function f which is nowhere defined is not in C (otherwise, switch C and its complement in the argument below). Let g be any function in C . The idea is that if we could decide A , we could tell the difference between indices computing f , and indices computing g ; and then we could use that capability to solve the halting problem.

Here’s how. Using the universal partial computable functions, we can define a function

$$h(x, y) \simeq \begin{cases} \text{undefined} & \text{if } \varphi_x(x) \uparrow \\ g(y) & \text{otherwise.} \end{cases}$$

To compute h , first we try to compute $\varphi_x(x)$; if that computation halts, we go on to compute $g(y)$; and if *that* computation halts, we return the output. More formally, we can write

$$h(x, y) \simeq P_0^2(g(y), \text{Un}(x, x)).$$

where $P_0^2(z_0, z_1) = z_0$ is the 2-place projection function returning the 0-th argument, which is computable.

Then h is a composition of partial computable functions, and the right side is defined and equal to $g(y)$ just when $\text{Un}(x, x)$ and $g(y)$ are both defined.

Notice that for a fixed x , if $\varphi_x(x)$ is undefined, then $h(x, y)$ is undefined for every y ; and if $\varphi_x(x)$ is defined, then $h(x, y) \simeq g(y)$. So, for any fixed value of x , either $h(x, y)$ acts just like f or it acts just like g , and deciding whether or not $\varphi_x(x)$ is defined amounts to deciding which of these two cases holds. But

this amounts to deciding whether or not $h_x(y) \simeq h(x, y)$ is in C or not, and if A were computable, we could do just that.

More formally, since h is partial computable, it is equal to the function φ_e for some index e . By the s - m - n theorem there is a primitive recursive function s such that for each x , $\varphi_{s(e,x)}(y) = h_x(y)$. Now we have that for each x , if $\varphi_x(x) \downarrow$, then $\varphi_{s(e,x)}$ is the same function as g , and so $s(e, x)$ is in A . On the other hand, if $\varphi_x(x) \uparrow$, then $\varphi_{s(e,x)}$ is the same function as f , and so $s(e, x)$ is not in A . In other words, we have that for every x , $x \in K$ if and only if $s(e, x) \in A$. If A were computable, K would be also, which is a contradiction. So A is not computable. $\square$

Rice's theorem is very powerful. The following immediate corollary shows some sample applications.

Corolário 30.24. *The following sets are undecidable.*

1. $\{x : 17 \text{ is in the range of } \varphi_x\}$
2. $\{x : \varphi_x \text{ is constant}\}$
3. $\{x : \varphi_x \text{ is total}\}$
4. $\{x : \text{whenever } y < y', \varphi_x(y) \downarrow, \text{ and if } \varphi_x(y') \downarrow, \text{ then } \varphi_x(y) < \varphi_x(y')\}$

Prova. These are all nontrivial index sets. $\square$

-NoValue- _____
 content/computability/computability-theory/fixed-point-thm.tex

30.21 The Fixed-Point Theorem

Let's consider the halting problem again. As temporary notation, let us write $\ulcorner \varphi_x(y) \urcorner$ for $\langle x, y \rangle$; think of this as representing a “name” for the value $\varphi_x(y)$. With this notation, we can reword one of our proofs that the halting problem is undecidable. cmp:thy:fix:sec

Question: is there a computable function h , with the following property? For every x and y ,

$$h(\ulcorner \varphi_x(y) \urcorner) = \begin{cases} 1 & \text{if } \varphi_x(y) \downarrow \\ 0 & \text{otherwise.} \end{cases}$$

Answer: No; otherwise, the partial function

$$g(x) \simeq \begin{cases} 0 & \text{if } h(\ulcorner \varphi_x(x) \urcorner) = 0 \\ \text{undefined} & \text{otherwise} \end{cases}$$

would be computable, and so have some index e . But then we have

$$\varphi_e(e) \simeq \begin{cases} 0 & \text{if } h(\ulcorner \varphi_e(e) \urcorner) = 0 \\ \text{undefined} & \text{otherwise,} \end{cases}$$

30.21. THE FIXED-POINT THEOREM

in which case $\varphi_e(e)$ is defined if and only if it isn't, a contradiction.

Now, take a look at the equation with φ_e . There is an instance of self-reference there, in a sense: we have arranged for the value of $\varphi_e(e)$ to depend on $\ulcorner \varphi_e(e) \urcorner$, in a certain way. The fixed-point theorem says that we *can* do this, in general—not just for the sake of proving contradictions.

Lema 30.25 gives two equivalent ways of stating the fixed-point theorem. Logically speaking, the fact that the statements are equivalent follows from the fact that they are both true; but what we really mean is that each one follows straightforwardly from the other, so that they can be taken as alternative statements of the same theorem.

cmp:thy:fix:
lem:fixed-equiv **Lema 30.25.** *The following statements are equivalent:*

1. *For every partial computable function $g(x, y)$, there is an index e such that for every y ,*

$$\varphi_e(y) \simeq g(e, y).$$

2. *For every computable function $f(x)$, there is an index e such that for every y ,*

$$\varphi_e(y) \simeq \varphi_{f(e)}(y).$$

Prova. (1) $\Rightarrow$ (2): Given f , define g by $g(x, y) \simeq \text{Un}(f(x), y)$. Use (1) to get an index e such that for every y ,

$$\begin{aligned} \varphi_e(y) &= \text{Un}(f(e), y) \\ &= \varphi_{f(e)}(y). \end{aligned}$$

(2) $\Rightarrow$ (1): Given g , use the s - m - n theorem to get f such that for every x and y , $\varphi_{f(x)}(y) \simeq g(x, y)$. Use (2) to get an index e such that

$$\begin{aligned} \varphi_e(y) &= \varphi_{f(e)}(y) \\ &= g(e, y). \end{aligned}$$

This concludes the proof. □

Before showing that statement (1) is true (and hence (2) as well), consider explanation how bizarre it is. Think of e as being a computer program; statement (1) says that given any partial computable $g(x, y)$, you can find a computer program e that computes $g_e(y) \simeq g(e, y)$. In other words, you can find a computer program that computes a function that references the program itself.

Teorema 30.26. *The two statements in **Lema 30.25** are true. Specifically, for every partial computable function $g(x, y)$, there is an index e such that for every y ,*

$$\varphi_e(y) \simeq g(e, y).$$

Prova. The ingredients are already implicit in the discussion of the halting problem above. Let $\text{diag}(x)$ be a computable function which for each x returns an index for the function $f_x(y) \simeq \varphi_x(x, y)$, O.s.,

$$\varphi_{\text{diag}(x)}(y) \simeq \varphi_x(x, y).$$

Think of diag as a function that transforms a program for a 2-ary function into a program for a 1-ary function, obtained by fixing the original program as its first argument. The function diag can be defined formally as follows: first define s by

$$s(x, y) \simeq \text{Un}^2(x, x, y),$$

where Un^2 is a 3-ary function that is universal for partial computable 2-ary functions. Then, by the s - m - n theorem, we can find a primitive recursive function diag satisfying

$$\varphi_{\text{diag}(x)}(y) \simeq s(x, y).$$

Now, define the function l by

$$l(x, y) \simeq g(\text{diag}(x), y).$$

and let $\ulcorner l \urcorner$ be an index for l . Finally, let $e = \text{diag}(\ulcorner l \urcorner)$. Then for every y , we have

$$\begin{aligned} \varphi_e(y) &\simeq \varphi_{\text{diag}(\ulcorner l \urcorner)}(y) \\ &\simeq \varphi_{\ulcorner l \urcorner}(\ulcorner l \urcorner, y) \\ &\simeq l(\ulcorner l \urcorner, y) \\ &\simeq g(\text{diag}(\ulcorner l \urcorner), y) \\ &\simeq g(e, y), \end{aligned}$$

as required. □

explanation What's going on? Suppose you are given the task of writing a computer program that prints itself out. Suppose further, however, that you are working with a programming language with a rich and bizarre library of string functions. In particular, suppose your programming language has a function diag which works as follows: given an input string s , diag locates each instance of the symbol 'x' occurring in s , and replaces it by a quoted version of the original string. For example, given the string

`hello x world`

as input, the function returns

`hello 'hello x world' world`

as output. In that case, it is easy to write the desired program; you can check that

30.21. THE FIXED-POINT THEOREM

```
print(diag('print(diag(x))'))
```

does the trick. For more common programming languages like C++ and Java, the same idea (with a more involved implementation) still works.

We are only a couple of steps away from the proof of the fixed-point theorem. Suppose a variant of the print function $\text{print}(x, y)$ accepts a string x and another numeric argument y , and prints the string x repeatedly, y times. Then the “program”

```
getinput(y);
print(diag('getinput(y); print(diag(x), y)'), y)
```

prints itself out y times, on input y . Replacing the getinput—print—diag skeleton by an arbitrary function $g(x, y)$ yields

```
g(diag('g(diag(x), y)'), y)
```

which is a program that, on input y , runs g on the program itself and y . Thinking of “quoting” with “using an index for,” we have the proof above.

For now, it is o.k. if you want to think of the proof as formal trickery, or black magic. But you should be able to reconstruct the details of the argument given above. When we prove the incompleteness theorems (and the related “fixed-point theorem”) we will discuss other ways of understanding why it works.

The same idea can be used to get a “fixed point” combinator. Suppose you have a lambda term g , and you want another term k with the property that k is β -equivalent to gk . Define terms

$$\text{diag}(x) = xx$$

and

$$l(x) = g(\text{diag}(x))$$

using our notational conventions; in other words, l is the term $\lambda x. g(xx)$. Let k be the term ll . Then we have

$$\begin{aligned} k &= (\lambda x. g(xx))(\lambda x. g(xx)) \\ &\rightarrow g((\lambda x. g(xx))(\lambda x. g(xx))) \\ &= gk. \end{aligned}$$

If one takes

$$Y = \lambda g. ((\lambda x. g(xx))(\lambda x. g(xx)))$$

then Yg and $g(Yg)$ reduce to a common term; so $Yg \equiv_{\beta} g(Yg)$. This is known as “Curry’s combinator.” If instead one takes

$$Y = (\lambda xg. g(xg))(\lambda xg. g(xg))$$

then in fact Yg reduces to $g(Yg)$, which is a stronger statement. This latter version of Y is known as “Turing’s combinator.”

-NoValue-

[content/computability/computability-theory/application-fixed-point.tex](#)

30.22 Applying the Fixed-Point Theorem

The fixed-point theorem essentially lets us define partial computable functions in terms of their indices. For example, we can find an index e such that for every y ,

$$\varphi_e(y) = e + y.$$

As another example, one can use the proof of the fixed-point theorem to design a program in Java or C++ that prints itself out.

Remember that if for each e , we let W_e be the domain of φ_e , then the sequence $W_0, W_1, W_2, \dots$ enumerates the computably enumerable sets. Some of these sets are computable. One can ask if there is an algorithm which takes as input a value x , and, if W_x happens to be computable, returns an index for its characteristic function. The answer is “no,” there is no such algorithm:

Teorema 30.27. *There is no partial computable function f with the following property: whenever W_e is computable, then $f(e)$ is defined and $\varphi_{f(e)}$ is its characteristic function.*

Prova. Let f be any computable function; we will construct an e such that W_e is computable, but $\varphi_{f(e)}$ is not its characteristic function. Using the fixed point theorem, we can find an index e such that

$$\varphi_e(y) \simeq \begin{cases} 0 & \text{if } y = 0 \text{ and } \varphi_{f(e)}(0) \downarrow = 0 \\ \text{undefined} & \text{otherwise.} \end{cases}$$

That is, e is obtained by applying the fixed-point theorem to the function defined by

$$g(x, y) \simeq \begin{cases} 0 & \text{if } y = 0 \text{ and } \varphi_{f(x)}(0) \downarrow = 0 \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Informally, we can see that g is partial computable, as follows: on input x and y , the algorithm first checks to see if y is equal to 0. If it is, the algorithm computes $f(x)$, and then uses the universal machine to compute $\varphi_{f(x)}(0)$. If this last computation halts and returns 0, the algorithm returns 0; otherwise, the algorithm doesn't halt.

But now notice that if $\varphi_{f(e)}(0)$ is defined and equal to 0, then $\varphi_e(y)$ is defined exactly when y is equal to 0, so $W_e = \{0\}$. If $\varphi_{f(e)}(0)$ is not defined, or is defined but not equal to 0, then $W_e = \emptyset$. Either way, $\varphi_{f(e)}$ is not the characteristic function of W_e , since it gives the wrong answer on input 0. $\square$

-NoValue-

[content/computability/computability-theory/def-functions-self-reference.tex](#)

30.23 Defining Functions using Self-Reference

cmp:thy:slf:
sec It is generally useful to be able to define functions in terms of themselves. For example, given computable functions k , l , and m , the fixed-point lemma tells us that there is a partial computable function f satisfying the following equation for every y :

$$f(y) \simeq \begin{cases} k(y) & \text{if } l(y) = 0 \\ f(m(y)) & \text{otherwise.} \end{cases}$$

Again, more specifically, f is obtained by letting

$$g(x, y) \simeq \begin{cases} k(y) & \text{if } l(y) = 0 \\ \varphi_x(m(y)) & \text{otherwise} \end{cases}$$

and then using the fixed-point lemma to find an index e such that $\varphi_e(y) = g(e, y)$.

For a concrete example, the “greatest common divisor” function $\text{gcd}(u, v)$ can be defined by

$$\text{gcd}(u, v) \simeq \begin{cases} v & \text{if } u = 0 \\ \text{gcd}(\text{mod}(v, u), u) & \text{otherwise} \end{cases}$$

where $\text{mod}(v, u)$ denotes the remainder of dividing v by u . An appeal to the fixed-point lemma shows that gcd is partial computable. (In fact, this can be put in the format above, letting y code the pair $\langle u, v \rangle$.) A subsequent induction on u then shows that, in fact, gcd is total.

Of course, one can cook up self-referential definitions that are much fancier than the examples just discussed. Most programming languages support definitions of functions in terms of themselves, one way or another. Note that this is a little bit less dramatic than being able to define a function in terms of an *index* for an algorithm computing the functions, which is what, in full generality, the fixed-point theorem lets you do.

Parte VI

Máquinas de Turing

Capítulo 31

Computações de Máquinas de Turing

`content/turing-machines/machines-computations/introduction.tex`

31.1 Introdução

O que significa dizer que uma função, digamos, de $\mathbb{N}$ para $\mathbb{N}$ é *computável*? Entre as primeiras respostas, e a mais conhecida, está a de que uma função é computável se puder ser computada por uma máquina de Turing. Essa noção foi apresentada por Alan Turing em 1936. As máquinas de Turing são um exemplo de *modelo de computação*—uma maneira matematicamente precisa de definir a ideia de um “procedimento computacional.” O que exatamente isso significa é debatido, mas há amplo consenso de que as máquinas de Turing são uma forma de especificar procedimentos computacionais. Embora o termo “máquina de Turing” evoque a imagem de uma máquina física com partes móveis, estritamente falando uma máquina de Turing é um construto puramente matemático e, como tal, idealiza a ideia de um procedimento computacional. Por exemplo, não impomos restrição alguma aos requisitos de tempo ou memória de uma máquina de Turing: máquinas de Turing podem computar algo mesmo que a computação exija mais espaço de armazenamento ou mais passos do que há átomos no universo.

tur:mac:int:
sec

explanation Talvez seja melhor pensar em uma máquina de Turing como um programa para um tipo especial de mecanismo imaginário. Esse mecanismo consiste em uma *fita* e uma *cabeça de leitura-escrita*. Na nossa versão das máquinas de Turing, a fita é infinita em uma direção (à direita) e é dividida em *quadrados*, cada um dos quais pode conter um símbolo de um *alfabeto* finito. Esses

31.1. INTRODUÇÃO

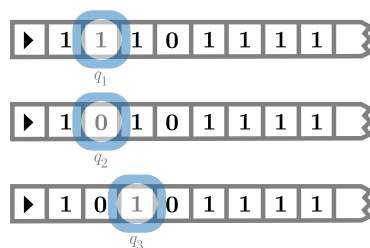


Figura 31.1: Uma máquina de Turing executando seu programa.

tur:mac:int:
fig:tm

alfabetos podem conter qualquer número de símbolos diferentes, mas nos contentaremos em geral com três: $\triangleright$, 0 e 1. Quando o mecanismo é iniciado, a fita está vazia (ou seja, cada quadrado contém o símbolo 0), exceto o quadrado mais à esquerda, que contém $\triangleright$, e um número finito de quadrados que contêm a *entrada*. A qualquer momento, o mecanismo está em um de um número finito de *estados*. No início, a cabeça examina o quadrado mais à esquerda e em um *estado inicial* especificado. A cada passo da execução do mecanismo, o conteúdo do quadrado atualmente examinado, junto com o estado em que o mecanismo se encontra e o programa da máquina de Turing, determinam o que acontece em seguida. O programa da máquina de Turing é dado por uma função parcial que recebe como entrada um estado q e um símbolo σ e produz uma tripla $\langle q', \sigma', D \rangle$. Sempre que o mecanismo está no estado q e lê o símbolo σ , substitui o símbolo no quadrado atual por σ' , a cabeça move-se à esquerda, à direita ou permanece no lugar conforme D seja L , R ou N , e o mecanismo passa ao estado q' .

Por exemplo, considere a situação em [Figura 31.1](#). A parte visível da fita da máquina de Turing contém o símbolo de fim de fita $\triangleright$ no quadrado mais à esquerda, seguido de três 1's, um 0 e mais quatro 1's. A cabeça está lendo o terceiro quadrado da esquerda, que contém um 1, e está no estado q_1 —dizemos que “a máquina está lendo um 1 no estado q_1 .” Se o programa da máquina de Turing retornar, para a entrada $\langle q_1, 1 \rangle$, a tripla $\langle q_2, 0, N \rangle$, a máquina substituiria o 1 no terceiro quadrado por um 0, deixaria a cabeça de leitura/escrita onde está e passaria ao estado q_2 . Se então o programa retornar $\langle q_3, 0, R \rangle$ para a entrada $\langle q_2, 0 \rangle$, a máquina sobrescreveria o 0 com outro 0 (efetivamente, deixando inalterado o conteúdo da fita sob a cabeça de leitura/escrita), moveria um quadrado à direita e entraria no estado q_3 . E assim por diante.

Dizemos que a máquina *para* quando encontra algum estado, q_n , e símbolo, σ , tais que não há instrução para $\langle q_n, \sigma \rangle$, ou seja, a função de transição para a entrada $\langle q_n, \sigma \rangle$ é indefinida. Em outras palavras, a máquina não tem instrução a executar e, nesse ponto, cessa a operação. A parada às vezes é representada por um estado de parada específico h . Isso será demonstrado com mais detalhes adiante.

A beleza do artigo de Turing, “On computable numbers,” está em que ele apresenta não só uma definição formal, mas também um argumento de que a

[digression](#)

definição captura a noção intuitiva de computabilidade. Pela definição, deve ficar claro que toda função computável por uma máquina de Turing é computável no sentido intuitivo. Turing oferece três tipos de argumento de que o inverso é verdadeiro, ou seja, que toda função que naturalmente consideraríamos computável é computável por tal máquina. São eles (nas palavras de Turing):

1. Um apelo direto à intuição.
2. Uma prova da equivalência de duas definições (caso a nova definição tenha maior apelo intuitivo).
3. Dar exemplos de grandes classes de números que são computáveis.

Nosso objetivo é tentar definir a noção de computabilidade “em princípio,” ou seja, sem levar em conta limitações práticas de tempo e espaço. Claro, com a definição mais ampla de computabilidade em vigor, pode-se então considerar computação com recursos limitados; isso forma o cerne do assunto conhecido como “complexidade computacional.”

Observações históricas Alan Turing inventou as máquinas de Turing em 1936. Embora seu interesse na época fosse a decidibilidade da lógica de primeira ordem, o artigo tem sido descrito como um trabalho definitivo sobre os fundamentos do projeto de computadores. No artigo, Turing concentra-se em números reais computáveis, ou seja, números reais cujas expansões decimais são computáveis; mas observa que não é difícil adaptar suas noções a funções computáveis sobre os números naturais, e assim por diante. Note que isso foi cinco anos inteiros antes de o primeiro computador de propósito geral em funcionamento ser construído em 1941 (pelo alemão Konrad Zuse na sala de estar dos pais), sete anos antes de Turing e seus colegas em Bletchley Park construírem o Colossus que quebrava códigos (1943), nove anos antes do ENIAC americano (1945), doze anos antes do primeiro computador britânico de propósito geral—a Manchester Small-Scale Experimental Machine—ser construída em Manchester (1948), e treze anos antes dos americanos testarem o BINAC (1949). A Manchester SSEM tem a distinção de ser o primeiro computador com programa armazenado—as máquinas anteriores tinham de ser recabeada manualmente para cada nova tarefa.

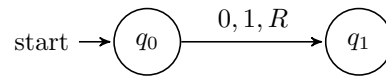
`content/turing-machines/machines-computations/representing-tms.tex`

31.2 Representando Máquinas de Turing

explanation As máquinas de Turing podem ser representadas visualmente por *diagramas de estado*. Os diagramas são compostos por células de estado conectadas por setas. Sem surpresa, cada célula de estado representa um estado da máquina. Cada seta representa uma instrução que pode ser executada a partir desse estado, com as especificações da instrução escritas acima ou abaixo da seta apropriada. tur:mac:rep: sec

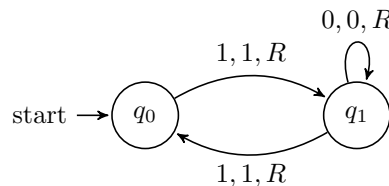
31.2. REPRESENTANDO MÁQUINAS DE TURING

Considere a seguinte máquina, que tem apenas dois estados internos, q_0 e q_1 , e uma instrução:



Lembre-se de que a máquina de Turing tem uma cabeça de leitura/escrita e uma fita com a entrada escrita nela. A instrução pode ser lida como *se lendo um 0 no estado q_0 , escreva um 1, mova-se à direita, e vá para o estado q_1* . Isto é equivalente à função de transição que mapeia $\langle q_0, 0 \rangle$ em $\langle q_1, 1, R \rangle$.

Exemplo 31.1. *Máquina Par:* A seguinte máquina de Turing para se, e somente se, houver um número par de 1 na fita (sob a suposição de que todos os 1 vêm antes do primeiro 0 na fita).



O diagrama de estado corresponde à seguinte função de transição:

$$\begin{aligned}\delta(q_0, 1) &= \langle q_1, 1, R \rangle, \\ \delta(q_1, 1) &= \langle q_0, 1, R \rangle, \\ \delta(q_1, 0) &= \langle q_1, 0, R \rangle\end{aligned}$$

A máquina acima para apenas quando a entrada é um número par de traços. Caso contrário, a máquina (teoricamente) continua a operar indefinidamente. Para qualquer máquina e entrada, é possível rastrear as *configurações* da máquina a fim de determinar a saída. Daremos uma definição formal de configurações mais adiante. Por enquanto, podemos pensar intuitivamente nas configurações como uma série de diagramas que mostram o estado da máquina em qualquer instante durante a operação. As configurações mostram o conteúdo da fita, o estado da máquina e a posição da cabeça de leitura/escrita.

[explanation](#)

Vamos rastrear as configurações da máquina par se ela for iniciada com uma entrada de quatro 1. Nesse caso, esperamos que a máquina pare. Em seguida, executaremos a máquina com uma entrada de três 1, na qual a máquina rodará para sempre.

A máquina inicia no estado q_0 , examinando o 1 mais à esquerda. Podemos representar o estado inicial da máquina da seguinte forma:

$$\triangleright 1_0 1110 \dots$$

A configuração acima é direta. Como se pode ver, a máquina inicia no estado um, examinando o 1 mais à esquerda. Isso é representado por um subscrito do

CAPÍTULO 31. COMPUTAÇÕES DE MÁQUINAS DE TURING

nome do estado no primeiro 1. A instrução aplicável neste ponto é $\delta(q_0, 1) = \langle q_1, 1, R \rangle$, e assim a máquina move-se à direita na fita e muda para o estado q_1 .

$\triangleright 11_1 110 \dots$

Como a máquina está agora no estado q_1 examinando um 1, temos que “seguir” a instrução $\delta(q_1, 1) = \langle q_0, 1, R \rangle$. Isso resulta na configuração

$\triangleright 111_0 10 \dots$

À medida que a máquina continua, as regras são aplicadas novamente na mesma ordem, resultando nas duas configurações a seguir:

$\triangleright 1111_1 0 \dots$

$\triangleright 11110_0 \dots$

A máquina está agora no estado q_0 examinando um 0. Com base no diagrama de transição, podemos ver facilmente que não há nenhuma instrução a ser executada, e portanto a máquina parou. Isso significa que a entrada foi aceita.

Suponha que, em seguida, iniciemos a máquina com uma entrada de três 1. As primeiras configurações são semelhantes, pois as mesmas instruções são executadas, com apenas uma pequena diferença na entrada da fita:

$\triangleright 1_0 110 \dots$

$\triangleright 11_1 10 \dots$

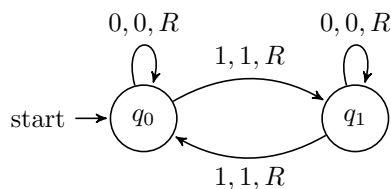
$\triangleright 111_0 0 \dots$

$\triangleright 1110_1 \dots$

A máquina já percorreu todos os 1 e está lendo um 0 no estado q_1 . Como mostrado no diagrama, há uma instrução da forma $\delta(q_1, 0) = \langle q_1, 0, R \rangle$. Como a fita está preenchida com 0 indefinidamente para a direita, a máquina continuará a executar esta instrução *para sempre*, permanecendo no estado q_1 e movendo-se cada vez mais à direita. A máquina nunca para, e não aceita a entrada.

explanation É importante observar que nem todas as máquinas param. Se parar significa que a máquina fica sem instruções a executar, então podemos criar uma máquina que nunca para, simplesmente garantindo que haja uma seta de saída para cada símbolo em cada estado. A máquina par pode ser modificada para rodar indefinidamente adicionando uma instrução para examinar um 0 em q_0 .

Exemplo 31.2.



31.2. REPRESENTANDO MÁQUINAS DE TURING

As tabelas de máquina são outra forma de representar máquinas de Turing. explanation
As tabelas de máquina têm o alfabeto da fita exibido no eixo x , e o conjunto de estados da máquina ao longo do eixo y . Dentro da tabela, na interseção de cada estado e símbolo, está escrito o resto da instrução—o novo estado, o novo símbolo e a direção do movimento. As tabelas de máquina facilitam determinar em que estado, e para qual símbolo, a máquina para. Sempre que há uma lacuna na tabela, há um ponto possível para a máquina parar. Ao contrário dos diagramas de estado e dos conjuntos de instruções, em que os pontos nos quais a máquina para nem sempre são imediatamente óbvios, quaisquer pontos de parada são rapidamente identificados ao encontrar as lacunas na tabela de máquina.

Exemplo 31.3. A tabela de máquina da máquina par é:

	0	1	$\triangleright$
q_0		$1, q_1, R$	
q_1	$0, q_1, R$	$1, q_0, R$	

Como podemos ver, a máquina para ao examinar um 0 no estado q_0 .

Até aqui consideramos apenas máquinas que leem e aceitam entradas. No explanation
entanto, as máquinas de Turing têm a capacidade de tanto ler quanto escrever. Um exemplo de tal máquina (embora haja muitos, muitos exemplos) é um *duplicador*. Um duplicador, quando iniciado com um bloco de n 1 na fita, produz como saída um bloco de $2n$ 1.

tur:mac:rep:
ex:doubler **Exemplo 31.4.** Antes de construir uma máquina duplicadora, é importante elaborar uma *estratégia* para resolver o problema. Como a máquina (tal como a formulamos) não consegue lembrar quantos 1 já leu, precisamos encontrar uma maneira de rastrear todos os 1 na fita. Uma dessas maneiras é separar a saída da entrada com um 0. A máquina pode então apagar o primeiro 1 da entrada, percorrer o resto da entrada, deixar um 0, e escrever dois novos 1. A máquina então voltará e encontrará o segundo 1 na entrada, e duplicará esse também. Para cada 1 de entrada, ela escreverá dois 1 de saída. Ao apagar a entrada conforme a máquina avança, podemos garantir que nenhum 1 seja perdido ou duplicado duas vezes. Quando toda a entrada é apagada, restarão $2n$ 1 na fita. O diagrama de estado da máquina de Turing resultante está representado na Figura 31.2.

Problema 31.1. Escolha uma entrada arbitrária e rastreie as configurações da máquina duplicadora em Exemplo 31.4.

Problema 31.2. Projete uma máquina de Turing com alfabeto $\{\triangleright, 0, A, B\}$ que aceite, isto é, pare, qualquer cadeia de A e B na qual o número de A seja igual ao número de B e todos os A precedam todos os B , e que rejeite, isto é, não pare, qualquer cadeia em que o número de A não seja igual ao número de B , ou os A não precedam todos os B . (Por exemplo, a máquina deve aceitar $AABB$, e $AAABBB$, mas rejeitar tanto AAB quanto $AABBAABB$.)

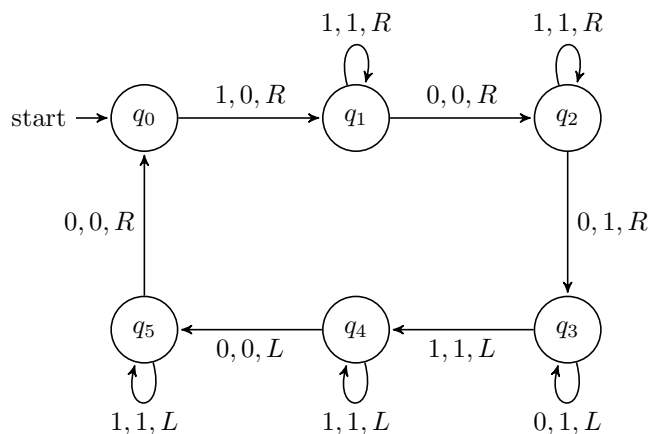


Figura 31.2: Uma máquina duplicadora

Problema 31.3. Projete uma máquina de Turing com alfabeto $\{\triangleright, 0, A, B\}$ que receba como entrada qualquer cadeia α de A e B e a duplique para produzir uma saída da forma $\alpha\alpha$. (Por exemplo, a entrada $ABBA$ deve resultar na saída $ABBAABBA$).

Problema 31.4. *Alfabética?*: Projete uma máquina de Turing com alfabeto $\{\triangleright, 0, A, B\}$ que, ao receber como entrada uma sequência finita de A e B , verifique se todos os A aparecem à esquerda de todos os B ou não. A máquina deve deixar a cadeia de entrada na fita, e ou parar, se a cadeia for “alfabética”, ou rodar para sempre, se a cadeia não for.

Problema 31.5. *Ordenador alfabético*: Projete uma máquina de Turing com alfabeto $\{\triangleright, 0, A, B\}$ que receba como entrada uma sequência finita de A e B e os reorganize de modo que todos os A fiquem à esquerda de todos os B . (Por exemplo, a sequência $BABAA$ deve tornar-se a sequência $AAABB$, e a sequência $ABBABB$ deve tornar-se a sequência $AABBBB$).

`content/turing-machines/machines-computations/turing-machines.tex`

31.3 Máquinas de Turing

explanation A definição formal do que constitui uma máquina de Turing parece abstrata, mas é na verdade simples: apenas reúne em uma estrutura matemática toda a informação necessária para especificar o funcionamento de uma máquina de Turing. Isso inclui (1) em quais estados a máquina pode estar, (2) quais

tur:mac:rep:
fig:doubler

tur:mac:tur:
sec

31.4. CONFIGURAÇÕES E COMPUTAÇÕES

símbolos são permitidos na fita, (3) em qual estado a máquina deve começar e (4) qual é o conjunto de instruções da máquina.

Definição 31.5 (Máquina de Turing). Uma *máquina de Turing* M é uma tupla $\langle Q, \Sigma, q_0, \delta \rangle$ consistindo de

1. um conjunto finito de *estados* Q ,
2. um *alfabeto* finito Σ que inclui $\triangleright$ e 0 ,
3. um *estado inicial* $q_0 \in Q$,
4. um *conjunto de instruções* finito $\delta: Q \times \Sigma \rightarrow Q \times \Sigma \times \{L, R, N\}$.

A função parcial δ também é chamada de *função de transição* de M .

Supomos que a fita é infinita em apenas uma direção. Por essa razão é útil designar um símbolo especial $\triangleright$ como marcador da extremidade esquerda da fita. Isso facilita que programas de máquinas de Turing saibam quando estão “em perigo” de sair da fita. Poderíamos supor que esse símbolo nunca é sobrescrito, ou seja, que $\delta(q, \triangleright) = \langle q', \triangleright, x \rangle$ se $\delta(q, \triangleright)$ estiver definido. Alguns livros-texto fazem isso; nós não. Basta ter cuidado ao construir sua máquina de Turing para que ela nunca sobrescreva $\triangleright$. Além disso, há casos em que permitir tal sobrescrita oferece alguma flexibilidade conveniente.

Exemplo 31.6. Máquina dos pares: A máquina dos pares é formalmente a quádrupla $\langle Q, \Sigma, q_0, \delta \rangle$ onde

$$\begin{aligned} Q &= \{q_0, q_1\} \\ \Sigma &= \{\triangleright, 0, 1\}, \\ \delta(q_0, 1) &= \langle q_1, 1, R \rangle, \\ \delta(q_1, 1) &= \langle q_0, 1, R \rangle, \\ \delta(q_1, 0) &= \langle q_1, 0, R \rangle. \end{aligned}$$

`content/turing-machines/machines-computations/configuration.tex`

31.4 Configurações e Computações

`cmp:tur:con:
sec`

Lembre-se de acompanhar as configurações da máquina dos pares anteriormente. O mecanismo imaginário formado pela fita, pela cabeça de leitura-escrita e pelo programa da máquina de Turing é, na verdade, apenas uma maneira intuitiva de visualizar o que é uma computação de máquina de Turing. Formalmente, podemos definir a computação de uma máquina de Turing sobre uma dada entrada como uma sequência de *configurações*—e uma configuração, por sua vez, é uma sequência de símbolos (correspondendo ao conteúdo da fita em um dado ponto da computação), um número que indica a posição da cabeça de leitura-escrita e um estado. Com isso, podemos definir o que a máquina de Turing M computa sobre uma dada entrada.

`explanation`

Definição 31.7 (Configuração). Uma *configuração* da máquina de Turing $M = \langle Q, \Sigma, q_0, \delta \rangle$ é uma tripla $\langle C, m, q \rangle$ onde

1. $C \in \Sigma^*$ é uma sequência finita de símbolos de Σ ,
2. $m \in \mathbb{N}$ é um número $< \text{len}(C)$, e
3. $q \in Q$

Intuitivamente, a sequência C é o conteúdo da fita (símbolos de todos os quadrados desde o quadrado mais à esquerda até o último não vazio ou previamente visitado), m é o número do quadrado que a cabeça de leitura-escrita está examinando (começando com 0 como o número do quadrado mais à esquerda), e q é o estado atual da máquina.

explanation A possível entrada para uma máquina de Turing é uma sequência de símbolos, geralmente uma sequência que codifica um número de alguma forma. A configuração inicial da máquina de Turing é aquela configuração em que iniciamos a máquina de Turing para trabalhar sobre essa entrada: a fita contém o marcador de fim de fita imediatamente seguido da entrada escrita nos quadrados à direita, a cabeça de leitura-escrita está examinando o quadrado mais à esquerda da entrada (ou seja, o quadrado à direita do marcador de fim esquerdo), e o mecanismo está no estado inicial designado q_0 .

Definição 31.8 (Configuração inicial). A *configuração inicial* de M para a entrada $I \in \Sigma^*$ é

$$\langle \triangleright \frown I, 1, q_0 \rangle.$$

explanation O símbolo $\frown$ denota *concatenação*—a cadeia de entrada começa imediatamente após o marcador de fim esquerdo.

Definição 31.9. Dizemos que uma configuração $\langle C, m, q \rangle$ *produz a configuração* $\langle C', m', q' \rangle$ *em um passo* (de acordo com M), sse

1. o m -ésimo símbolo de C é σ ,
2. o conjunto de instruções de M especifica $\delta(q, \sigma) = \langle q', \sigma', D \rangle$,
3. o m -ésimo símbolo de C' é σ' , e
4. a) $D = L$ e $m' = m - 1$ se $m > 0$, caso contrário $m' = 0$, ou
b) $D = R$ e $m' = m + 1$, ou
c) $D = N$ e $m' = m$,
5. se $m' = \text{len}(C)$, então $\text{len}(C') = \text{len}(C) + 1$ e o m' -ésimo símbolo de C' é 0. Caso contrário $\text{len}(C') = \text{len}(C)$.
6. para todo i tal que $i < \text{len}(C)$ e $i \neq m$, $C'(i) = C(i)$,

31.5. REPRESENTAÇÃO UNÁRIA DE NÚMEROS

cmp:tur:con:
defn:run-output

Definição 31.10. Uma *execução de M sobre a entrada I* é uma sequência C_i de configurações de M , onde C_0 é a configuração inicial de M para a entrada I , e cada C_i produz C_{i+1} em um passo.

Dizemos que M *para sobre a entrada I após k passos* se $C_k = \langle C, m, q \rangle$, o m -ésimo símbolo de C é σ , e $\delta(q, \sigma)$ é indefinido. Nesse caso, a *saída* de M para a entrada I é O , onde O é uma cadeia de símbolos que não termina em 0 tal que $C = \triangleright \frown O \frown 0^j$ para algum $j \in \mathbb{N}$. (0^j é uma sequência de j 0's.)

De acordo com esta definição, a saída O de M sempre termina em um símbolo diferente de 0, ou, se no instante k toda a fita estiver preenchida com 0 (exceto o $\triangleright$ mais à esquerda), O é a cadeia vazia. explanation

content/turing-machines/machines-computations/unary-numbers.tex

31.5 Representação Unária de Números

tur:mac:una:
sec

As máquinas de Turing trabalham sobre sequências de símbolos escritos em sua fita. Dependendo do alfabeto que uma máquina de Turing usa, essas sequências de símbolos podem representar diversas entradas e saídas. De particular interesse, é claro, estão as máquinas de Turing que computam funções *aritméticas*, isto é, funções de números naturais. Uma maneira simples de representar inteiros positivos é codificá-los como sequências de um único símbolo 1. Se $n \in \mathbb{N}$, seja 1^n a sequência vazia se $n = 0$, e caso contrário a sequência composta de exatamente n 1's. explanation

Definição 31.11 (Computação). Uma máquina de Turing M *computa* a função $f: \mathbb{N}^k \rightarrow \mathbb{N}$ sse M para com a entrada

$$1^{n_1}01^{n_2}0 \dots 01^{n_k}$$

produzindo a saída $1^{f(n_1, \dots, n_k)}$.

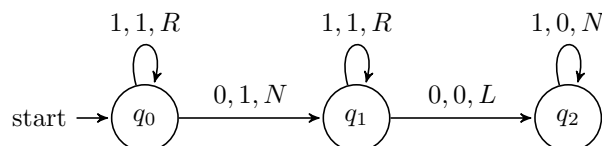
Problema 31.6. Dê uma definição para quando uma máquina de Turing M computa a função $f: \mathbb{N}^k \rightarrow \mathbb{N}^m$.

tur:mac:una:
ex:adder

Exemplo 31.12. *Adição:* Vamos construir uma máquina que computa a função $f(n, m) = n + m$. Isso requer uma máquina que comece com dois blocos de 1 de comprimento n e m na fita, e pare com um único bloco composto de $n + m$ 1's. Os dois blocos de entrada de 1 estão separados por um 0, então um método seria escrever um traço no quadrado que contém o 0, e apagar o último 1.

Problema 31.7. Rastreie as configurações da máquina de **Exemplo 31.12** para a entrada $\langle 3, 2 \rangle$. O que acontece se a máquina computa $0 + 0$?

Em **Exemplo 31.4**, demos um exemplo de uma máquina de Turing que recebe como entrada uma sequência de 1 e para com uma sequência de duas explanation

Figura 31.3: Uma máquina que computa $f(x, y) = x + y$

vezes mais 1 na fita—a máquina duplicadora. No entanto, como a saída contém 0 à esquerda do bloco duplicado de 1, ela não computa de fato a função $f(x) = 2x$, como você poderia ter suposto. Descreveremos duas maneiras de corrigir isso.

Exemplo 31.13. A máquina em Figura 31.4 computa a função $f(x) = 2x$. Em vez de apagar a entrada e escrever dois 1 na extremidade direita para cada 1 da entrada, como faz a máquina de Exemplo 31.4, esta máquina adiciona um único 1 à direita para cada 1 da entrada. Ela precisa rastrear onde a entrada termina, então deixa um 0 entre a entrada e os traços adicionados, que ela preenche com um 1 bem ao final. E temos que “lembrar” onde estamos na entrada, então substituímos temporariamente um 1 do bloco de entrada por um 0.

Exemplo 31.14. Uma segunda possibilidade para computar $f(x) = 2x$ é manter a máquina duplicadora original, mas adicionar estados e instruções ao final que movam o bloco duplicado de traços para a extremidade esquerda da fita. A máquina em Figura 31.5 faz exatamente esta última parte: iniciada em uma fita composta de um bloco de 0 seguido de um bloco de 1 (e com a cabeça posicionada em qualquer lugar do bloco de 0), ela apaga os 1 um a um e os escreve no início da fita. Para conseguir saber quando ela termina, primeiro marca o fim do bloco de 1 com um símbolo $\triangleright$, que é apagado ao final. Começamos a numerar os estados em q_6 , de modo que possam ser adicionados à máquina duplicadora. Tudo de que você precisará é uma instrução adicional $\delta(q_0, 0) = \langle q_6, 0, N \rangle$, isto é, uma seta de q_0 para q_6 rotulada $0, 0, N$. (Há um problema sutil: a máquina resultante não funciona para a entrada $x = 0$. Deixaremos isso como exercício.)

Problema 31.8. Em Exemplo 31.14 descrevemos uma máquina composta de uma combinação da máquina duplicadora de Figura 31.4 e da máquina move-dora de Figura 31.5. O que acontece se você inicia essa máquina combinada com a entrada $x = 0$, isto é, em uma fita vazia? Como você corrigiria a máquina de modo que, nesse caso, ela parasse com saída $2x = 0$? (Você deve conseguir fazer isso adicionando um estado e uma transição.)

31.5. REPRESENTAÇÃO UNÁRIA DE NÚMEROS

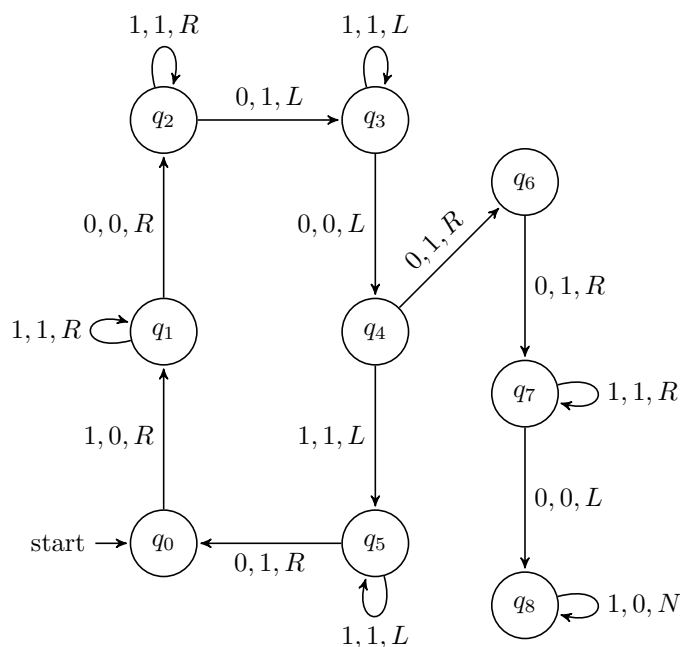


Figura 31.4: Uma máquina que computa $f(x) = 2x$

tur:mac:una:
fig:doubler-disc

Problema 31.9. Subtração: Projete uma máquina de Turing que, ao receber uma entrada de duas cadeias não vazias de traços de comprimento n e m , em que $n > m$, compute a função $f(n, m) = n - m$.

Problema 31.10. Igualdade: Projete uma máquina de Turing para computar a seguinte função:

$$\text{igualdade}(n, m) = \begin{cases} 1 & \text{se } n = m \\ 0 & \text{se } n \neq m \end{cases}$$

em que n e $m \in \mathbb{Z}^+$.

Problema 31.11. Projete uma máquina de Turing para computar a função $\min(x, y)$, em que x e y são inteiros positivos representados na fita por cadeias de 1 separadas por um 0. Você pode usar símbolos adicionais no alfabeto da máquina.

A função $\min$ seleciona o menor valor entre seus argumentos, de modo que $\min(3, 5) = 3$, $\min(20, 16) = 16$, e $\min(4, 4) = 4$, e assim por diante.

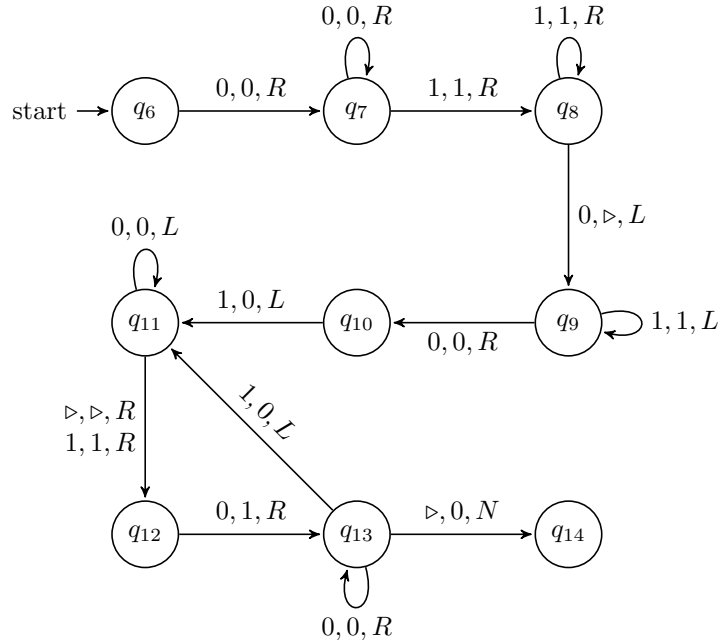


Figura 31.5: Movendo um bloco de 1 para a esquerda

Definição 31.15. Uma máquina de Turing M computa a função parcial $f: \mathbb{N}^k \rightarrow \mathbb{N}$ se,

1. M para com a entrada $1^{n_1} \frown 0 \frown \dots \frown 0 \frown 1^{n_k}$ produzindo a saída 1^m se $f(n_1, \dots, n_k) = m$.
2. M não para de modo algum, ou para com uma saída que não é um único bloco de 1 se $f(n_1, \dots, n_k)$ é indefinida.

[content/turing-machines/machines-computations/halting-states.tex](#)

31.6 Estados de Parada

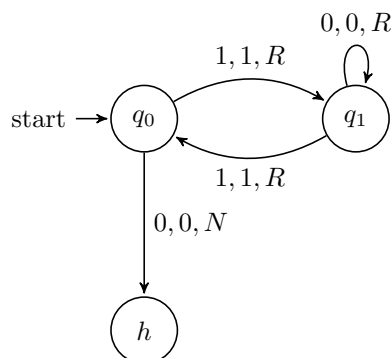
Embora tenhamos definido nossas máquinas para parar apenas quando não há instrução a executar, representações comuns de máquinas de Turing têm um *estado de parada* dedicado h , de modo que $h \in Q$.

A ideia por trás de um estado de parada é simples: quando a máquina terminou a operação (está pronta para aceitar a entrada, ou terminou de escrever

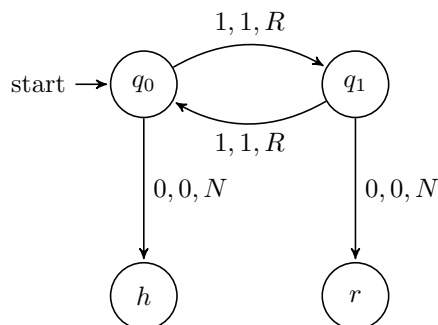
31.6. ESTADOS DE PARADA

a saída), ela entra em um estado h onde para. Algumas máquinas têm dois estados de parada, um que aceita a entrada e outro que rejeita a entrada.

Exemplo 31.16. *Estados de parada.* Para elucidar esse conceito, comecemos com uma alteração da máquina dos pares. Em vez de fazer a máquina parar no estado q_0 se a entrada for par, podemos adicionar uma instrução para enviar a máquina a um estado de parada.



Vamos expandir ainda mais o exemplo. Quando a máquina determina que a entrada é ímpar, ela nunca para. Podemos alterar a máquina para incluir um estado de *rejeição* substituindo a instrução em laço por uma instrução para ir a um estado de rejeição r .



Adicionar um estado de parada dedicado pode ser vantajoso em casos como [este](#), em que deixa explícito quando a máquina aceita/rejeita certas entradas. No entanto, é importante notar que nenhum poder computacional é ganho ao adicionar um estado de parada dedicado. Da mesma forma, uma noção menos formal de parada tem suas próprias vantagens. A definição de parada usada até agora neste capítulo torna a prova do *Problema da Parada* intuitiva e fácil de demonstrar. Por essa razão, continuamos com nossa definição original.

31.7 Máquinas Disciplinadas

explanation Na **Seção 31.6**, consideramos máquinas de Turing que têm um único estado de parada designado h —tais máquinas são garantidas de parar, se pararem de fato, no estado h . Dessa forma, máquinas com um único estado de parada são mais “disciplinadas” do que permitimos que máquinas de Turing sejam em geral. Há outras restrições que podemos impor ao comportamento das máquinas de Turing. Por exemplo, também não proibimos máquinas de Turing de apagar o marcador de fim de fita no quadrado 0, ou de tentar mover-se à esquerda a partir do quadrado 0. (Nossa definição afirma que a cabeça simplesmente permanece no quadrado 0 nesse caso; outras definições fazem a máquina parar.) Da mesma forma, às vezes é desejável poder supor que uma máquina de Turing, se parar de fato, para no quadrado 1. tur:mac:dis: sec

Definição 31.17. Uma máquina de Turing M é *disciplinada* sse tur:mac:dis: defn:disciplined

1. ela tem um único estado de parada designado h ,
2. ela para, se parar de fato, enquanto examina o quadrado 1,
3. ela nunca apaga o símbolo $\triangleright$ no quadrado 0, e
4. ela nunca tenta mover-se à esquerda a partir do quadrado 0.

explanation Já discutimos que qualquer máquina de Turing pode ser transformada em uma com o mesmo comportamento, mas com um estado de parada designado. Isso é feito simplesmente adicionando um novo estado h , e adicionando uma instrução $\delta(q, \sigma) = \langle h, \sigma, N \rangle$ para qualquer par $\langle q, \sigma \rangle$ onde o δ original é indefinido. É verdade, embora tedioso de provar, que qualquer máquina de Turing M pode ser transformada em uma máquina de Turing disciplinada M' que para nas mesmas entradas e produz a mesma saída. Por exemplo, se a máquina de Turing para e não está no quadrado 1, podemos adicionar algumas instruções para fazer a cabeça mover-se à esquerda até encontrar o marcador de fim de fita, depois mover um quadrado à direita, e então parar. Deixamos a você pensar sobre como as outras condições podem ser tratadas.

Exemplo 31.18. Em **Figura 31.6**, transformamos a máquina de adição de **Exemplo 31.12** em uma máquina disciplinada.

Proposição 31.19. *Para toda máquina de Turing M , há uma máquina de Turing disciplinada M' que para com saída O se M para com saída O , e não para se M não para. Em particular, qualquer função $f: \mathbb{N}^n \rightarrow \mathbb{N}$ computável por uma máquina de Turing também é computável por uma máquina de Turing disciplinada.* tur:mac:dis: prop:disciplined

Problema 31.12. Dê uma máquina disciplinada que computa $f(x) = x + 1$.

Problema 31.13. Encontre uma máquina disciplinada que, quando iniciada com entrada 1^n produz saída $1^n \frown 0 \frown 1^n$.

31.8. COMBINANDO MÁQUINAS DE TURING

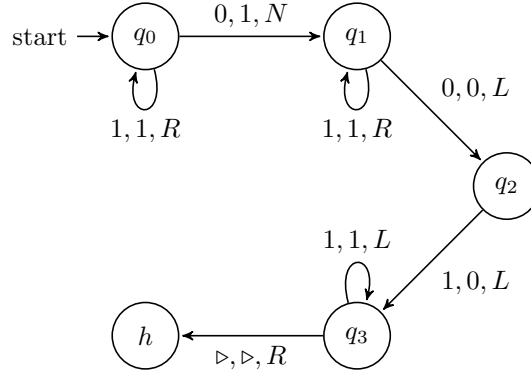


Figura 31.6: Uma máquina de adição disciplinada

tur:mac:dis:
fig:adder-disc

content/turing-machines/machines-computations/combining-machines.tex

31.8 Combinando Máquinas de Turing

tur:mac:emb:
sec

Os exemplos de máquinas de Turing que vimos até agora foram bastante simples por natureza. Mas, de fato, qualquer problema que possa ser resolvido com qualquer linguagem de programação moderna também pode ser resolvido com máquinas de Turing. Para construir máquinas de Turing mais complexas, é importante nos convencer de que podemos combiná-las, de modo a construir máquinas que resolvam problemas mais complexos quebrando o procedimento em partes mais simples. Se conseguirmos encontrar uma maneira natural de quebrar um problema complexo em partes constituintes, podemos atacar o problema em vários estágios, criando várias máquinas de Turing simples e combinando-as em uma única máquina capaz de resolver o problema. Esse ponto é especialmente importante ao atacar o Problema da Parada na próxima seção.

explanation

Como combinamos as máquinas de Turing $M = \langle Q, \Sigma, q_0, \delta \rangle$ e $M' = \langle Q', \Sigma', q'_0, \delta' \rangle$? Usamos agora a configuração da fita após M ter parado como a configuração de entrada de uma execução da máquina M' . Para obter uma única máquina de Turing $M \frown M'$ que faz isso, faça o seguinte:

1. Renumere (ou rerrotule) todos os estados Q' de M' de modo que M e M' não tenham estados em comum ($Q \cap Q' = \emptyset$).
2. Os estados de $M \frown M'$ são $Q \cup Q'$.
3. O alfabeto da fita é $\Sigma \cup \Sigma'$.

CAPÍTULO 31. COMPUTAÇÕES DE MÁQUINAS DE TURING

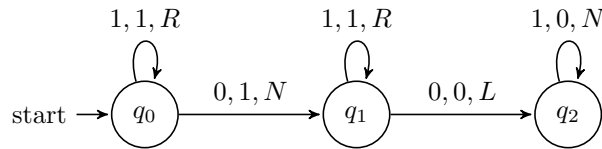
4. O estado inicial é q_0 .
5. A função de transição é a função δ'' dada por:

$$\delta''(q, \sigma) = \begin{cases} \delta(q, \sigma) & \text{se } q \in Q \\ \delta'(q, \sigma) & \text{se } q \in Q' \\ \langle q'_0, \sigma, N \rangle & \text{se } q \in Q \text{ e } \delta(q, \sigma) \text{ é indefinida} \end{cases}$$

A máquina resultante usa as instruções de M quando está em um estado $q \in Q$, e as instruções de M' quando está em um estado $q \in Q'$. Quando está em um estado $q \in Q$ e examina um símbolo σ para o qual M não tem transição (isto é, M teria parado), ela entra no estado inicial de M' (e deixa o conteúdo da fita e a posição da cabeça como estão).

Note que, a menos que a máquina M seja disciplinada, não sabemos onde está a cabeça da fita quando M para, então a configuração de parada de M não precisa ter a cabeça examinando o quadrado 1. Ao combinar máquinas, é importante ter isso em mente.

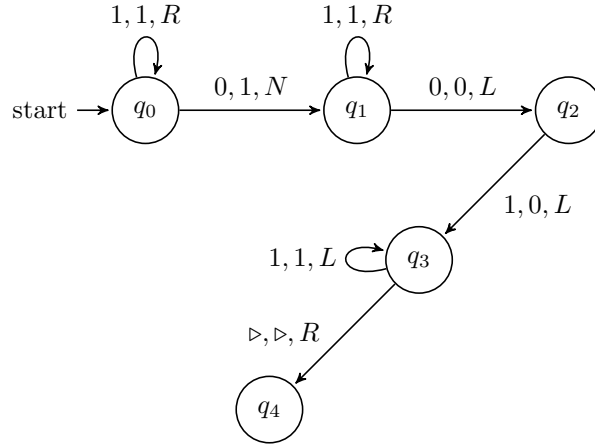
Exemplo 31.20. Combinando Máquinas: Projetaremos uma máquina que, quando iniciada com uma entrada composta de dois blocos de 1 de comprimento n e m , para com um único bloco de $2(m + n)$ 1 na fita. Para construir essa máquina, podemos combinar duas máquinas que já conhecemos: a máquina de adição e a duplicadora. Começamos desenhando um diagrama de estado para a máquina de adição.



Em vez de parar no estado q_2 , queremos continuar a operação a fim de duplicar a saída. Lembre-se de que a máquina duplicadora apaga o primeiro traço da entrada e escreve dois traços em uma saída separada. Vamos adicionar uma instrução para garantir que a cabeça da fita esteja lendo o primeiro traço da

31.9. VARIANTES DE MÁQUINAS DE TURING

saída da máquina de adição.



Agora é fácil duplicar a entrada—tudo o que temos a fazer é conectar a máquina duplicadora ao estado q_4 . Isso requer renomear os estados da máquina duplicadora de modo que comecem em q_4 em vez de q_0 —dessa forma não acabamos com dois estados iniciais. O diagrama final deve ter a aparência da **Figura 31.7**.

Proposição 31.21. *Se M e M' são disciplinadas e computam as funções $f: \mathbb{N}^k \rightarrow \mathbb{N}$ e $f': \mathbb{N} \rightarrow \mathbb{N}$, respectivamente, então $M \frown M'$ é disciplinada e computa $f' \circ f$.*

Prova. Como M é disciplinada, quando para com saída $f(n_1, \dots, n_k) = m$, a cabeça está examinando o quadrado 1. Se agora entrarmos no estado inicial de M' , a máquina parará com saída $f'(m)$, novamente examinando o quadrado 1. As demais condições de **Definição 31.17** também são satisfeitas. $\square$

Problema 31.14. Dê uma máquina de Turing disciplinada que computa $f(x) = x + 2$ tomando a máquina M de **Problema 31.12** e construindo $M \frown M$.

`content/turing-machines/machines-computations/variants.tex`

31.9 Variantes de Máquinas de Turing

tur:mac:var:
sec

Na verdade, há muitas maneiras possíveis de definir máquinas de Turing, das quais a nossa é apenas uma. De certas formas, nossa definição é mais liberal do que outras. Permitimos alfabetos finitos arbitrários; uma definição mais restrita poderia permitir apenas dois símbolos de fita, 1 e 0. Permitimos que a máquina escreva um símbolo na fita e se mova ao mesmo tempo; outras definições permitem escrever ou mover. Permitimos a possibilidade de escrever

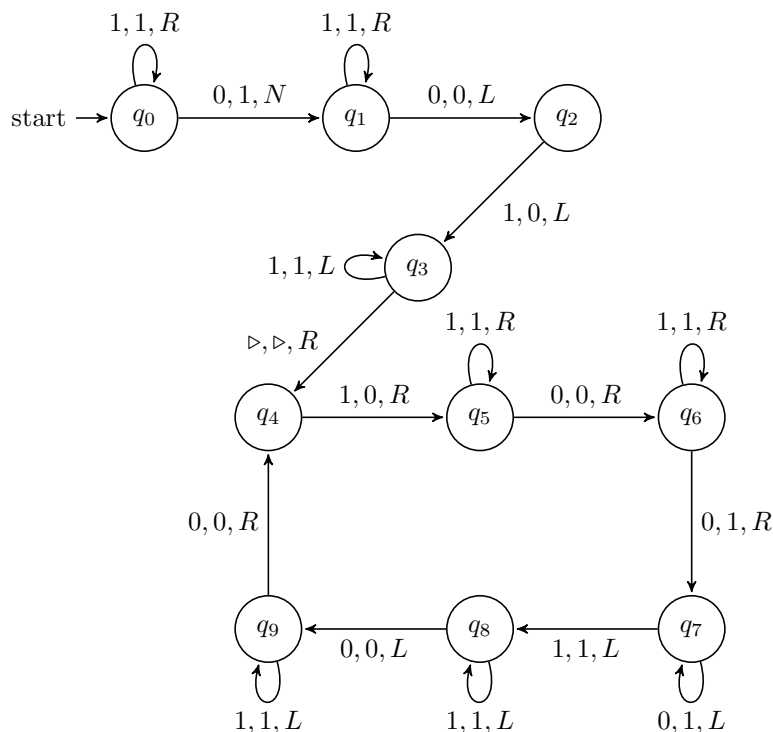


Figura 31.7: Combinando as máquinas de adição e duplicadora

tur:mac:cmb:
fig:combined

sem mover a cabeça da fita; outras definições omitem a “instrução” N . Em outros aspectos, nossa definição é mais restritiva. Assumimos que a fita é infinita em apenas uma direção; outras definições permitem que a fita seja infinita tanto para a esquerda quanto para a direita. Na verdade, pode-se até mesmo permitir qualquer número de fitas separadas, ou até mesmo uma grade infinita de quadrados. Representamos o conjunto de instruções da máquina de Turing por uma função de transição; outras definições usam uma relação de transição em que a máquina tem mais de uma instrução possível em qualquer situação dada.

Essa última flexibilização da definição é particularmente interessante. Em nossa definição, quando a máquina está no estado q lendo o símbolo σ , $\delta(q, \sigma)$ determina qual é o novo símbolo, o novo estado e a nova posição da cabeça da fita. Mas se permitirmos que o conjunto de instruções seja uma relação entre pares estado-símbolo atuais $\langle q, \sigma \rangle$ e novas triplas estado-símbolo-direção $\langle q', \sigma', D \rangle$, a ação da máquina de Turing pode não ser exclusivamente determinada — a relação de instrução pode conter tanto $\langle q, \sigma, q', \sigma', D \rangle$ quanto

31.10. A TESE DE CHURCH–TURING

$\langle q, \sigma, q'', \sigma'', D' \rangle$. Nesse caso, temos uma máquina de Turing *não determinística*. Estas desempenham um papel importante na teoria da complexidade computacional.

Há também diferentes convenções para quando uma máquina de Turing para: dizemos que ela para quando a função de transição é indefinida; outras definições exigem que a máquina esteja em um estado de parada especial designado. Explicamos na [Seção 31.6](#) por que exigir um estado de parada designado não é uma restrição que impacta o que as máquinas de Turing podem calcular. Como as fitas de nossas máquinas de Turing são infinitas em apenas uma direção, há casos em que uma máquina de Turing não consegue executar adequadamente uma instrução: se ela lê o quadrado mais à esquerda e deveria se mover para a esquerda. De acordo com a nossa definição, ela simplesmente permanece no lugar em vez de “cair”, mas poderíamos tê-la definido de modo que ela pare quando isso acontecer. Essa definição também é equivalente: poderíamos simular o comportamento de uma máquina de Turing que para quando tenta se mover para a esquerda do quadrado 0 excluindo cada transição $\delta(q, \triangleright) = \langle q', \sigma, L \rangle$ —então, em vez de tentar mover para a esquerda em $\triangleright$, a máquina para.¹

Há também diferentes maneiras de representar números (e, portanto, a função de entrada-saída calculada por uma máquina de Turing): usamos representação unária, mas você também pode usar representação binária. Isso requer dois símbolos além de 0 e $\triangleright$.

Agora, aqui está um fato interessante: nenhuma dessas variações importa para quais funções são Turing-computáveis. *Se uma função é Turing-computável segundo uma definição, ela é Turing-computável segundo todas elas.*

Não entraremos nos detalhes de como verificar isso. Eis apenas um exemplo: não ganhamos nenhum poder computacional adicional ao permitir uma fita que é infinita em ambas as direções, ou múltiplas fitas. A razão é, aproximadamente, que uma máquina de Turing com uma única fita infinita unidirecional pode simular múltiplas fitas ou fitas infinitas bidirecionais. Por exemplo, usando estados e instruções adicionais, podemos “traduzir” um programa para uma máquina com múltiplas fitas ou fita infinita bidirecional em um programa para uma máquina com uma única fita infinita unidirecional. A máquina traduzida pode usar os quadrados pares para os quadrados da fita 1 (ou os quadrados “positivos” de uma fita infinita bidirecional) e os quadrados ímpares para os quadrados da fita 2 (ou os quadrados “negativos”).

`content/turing-machines/machines-computations/church-turing-thesis.tex`

31.10 A Tese de Church–Turing

tur:mac:ctt:
sec

¹Isso não funciona *muito bem*, pois nada nos impede de escrever e ler $\triangleright$ em quadrados que não sejam o quadrado 0 (veja [Exemplo 31.14](#)). Podemos contornar isso adicionando um segundo símbolo $\triangleright'$ para usar em seu lugar para tal propósito.

As máquinas de Turing devem ser um substituto preciso do conceito de procedimento efetivo. Turing pensava que quem compreendesse tanto o conceito de procedimento efetivo quanto o de máquina de Turing teria a intuição de que tudo o que pudesse ser feito por meio de um procedimento efetivo poderia ser feito por uma máquina de Turing. Essa afirmação é sustentada pelo fato de que todas as outras propostas precisas de substituto do conceito de procedimento efetivo se mostram extensionalmente equivalentes ao conceito de máquina de Turing —ou seja, podem computar exatamente o mesmo conjunto de funções. Essa afirmação chama-se *tese de Church–Turing*.

Definição 31.22 (Tese de Church–Turing). A *tese de Church–Turing* afirma que tudo o que é computável por meio de um procedimento efetivo é computável por máquina de Turing.

A tese de Church–Turing é invocada de duas maneiras. O primeiro tipo de uso é uma desculpa para a preguiça. Suponha que temos uma descrição de um procedimento efetivo para computar algo, digamos, em “pseudocódigo.” Então podemos invocar a tese de Church–Turing para justificar a afirmação de que a mesma função é computada por alguma máquina de Turing, mesmo que não a tenhamos de fato construído.

O outro uso da tese de Church–Turing é mais filosoficamente interessante. Pode-se mostrar que existem funções que não podem ser computadas por máquinas de Turing. Disso, usando a tese de Church–Turing, conclui-se que não podem ser computadas efetivamente, usando qualquer procedimento que seja. Pois se houvesse tal procedimento, pela tese de Church–Turing seguiria que haveria uma máquina de Turing para ele. Assim, se podemos provar que não há máquina de Turing que o compute, também não pode haver um procedimento efetivo. Em particular, a tese de Church–Turing é invocada para afirmar que o chamado problema da parada não só não pode ser resolvido por máquinas de Turing, como não pode ser resolvido efetivamente de modo algum.

Capítulo 32

Indecidibilidade

32.1. INTRODUÇÃO

32.1 Introdução

tur:und:int:
sec

Pode parecer óbvio que nem toda função, nem mesmo toda função aritmética, pode ser computável. Há simplesmente um número grande demais delas, cujo comportamento é complicado demais. Funções definidas a partir do decaimento de partículas radioativas, por exemplo, ou de outro comportamento caótico ou aleatório. Suponha que comecemos a contar intervalos de 1 segundo a partir de um dado instante e definamos a função $f(n)$ como o número de partículas no universo que decaem no n -ésimo intervalo de 1 segundo após aquele momento inicial. Esta parece ser uma candidata a função que jamais poderemos ter esperança de computar.

Mas uma coisa é não conseguir imaginar como se computaria tais funções, e outra bem diferente é de fato provar que elas são incomputáveis. Na verdade, mesmo funções que parecem irremediavelmente complicadas podem, em um sentido abstrato, ser computáveis. Por exemplo, suponha que o universo seja finito no tempo—algum dia, num futuro muito distante, o universo se contrairá em um único ponto, como preveem algumas teorias cosmológicas. Então há apenas um número finito (mas incrivelmente grande) de segundos a partir daquele momento inicial para os quais $f(n)$ está definida. E qualquer função que esteja definida para apenas finitas entradas é computável: poderíamos listar as saídas em uma grande tabela, ou codificá-la em um diagrama de transição de estados de máquina de Turing muito grande.

Frequentemente estamos interessados em casos especiais de funções cujos valores fornecem as respostas a perguntas de sim/não. Por exemplo, a pergunta “ n é um número primo?” está associada à função

$$\text{primo}(n) = \begin{cases} 1 & \text{se } n \text{ é primo} \\ 0 & \text{caso contrário.} \end{cases}$$

Dizemos que uma pergunta de sim/não pode ser *efetivamente decidida* se a função associada de valores 1/0 é efetivamente computável.

Para provar matematicamente que há funções que não podem ser efetivamente computadas, ou problemas que não podem ser efetivamente decididos, é essencial fixar um modelo específico de computação e mostrar que há funções que ele não pode computar ou problemas que ele não pode decidir. Podemos mostrar, por exemplo, que nem toda função pode ser computada por máquinas de Turing, e nem todo problema pode ser decidido por máquinas de Turing. Podemos então apelar para a tese de Church-Turing para concluir que não apenas as máquinas de Turing não são poderosas o suficiente para computar toda função, como também nenhum procedimento efetivo o é.

A chave para provar tais resultados negativos é o fato de que podemos atribuir números às próprias máquinas de Turing. A maneira mais fácil de fazer isso é enumerá-las, talvez fixando um modo específico de escrever máquinas de Turing e seus programas e, em seguida, listando-os de maneira sistemática. Uma vez que vejamos que isso pode ser feito, então a existência de funções Turing-incomputáveis decorre de simples considerações de cardinalidade: o

conjunto de funções de $\mathbb{N}$ a $\mathbb{N}$ (na verdade, mesmo apenas de $\mathbb{N}$ a $\{0,1\}$) é **não enumerável**, mas como podemos enumerar todas as máquinas de Turing, o conjunto de funções Turing-computáveis é apenas **infinito enumerável**.

Também podemos definir funções e problemas *específicos* que podemos provar ser incomputáveis e indecidíveis, respectivamente. Um desses problemas é o assim chamado *Problema da Parada*. Máquinas de Turing podem ser finitamente descritas listando-se suas instruções. Tal descrição de uma máquina de Turing, isto é, um programa de máquina de Turing, pode é claro ser usada como entrada para outra máquina de Turing. Assim, podemos considerar máquinas de Turing que decidem perguntas sobre outras máquinas de Turing. Uma pergunta particularmente interessante é esta: “A máquina de Turing dada eventualmente para quando iniciada na entrada n ?” Seria bom se houvesse uma máquina de Turing que pudesse decidir essa pergunta: pense nela como uma máquina de Turing de controle de qualidade que garante que as máquinas de Turing não fiquem presas em laços infinitos e coisas do tipo. O fato interessante, que Turing provou, é que não pode haver tal máquina de Turing. Não pode haver uma única máquina de Turing que, quando iniciada em uma entrada que consiste em uma descrição de uma máquina de Turing M e algum número n , sempre pare com saída 1 ou 0 conforme a máquina M teria parado ou não quando iniciada na entrada n .

Uma vez que tenhamos exemplos de problemas indecidíveis específicos, podemos usá-los para mostrar que outros problemas também são indecidíveis. Por exemplo, um célebre problema indecidível é a pergunta “A **fórmula** φ de primeira ordem é válida?”. Não há máquina de Turing que, dada como entrada uma **fórmula** φ de primeira ordem, tenha garantia de parar com saída 1 ou 0 conforme φ seja válida ou não. Historicamente, a questão de encontrar um procedimento para resolver efetivamente esse problema foi chamada simplesmente de “o” problema da decisão; e assim dizemos que o problema da decisão é insolúvel. Turing e Church provaram esse resultado independentemente mais ou menos na mesma época, de modo que ele também é chamado de Teorema de Church-Turing.

`content/turing-machines/undecidability/enumerating-tms.tex`

32.2 Enumerando Máquinas de Turing

explanation Podemos mostrar que o conjunto de todas as máquinas de Turing é **enumerável**. tur:und:enu:sec Isso decorre do fato de que cada máquina de Turing pode ser finitamente descrita. O conjunto de estados e o vocabulário da fita são conjuntos finitos. A função de transição é uma função parcial de $Q \times \Sigma$ a $Q \times \Sigma \times \{L, R, N\}$ e, portanto, também pode ser especificada listando-se seus valores para os finitos pares de argumentos para os quais está definida.

Isso é verdade até certo ponto, mas há uma diferença sutil. A definição de máquinas de Turing não fez restrição alguma sobre quais **membros** o conjunto de estados e o alfabeto da fita podem ter. Assim, por exemplo, para todo

32.2. ENUMERANDO MÁQUINAS DE TURING

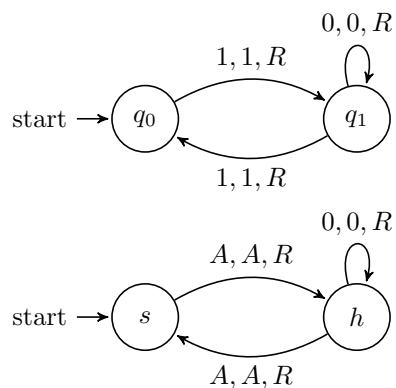


Figura 32.1: Variantes da máquina *Par*

tur:und:enu:
fig:variants

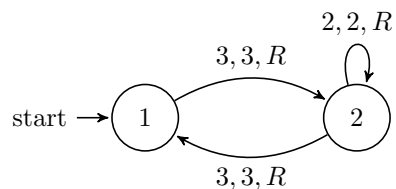


Figura 32.2: Uma máquina *Par* padrão

tur:und:enu:
fig:standard-even

número real, há tecnicamente uma máquina de Turing que usa esse número como um estado. Contudo, o *comportamento* da máquina de Turing é independente de quais objetos servem como estados e vocabulário. Considere as duas máquinas de Turing na [Figura 32.1](#). Esses dois diagramas correspondem a duas máquinas, M com o alfabeto de fita $\Sigma = \{\triangleright, 0, 1\}$ e conjunto de estados $\{q_0, q_1\}$, e M' com alfabeto $\Sigma' = \{\triangleright, 0, A\}$ e estados $\{s, h\}$. Mas, quanto ao resto, suas instruções são as mesmas: M para em uma sequência de n 1 se, e somente se, n é par, e M' para em uma sequência de n A se, e somente se, n é par. Tudo o que fizemos foi renomear 1 para A , q_0 para s e q_1 para h . Esse exemplo se generaliza: podemos considerar máquinas de Turing como sendo a mesma desde que uma resulte da outra por tal renomeação de símbolos e estados. Na verdade, podemos simplesmente pensar nos símbolos e estados de uma máquina de Turing como inteiros positivos: em vez de σ_0 pense 1, em vez de σ_1 pense 2, etc.; $\triangleright$ é 1, 0 é 2, etc. Dessa forma, a máquina *Par* torna-se a máquina representada na [Figura 32.2](#). Poderíamos chamar de máquina *padrão* uma máquina de Turing cujos estados e símbolos são inteiros positivos,

e considerar apenas máquinas padrão de agora em diante.¹

Queríamos mostrar que o conjunto de máquinas de Turing é **enumerável** e, tendo em mente as considerações acima, basta mostrar que o conjunto de máquinas de Turing padrão é **enumerável**. Suponha que nos seja dada uma máquina de Turing padrão $M = \langle Q, \Sigma, q_0, \delta \rangle$. Como poderíamos descrevê-la usando uma cadeia finita de inteiros positivos? Primeiro listaremos o número de estados, os próprios estados, o número de símbolos, os próprios símbolos e o estado inicial. (Lembre-se, todos esses são inteiros positivos, já que M é uma máquina padrão.) E quanto a δ ? O conjunto de argumentos possíveis, isto é, pares $\langle q, \sigma \rangle$, é finito, já que Q e Σ são finitos. Assim, a informação em δ é simplesmente a lista finita de todas as 5-uplas $\langle q, \sigma, q', \sigma', d \rangle$ em que $\delta(q, \sigma) = \langle q', \sigma', D \rangle$, e d é um número que codifica a direção D (digamos, 1 para L , 2 para R e 3 para N).

Dessa forma, toda máquina de Turing padrão pode ser descrita por uma lista finita de inteiros positivos, isto é, como uma sequência $s_M \in (\mathbb{Z}^+)^*$. Por exemplo, a máquina *Par* padrão é codificada pela sequência

$$\underbrace{2, 1, 2, 3}_Q, \underbrace{1, 2, 3, 1}_{\Sigma}, \underbrace{1, 3, 2, 3, 2}_{\delta(1,3)=\langle 2,3,R \rangle}, \underbrace{2, 2, 2, 2, 2}_{\delta(2,2)=\langle 2,2,R \rangle}, \underbrace{2, 3, 1, 3, 2}_{\delta(2,3)=\langle 1,3,R \rangle}.$$

Teorema 32.1. *Há funções de $\mathbb{N}$ a $\mathbb{N}$ que não são Turing-computáveis.*

Prova. Sabemos que o conjunto de sequências finitas de inteiros positivos $(\mathbb{Z}^+)^*$ é **enumerável** (**Problema 4.7**). Isso nos dá que o conjunto de descrições de máquinas de Turing padrão, como um subconjunto de $(\mathbb{Z}^+)^*$, é ele próprio enumerável. Toda função Turing-computável de $\mathbb{N}$ a $\mathbb{N}$ é computada por alguma (na verdade, muitas) máquina de Turing. Renomeando seus estados e símbolos para inteiros positivos (em particular, $\triangleright$ como 1, 0 como 2 e 1 como 3), podemos ver que toda função Turing-computável é computada por uma máquina de Turing padrão. Isso significa que o conjunto de todas as funções Turing-computáveis de $\mathbb{N}$ a $\mathbb{N}$ é também enumerável.

Por outro lado, o conjunto de todas as funções de $\mathbb{N}$ a $\mathbb{N}$ não é **enumerável** (**Problema 4.35**). Se todas as funções fossem computáveis por alguma máquina de Turing, poderíamos enumerar o conjunto de todas as funções listando todas as descrições de máquinas de Turing que as computam. Logo, há algumas funções que não são Turing-computáveis. $\square$

Problema 32.1. Você consegue pensar em uma maneira de descrever máquinas de Turing que não exija que os estados e os símbolos do alfabeto sejam listados explicitamente? Você pode definir sua própria noção de máquina “padrão”, mas diga algo sobre por que toda máquina de Turing pode ser computada por uma máquina “padrão” no seu novo sentido.

content/turing-machines/undecidability/universal-tm.tex

¹A terminologia “máquina padrão” não é padrão.

32.3. MÁQUINAS DE TURING UNIVERSAIS

32.3 Máquinas de Turing Universais

tur:und:uni:
sec

Na **Seção 32.2** discutimos como toda máquina de Turing pode ser descrita por uma sequência finita de inteiros. Essa sequência codifica os estados, o alfabeto, o estado inicial e as instruções da máquina de Turing. Também observamos que o conjunto de todas essas descrições é **enumerável**. Como o conjunto de tais descrições é **infinito enumerável**, isso significa que há uma função **sobrejetiva** de $\mathbb{N}$ nessas descrições. Uma tal função **sobrejetiva** pode ser obtida, por exemplo, usando o método zigue-zague de Cantor. Ele nos dá uma maneira de enumerar todas as (descrições de) máquinas de Turing. Se fixarmos uma tal enumeração, passa a fazer sentido falar da primeira, da segunda, $\dots$, e -ésima máquina de Turing. Esses números são chamados de *índices*.

Definição 32.2. Se M é a e -ésima máquina de Turing (em nossa enumeração fixada), dizemos que e é um *índice* de M . Escrevemos M_e para a e -ésima máquina de Turing.

Uma máquina pode ter mais de um índice; por exemplo, duas descrições de M podem diferir na ordem em que listamos suas instruções, e essas diferentes descrições terão índices diferentes.

É importante notar que é possível dar a enumeração das descrições de máquinas de Turing de tal modo que possamos computar efetivamente a descrição de M a partir de seu índice, e computar efetivamente um índice de uma máquina M a partir de sua descrição. Pela tese de Church-Turing, é então possível encontrar uma máquina de Turing que recupera a descrição da máquina de Turing com índice e e escreve a descrição correspondente em sua fita como saída. A descrição seria uma sequência de blocos de 1 (representando os inteiros positivos na sequência que descreve M_e).

Diante disso, torna-se natural perguntar: que funções de índices de máquinas de Turing são elas próprias computáveis por máquinas de Turing? Que propriedades de índices de máquinas de Turing podem ser decididas por máquinas de Turing? Um exemplo: a função que leva um índice e ao número de estados que a máquina de Turing com índice e possui é computável por uma máquina de Turing. Eis o que tal máquina de Turing faria: iniciada em uma fita contendo um único bloco de e 1, ela primeiro decodificaria e em sua descrição. A descrição passa a ser representada por uma sequência de blocos de 1 na fita. Como o primeiro **membro** nessa sequência é o número de estados, tudo o que resta a fazer agora é apagar tudo, exceto o primeiro bloco de 1, e então parar.

Um resultado notável é o seguinte:

tur:und:uni:
thm:universal-tm

Teorema 32.3. *Há uma máquina de Turing universal U que, quando iniciada na entrada $\langle e, n \rangle$,*

1. *para se, e somente se, M_e para na entrada n , e*
2. *se M_e para com saída m , então U também para.*

Assim, U computa a função $f: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ dada por $f(e, n) = m$ se M_e iniciada na entrada n para com saída m , e indefinida caso contrário.

Prova. Produzir de fato U é basicamente impossível, já que se trata de uma máquina extremamente complicada. Mas podemos descrever em linhas gerais como ela funciona, e então invocar a tese de Church-Turing. Quando ela começa, a fita de U contém um bloco de e 1 seguido de um bloco de n 1. Ela primeiro “decodifica” o índice e à direita da entrada n . Isso produz uma lista de números (isto é, blocos de 1 separados por 0) que descreve as instruções da máquina M_e . Em seguida, U escreve o número do estado inicial de M_e e o número 1 na fita, à direita da descrição de M_e . (Novamente, esses são representados em unário, como blocos de 1.) Depois, ela copia a entrada (bloco de n 1) para a direita—mas substitui cada 1 por um bloco de três 1 (lembre-se, o número do símbolo 1 é 3, sendo 1 o número de $\triangleright$ e 2 o número de 0). Na extremidade esquerda dessa sequência de blocos (separados por símbolos 0 na fita de U), ela escreve um único 1, o código de $\triangleright$.

U agora tem em sua fita: o índice e , o número n , o número de código do estado inicial (o “estado atual”), o número da posição inicial do cabeçote 1 (a “posição atual do cabeçote”) e o conteúdo inicial da “fita” (uma sequência de blocos de 1 representando os números de código dos símbolos de M_e —os “símbolos”—separados por 0).

Ela agora simula o que M_e faria se iniciada na entrada n , fazendo o seguinte:

1. Encontrar o número k da “posição atual do cabeçote” (no início, esse é 1),
2. Mover-se para o k -ésimo bloco na “fita” para ver qual é o “símbolo” ali,
3. Encontrar a instrução que corresponde ao “estado” e ao “símbolo” atuais, tur:und:uni:
find-inst
4. Voltar para o k -ésimo bloco na “fita” e substituir o “símbolo” ali pelo número de código do símbolo que M_e escreveria,
5. Mover o cabeçote para onde ela registra o “estado” atual e substituir o número ali pelo número do novo estado,
6. Mover-se para o lugar onde ela registra a “posição na fita” e apagar um 1 ou adicionar um 1 (se a instrução mandar mover para a esquerda ou para a direita, respectivamente).
7. Repetir.²

Se M_e iniciada na entrada n nunca para, então U também nunca para, de modo que sua saída é indefinida.

²Estamos passando por cima de algumas dificuldades sutis aqui. Por exemplo, U pode precisar de algum espaço extra quando incrementa o contador onde mantém o registro da “posição atual do cabeçote”—nesse caso, terá de mover toda a “fita” para a direita.

32.4. O PROBLEMA DA PARADA

Se no passo (3) acontecer de a descrição de M_e não conter nenhuma instrução para o par “estado”/“símbolo” atual, então M_e pararia. Se isso acontecer, U apaga a parte de sua fita à esquerda da “fita”. Para cada bloco de três 1 (representando um 1 na fita de M_e), ela escreve um 1 na extremidade esquerda de sua própria fita, e apaga sucessivamente a “fita”. Quando isso está concluído, a fita de U contém um único bloco de 1 de comprimento m .

Se U encontrar algo diferente de um bloco de três 1 na “fita”, ela para imediatamente. Como a fita de U nesse caso não contém um único bloco de 1, sua saída não é um número natural, isto é, $f(e, n)$ é indefinida nesse caso. $\square$

content/turing-machines/undecidability/halting-problem.tex

32.4 O Problema da Parada

tur:und:hal:
sec

Suponha que tenhamos fixado alguma enumeração das descrições de máquinas de Turing. Cada máquina de Turing recebe assim um *índice*: seu lugar na enumeração $M_1, M_2, M_3, \dots$ das descrições de máquinas de Turing.

explanation

Sabemos que devem existir funções não-Turing-computáveis: o conjunto de descrições de máquinas de Turing—e, portanto, o conjunto de máquinas de Turing—é **enumerável**, mas o conjunto de todas as funções de $\mathbb{N}$ a $\mathbb{N}$ não é. Mas também podemos encontrar exemplos específicos de funções não computáveis. Uma dessas funções é a função de parada.

Definição 32.4 (Função de parada). A *função de parada* h é definida como

$$h(e, n) = \begin{cases} 0 & \text{se a máquina } M_e \text{ não para na entrada } n \\ 1 & \text{se a máquina } M_e \text{ para na entrada } n \end{cases}$$

Definição 32.5 (Problema da parada). O *Problema da Parada* é o problema de determinar (para quaisquer e, n) se a máquina de Turing M_e para com uma entrada de n traços.

Mostramos que h não é Turing-computável mostrando que uma função relacionada, s , não é Turing-computável. Esta prova baseia-se no fato de que tudo o que pode ser computado por uma máquina de Turing pode ser computado por uma máquina de Turing disciplinada (**Seção 31.7**), e no fato de que duas máquinas de Turing podem ser conectadas para criar uma única máquina (**Seção 31.8**).

explanation

Definição 32.6. A função s é definida como

$$s(e) = \begin{cases} 0 & \text{se a máquina } M_e \text{ não para na entrada } e \\ 1 & \text{se a máquina } M_e \text{ para na entrada } e \end{cases}$$

Lema 32.7. A função s não é Turing-computável.

Prova. Suponha, por contradição, que a função s seja Turing-computável. Então haveria uma máquina de Turing S que computa s . Podemos supor, sem perda de generalidade, que quando S para, ela o faz examinando o primeiro quadrado (isto é, que ela é disciplinada). Esta máquina pode ser “conectada” a outra máquina J , que para se for iniciada na entrada 0 (isto é, se ela lê 0 no estado inicial enquanto examina o quadrado à direita do símbolo de fim de fita), e caso contrário vagueia para a direita, sem nunca parar. $S \frown J$, a máquina criada conectando-se S a J , é uma máquina de Turing, logo é M_e para algum e (isto é, aparece em algum lugar da enumeração). Inicie M_e em uma entrada de e 1. Há duas possibilidades: ou M_e para ou não para.

1. Suponha que M_e pare para uma entrada de e 1. Então $s(e) = 1$. Assim, S , quando iniciada em e , para com um único 1 como saída na fita. Então J começa com um 1 na fita. Nesse caso, J não para. Mas M_e é a máquina $S \frown J$, de modo que ela deveria fazer exatamente o que S seguida de J faria (isto é, neste caso, vaguear para a direita e nunca parar). Logo, M_e não pode parar para uma entrada de e 1.
2. Agora suponha que M_e não pare para uma entrada de e 1. Então $s(e) = 0$, e S , quando iniciada na entrada e , para com uma fita em branco. J , quando iniciada em uma fita em branco, para imediatamente. Novamente, M_e faz o que S seguida de J faria, de modo que M_e deve parar para uma entrada de e 1.

Em cada caso chegamos a uma contradição com nossa suposição. Isso mostra que não pode haver uma máquina de Turing S : s não é Turing-computável. $\square$

Teorema 32.8 (Insolubilidade do Problema da Parada). *O problema da parada é insolúvel, isto é, a função h não é Turing-computável.* tur:und:hal:
thm:halting-problem

Prova. Suponha que h fosse Turing-computável, digamos, por uma máquina de Turing H . Poderíamos usar H para construir uma máquina de Turing que computa s : primeiro, fazer uma cópia da entrada (separada por um símbolo 0). Depois, voltar ao início e executar H . Claramente podemos fazer uma máquina que realiza a primeira tarefa (veja [Problema 31.13](#)), e se H existisse, poderíamos “conectá-la” a tal máquina copiadora para obter uma nova máquina que determinaria se M_e para na entrada e , isto é, computa s . Mas já mostramos que não pode existir tal máquina. Portanto, h também não é Turing-computável. $\square$

Problema 32.2. O problema da Parada com Três (3-Parada) é o problema de dar um procedimento de decisão para determinar se uma máquina de Turing escolhida arbitrariamente para ou não para uma entrada de três 1 em uma fita que, fora isso, está em branco. Prove que o problema 3-Parada é insolúvel.

Problema 32.3. Mostre que se o problema da parada é solúvel para pares de máquina de Turing e entrada M_e e n em que $e \neq n$, então ele também é solúvel para os casos em que $e = n$.

32.5. O PROBLEMA DA DECISÃO

Problema 32.4. Provamos que o problema da parada é insolúvel se a entrada é um número e , que identifica uma máquina de Turing M_e por meio de uma enumeração de todas as máquinas de Turing. E se permitirmos a descrição de máquinas de Turing de [Seção 32.2](#) diretamente como entrada? Pode haver uma máquina de Turing que decide o problema da parada, mas que recebe como entrada descrições de máquinas de Turing em vez de índices? Explique por que sim ou por que não.

Problema 32.5. Mostre que a função *parcial* s' definida como

$$s'(e) = \begin{cases} 1 & \text{se a máquina } M_e \text{ para na entrada } e \\ \text{indefinida} & \text{se a máquina } M_e \text{ não para na entrada } e \end{cases}$$

é Turing-computável.

[content/turing-machines/undecidability/decision-problem.tex](#)

32.5 O Problema da Decisão

tur:und:dec:
sec

Dizemos que a lógica de primeira ordem é *decidível* se, e somente se, há um método efetivo para determinar se uma dada [sentença](#) é válida ou não. Acontece que não existe tal método: o problema de decidir a validade de sentenças de primeira ordem é insolúvel.

A fim de estabelecer esse importante resultado negativo, provamos que o problema da decisão não pode ser resolvido por uma máquina de Turing. Isto é, mostramos que não existe máquina de Turing que, sempre que iniciada em uma fita que contém uma [sentença](#) de primeira ordem, eventualmente para e produz 1 ou 0 conforme a [sentença](#) seja válida ou não. Pela tese de Church-Turing, toda função que é computável é Turing-computável. Logo, se essa “função de validade” fosse de algum modo efetivamente computável, ela seria Turing-computável. Se ela não é Turing-computável, então também não pode ser efetivamente computável.

Nossa estratégia para provar que o problema da decisão é insolúvel é reduzir o problema da parada a ele. Isso significa o seguinte: provamos que a função $h(e, w)$ que para com saída 1 se a máquina de Turing descrita por e para na entrada w e produz 0 caso contrário não é Turing-computável. Mostraremos que se houvesse uma máquina de Turing que decide a validade de sentenças de primeira ordem, então haveria também uma máquina de Turing que computa h . Como h não pode ser computada por uma máquina de Turing, também não pode haver uma máquina de Turing que decide a validade.

O primeiro passo nessa estratégia é mostrar que, para toda entrada w e máquina de Turing M , podemos descrever efetivamente [a sentença](#) $\tau(M, w)$ representando o conjunto de instruções de M e a entrada w , e [a sentença](#) $\alpha(M, w)$ exprimindo “ M eventualmente para”, tal que:

$\models \tau(M, w) \rightarrow \alpha(M, w)$ se, e somente se, M para na entrada w .

A maior parte de nossa prova consistirá em descrever essas sentenças $\tau(M, w)$ e $\alpha(M, w)$ e em verificar que $\tau(M, w) \rightarrow \alpha(M, w)$ é válida se, e somente se, M para na entrada w .

`content/turing-machines/undecidability/representing-tms.tex`

32.6 Representando Máquinas de Turing

explanation A fim de representar máquinas de Turing e seu comportamento por meio de tur:und:rep: sec a **sentença** da lógica de primeira ordem, temos de definir uma linguagem adequada. A linguagem consiste em duas partes: **símbolos de predicado** para descrever configurações da máquina, e expressões para numerar os passos de execução (“momentos”) e as posições na fita.

Introduzimos dois tipos de **símbolos de predicado**, ambos de 2 lugares: para cada estado q , o **símbolo de predicado** Q_q , e para cada símbolo de fita σ , o **símbolo de predicado** S_σ . Os primeiros permitem-nos descrever o estado de M e a posição de seu cabeçote de fita; os segundos permitem-nos descrever o conteúdo da fita.

A fim de exprimir as posições do cabeçote de fita e o número de passos executados, precisamos de uma maneira de exprimir números. Isso é feito usando o **símbolo de constante** o e uma função de 1 lugar ι , a função sucessor. Por convenção, ela é escrita *depois* de seu argumento (e omitimos os parênteses).

Para cada número n há um termo canônico $\bar{n}$, o *numeral* de n , que o representa em $\mathcal{L}_M$. $\bar{0}$ é o , $\bar{1}$ é o' , $\bar{2}$ é o'' , e assim por diante. Mais formalmente:

$$\begin{aligned}\bar{0} &= o \\ \overline{n+1} &= \bar{n}'\end{aligned}$$

O termo $\bar{0}$, isto é, o , nomeia a posição mais à esquerda na fita, bem como o instante anterior ao primeiro passo de execução (a configuração inicial). O termo $\bar{1}$, isto é, o' , nomeia o quadrado à direita do quadrado mais à esquerda, e o instante posterior ao primeiro passo de execução, e assim por diante.

Também introduzimos o **símbolo de predicado** $<$ para exprimir tanto a ordenação das posições na fita (quando significa “à esquerda de”) quanto a dos passos de execução (caso em que significa “antes”).

Uma vez que tenhamos a linguagem montada, listamos os “axiomas” de $\tau(M, w)$, isto é, as **sentenças** que, tomadas em conjunto, descrevem o comportamento de M quando executada na entrada w . Haverá **sentenças** que estabelecem condições sobre o , ι e $<$, **sentenças** que descrevem a configuração de entrada, e **sentenças** que descrevem qual é a configuração de M após executar uma instrução particular.

Definição 32.9. Dada uma máquina de Turing $M = \langle Q, \Sigma, q_0, \delta \rangle$, a linguagem $\mathcal{L}_M$ consiste em:

tur:und:rep: defn:tm-descr

32.6. REPRESENTANDO MÁQUINAS DE TURING

1. Um **símbolo de predicado** de dois lugares $Q_q(x, y)$ para todo estado $q \in Q$. Intuitivamente, $Q_q(\overline{m}, \overline{n})$ exprime “após n passos, M está no estado q examinando o m -ésimo quadrado.”
2. Um **símbolo de predicado** de dois lugares $S_\sigma(x, y)$ para todo símbolo $\sigma \in \Sigma$. Intuitivamente, $S_\sigma(\overline{m}, \overline{n})$ exprime “após n passos, o m -ésimo quadrado contém o símbolo σ .”
3. Um **símbolo de constante** o
4. Um **símbolo de função** de um lugar $'$
5. Um **símbolo de predicado** de dois lugares $<$

As **sentenças** que descrevem a operação da máquina de Turing M na entrada $w = \sigma_{i_1} \dots \sigma_{i_k}$ são as seguintes:

1. Axiomas que descrevem números e $<$:
 - a) A **sentença** que diz que todo número é menor que seu sucessor:

$$\forall x x < x'$$

- b) A **sentença** que garante que $<$ é transitivo:

$$\forall x \forall y \forall z ((x < y \wedge y < z) \rightarrow x < z)$$

2. Axiomas que descrevem a configuração de entrada:
 - a) Após 0 passos—antes de a máquina começar— M está no estado inicial q_0 , examinando o quadrado 1:

$$Q_{q_0}(\overline{1}, \overline{0})$$

- b) Os primeiros $k + 1$ quadrados contêm os símbolos $\triangleright, \sigma_{i_1}, \dots, \sigma_{i_k}$:

$$S_{\triangleright}(\overline{0}, \overline{0}) \wedge S_{\sigma_{i_1}}(\overline{1}, \overline{0}) \wedge \dots \wedge S_{\sigma_{i_k}}(\overline{k}, \overline{0})$$

- c) Caso contrário, a fita está vazia:

$$\forall x (\overline{k} < x \rightarrow S_0(x, \overline{0}))$$

3. Axiomas que descrevem a transição de uma configuração para a seguinte:
Para o que segue, seja $\varphi(x, y)$ a conjunção de todas as **sentenças** da forma

$$\forall z (((z < x \vee x < z) \wedge S_\sigma(z, y)) \rightarrow S_\sigma(z, y'))$$

em que $\sigma \in \Sigma$. Usamos $\varphi(\overline{m}, \overline{n})$ para exprimir “exceto no quadrado m , a fita após $n + 1$ passos é a mesma que após n passos.”

- a) Para toda instrução $\delta(q_i, \sigma) = \langle q_j, \sigma', R \rangle$, a **sentença**:

tur:und:rep:
rep-right

$$\forall x \forall y ((Q_{q_i}(x, y) \wedge S_\sigma(x, y)) \rightarrow (Q_{q_j}(x', y') \wedge S_{\sigma'}(x, y') \wedge \varphi(x, y)))$$

Isto diz que se, após y passos, a máquina está no estado q_i examinando o quadrado x que contém o símbolo σ , então após $y + 1$ passos ela está examinando o quadrado $x + 1$, está no estado q_j , o quadrado x agora contém σ' , e todo quadrado distinto de x contém o mesmo símbolo que continha após y passos.

- b) Para toda instrução $\delta(q_i, \sigma) = \langle q_j, \sigma', L \rangle$, a **sentença**:

tur:und:rep:
rep-left

$$\begin{aligned} \forall x \forall y ((Q_{q_i}(x', y) \wedge S_\sigma(x', y)) \rightarrow (Q_{q_j}(x, y') \wedge S_{\sigma'}(x', y') \wedge \varphi(x, y))) \wedge \\ \forall y ((Q_{q_i}(\bar{0}, y) \wedge S_\sigma(\bar{0}, y)) \rightarrow (Q_{q_j}(\bar{0}, y') \wedge S_{\sigma'}(\bar{0}, y') \wedge \varphi(\bar{0}, y))) \end{aligned}$$

Reserve um momento para pensar em como isso funciona: agora não começamos com “se examinando o quadrado $x \dots$ ”, mas: “se examinando o quadrado $x + 1 \dots$ ”. Um movimento para a esquerda significa que no próximo passo a máquina está examinando o quadrado x . Mas o quadrado em que se escreve é $x + 1$. Fazemos assim porque não temos subtração nem uma função predecessor.

Note que os números da forma $x + 1$ são $1, 2, \dots$, isto é, isso não cobre o caso em que a máquina está examinando o quadrado 0 e deveria mover-se para a esquerda (o que, é claro, não pode fazer—ela simplesmente permanece onde está). Esse caso especial é coberto pela segunda conjunção: ela diz que se, após y passos, a máquina está examinando o quadrado 0 no estado q_i e o quadrado 0 contém o símbolo σ , então após $y + 1$ passos ela ainda está examinando o quadrado 0 , está agora no estado q_j , o símbolo no quadrado 0 é σ' , e os quadrados distintos do quadrado 0 contém os mesmos símbolos que continham após y passos.

- c) Para toda instrução $\delta(q_i, \sigma) = \langle q_j, \sigma', N \rangle$, a **sentença**:

tur:und:rep:
rep-stay

$$\forall x \forall y ((Q_{q_i}(x, y) \wedge S_\sigma(x, y)) \rightarrow (Q_{q_j}(x, y') \wedge S_{\sigma'}(x, y') \wedge \varphi(x, y)))$$

Seja $\tau(M, w)$ a conjunção de todas as **sentenças** acima para a máquina de Turing M e a entrada w .

A fim de exprimir que M eventualmente para, temos de encontrar uma **sentença** que diga “após algum número de passos, a função de transição será indefinida.” Seja X o conjunto de todos os pares $\langle q, \sigma \rangle$ tais que $\delta(q, \sigma)$ é indefinida. Seja então $\alpha(M, w)$ a **sentença**

$$\exists x \exists y (\bigvee_{\langle q, \sigma \rangle \in X} (Q_q(x, y) \wedge S_\sigma(x, y)))$$

32.7. VERIFICANDO A REPRESENTAÇÃO

Se usarmos uma máquina de Turing com um estado de parada designado h , é ainda mais fácil: então a **sentença** $\alpha(M, w)$

$$\exists x \exists y Q_h(x, y)$$

exprime que a máquina eventualmente para.

Proposição 32.10. *Se $m < k$, então $\tau(M, w) \models \overline{m} < \overline{k}$*

*tur:und:rep:
prop:mlessk*

Prova. Exercício. □

Problema 32.6. Prove **Proposição 32.10**. (Dica: use indução em $k - m$).

content/turing-machines/undecidability/verification.tex

32.7 Verificando a Representação

*tur:und:ver:
sec*

A fim de verificar que nossa representação funciona, temos de provar duas coisas. Primeiro, temos de mostrar que se M para na entrada w , então $\tau(M, w) \rightarrow \alpha(M, w)$ é válida. Em seguida, temos de mostrar a recíproca, isto é, que se $\tau(M, w) \rightarrow \alpha(M, w)$ é válida, então M de fato eventualmente para quando executada na entrada w .

explanation

A estratégia para prová-las é bem diferente. Para o primeiro resultado, temos de mostrar que a **sentença** da lógica de primeira ordem (a saber, $\tau(M, w) \rightarrow \alpha(M, w)$) é válida. A maneira mais fácil de fazer isso é dar a **derivação**. Nossa prova, porém, deve funcionar para todos os M e w , de modo que não há realmente uma única **sentença** para a qual tenhamos de dar a **derivação**, mas infinitas. Assim, o melhor que podemos fazer é provar por indução que, sejam quais forem M e w , e por mais passos que M leve para parar na entrada w , haverá a **derivação** de $\tau(M, w) \rightarrow \alpha(M, w)$.

Naturalmente, nossa indução procederá sobre o número de passos que M leva antes de atingir uma configuração de parada. Em nossa prova indutiva, estabeleceremos que para cada passo n da execução de M na entrada w , $\tau(M, w) \models \chi(M, w, n)$, em que $\chi(M, w, n)$ descreve corretamente a configuração de M executada em w após n passos. Ora, se M para na entrada w após, digamos, n passos, $\chi(M, w, n)$ descreverá uma configuração de parada. Mostraremos também que $\chi(M, w, n) \models \alpha(M, w)$ sempre que $\chi(M, w, n)$ descreve uma configuração de parada. Assim, se M para na entrada w , então para algum n , M estará em uma configuração de parada após n passos. Logo, $\tau(M, w) \models \chi(M, w, n)$ em que $\chi(M, w, n)$ descreve uma configuração de parada, e como nesse caso $\chi(M, w, n) \models \alpha(M, w)$, obtemos que $T(M, w) \models \alpha(M, w)$, isto é, que $\models \tau(M, w) \rightarrow \alpha(M, w)$.

A estratégia para a recíproca é bem diferente. Aqui supomos que $\models \tau(M, w) \rightarrow \alpha(M, w)$ e temos de provar que M para na entrada w . Da hipótese obtemos que $\tau(M, w) \models \alpha(M, w)$, isto é, $\alpha(M, w)$ é verdadeira em toda **estrutura** em que $\tau(M, w)$ é verdadeira. Assim, descreveremos a **estrutura** $\mathfrak{M}$ em que $\tau(M, w)$

é verdadeira: seu domínio será $\mathbb{N}$, e a interpretação de todos os Q_q e S_σ será dada pelas configurações de M durante uma execução na entrada w . Assim, por exemplo, $\mathfrak{M} \models Q_q(\bar{m}, \bar{n})$ se, e somente se, T , quando executada na entrada w por n passos, está no estado q e examinando o quadrado m . Ora, como $\tau(M, w) \models \alpha(M, w)$ por hipótese, e como $\mathfrak{M} \models \tau(M, w)$ por construção, $\mathfrak{M} \models \alpha(M, w)$. Mas $\mathfrak{M} \models \alpha(M, w)$ se, e somente se, há algum $n \in |\mathfrak{M}| = \mathbb{N}$ tal que M , executada na entrada w , está em uma configuração de parada após n passos.

Definição 32.11. Seja $\chi(M, w, n)$ a **sentença**

$$Q_q(\bar{m}, \bar{n}) \wedge S_{\sigma_0}(\bar{0}, \bar{n}) \wedge \cdots \wedge S_{\sigma_k}(\bar{k}, \bar{n}) \wedge \forall x (\bar{k} < x \rightarrow S_0(x, \bar{n}))$$

em que q é o estado de M no instante n , M está examinando o quadrado m no instante n , o quadrado i contém o símbolo σ_i no instante n para $0 \leq i \leq k$, e k é o quadrado não-vazio mais à direita da fita no instante 0, ou o quadrado mais à direita que o cabeçote de fita visitou após n passos, o que for maior.

Lema 32.12. Se M executada na entrada w está em uma configuração de parada após n passos, então $\chi(M, w, n) \models \alpha(M, w)$.

*tur:und:ver:
lem:halt-config-implies-halt*

Prova. Suponha que M pare para a entrada w após n passos. Há algum estado q , quadrado m e símbolo σ tais que:

1. Após n passos, M está no estado q examinando o quadrado m no qual aparece σ .
2. A função de transição $\delta(q, \sigma)$ é indefinida.

$\chi(M, w, n)$ é a descrição dessa configuração e incluirá as cláusulas $Q_q(\bar{m}, \bar{n})$ e $S_\sigma(\bar{m}, \bar{n})$. Essas cláusulas juntas implicam $\alpha(M, w)$:

$$\exists x \exists y \left(\bigvee_{\langle q, \sigma \rangle \in X} (Q_q(x, y) \wedge S_\sigma(x, y)) \right)$$

já que $Q_{q'}(\bar{m}, \bar{n}) \wedge S_{\sigma'}(\bar{m}, \bar{n}) \models \bigvee_{\langle q, \sigma \rangle \in X} (Q_q(\bar{m}, \bar{n}) \wedge S_\sigma(\bar{m}, \bar{n}))$, pois $\langle q', \sigma' \rangle \in X$. $\square$

explanation

Assim, se M para na entrada w , então há algum n tal que $\chi(M, w, n) \models \alpha(M, w)$. Mostraremos agora que para qualquer instante n , $\tau(M, w) \models \chi(M, w, n)$.

Lema 32.13. Para cada n , se M não parou após n passos, $\tau(M, w) \models \chi(M, w, n)$.

*tur:und:ver:
lem:config*

Prova. Base da indução: Se $n = 0$, então as fórmulas componentes de $\chi(M, w, 0)$ são também fórmulas componentes de $\tau(M, w)$, logo implicadas por ela.

Hipótese de indução: Se M não parou antes do n -ésimo passo, então $\tau(M, w) \models \chi(M, w, n)$. Temos de mostrar que (a menos que $\chi(M, w, n)$ descreva uma configuração de parada), $\tau(M, w) \models \chi(M, w, n + 1)$.

32.7. VERIFICANDO A REPRESENTAÇÃO

Suponha $n > 0$ e que, após n passos, M iniciada em w esteja no estado q examinando o quadrado m . Como M não para após n passos, deve haver no programa de M uma instrução de uma das três formas seguintes:

tur:und:ver: right tur:und:ver: left tur:und:ver: stay	1. $\delta(q, \sigma) = \langle q', \sigma', R \rangle$ 2. $\delta(q, \sigma) = \langle q', \sigma', L \rangle$ 3. $\delta(q, \sigma) = \langle q', \sigma', N \rangle$

Consideraremos cada um desses três casos por sua vez.

1. Suponha que haja uma instrução da forma (1). Por **Definição 32.9(3a)**, isso significa que

$$\forall x \forall y ((Q_q(x, y) \wedge S_\sigma(x, y)) \rightarrow (Q_{q'}(x', y') \wedge S_{\sigma'}(x, y') \wedge \varphi(x, y)))$$

é uma fórmula componente de $\tau(M, w)$. Isso implica a seguinte **sentença** (instanciação universal, $\bar{m}$ por x e $\bar{n}$ por y):

$$(Q_q(\bar{m}, \bar{n}) \wedge S_\sigma(\bar{m}, \bar{n})) \rightarrow (Q_{q'}(\bar{m}', \bar{n}') \wedge S_{\sigma'}(\bar{m}, \bar{n}') \wedge \varphi(\bar{m}, \bar{n})).$$

Por hipótese de indução, $\tau(M, w) \models \chi(M, w, n)$, isto é,

$$Q_q(\bar{m}, \bar{n}) \wedge S_{\sigma_0}(\bar{0}, \bar{n}) \wedge \cdots \wedge S_{\sigma_k}(\bar{k}, \bar{n}) \wedge \forall x (\bar{k} < x \rightarrow S_0(x, \bar{n}))$$

Como após n passos o quadrado de fita m contém σ , a fórmula componente correspondente é $S_\sigma(\bar{m}, \bar{n})$, de modo que isso implica:

$$Q_q(\bar{m}, \bar{n}) \wedge S_\sigma(\bar{m}, \bar{n})$$

Obtemos agora

$$Q_{q'}(\bar{m}', \bar{n}') \wedge S_{\sigma'}(\bar{m}, \bar{n}') \wedge S_{\sigma_0}(\bar{0}, \bar{n}') \wedge \cdots \wedge S_{\sigma_k}(\bar{k}, \bar{n}') \wedge \forall x (\bar{k} < x \rightarrow S_0(x, \bar{n}'))$$

como segue: A primeira linha vem diretamente do consequente do condicional precedente, por modus ponens. Cada fórmula componente na linha do meio—que exclui $S_{\sigma_m}(\bar{m}, \bar{n}')$ —decorre da fórmula componente correspondente em $\chi(M, w, n)$ junto com $\varphi(\bar{m}, \bar{n})$.

Se $m < k$, $\tau(M, w) \vdash \bar{m} < \bar{k}$ (**Proposição 32.10**) e, por transitividade de $<$, temos $\forall x (\bar{k} < x \rightarrow \bar{m} < x)$. Se $m = k$, então $\forall x (\bar{k} < x \rightarrow \bar{m} < x)$

apenas por lógica. A última linha decorre então da fórmula componente correspondente em $\chi(M, w, n)$, de $\forall x (\bar{k} < x \rightarrow \bar{m} < x)$ e de $\varphi(\bar{m}, \bar{n})$. Se $m < k$, isto já é $\chi(M, w, n + 1)$.

Agora suponha $m = k$. Nesse caso, após $n + 1$ passos, o cabeçote de fita também visitou o quadrado $k + 1$, que agora é o quadrado mais à direita visitado. Assim, $\chi(M, w, n + 1)$ tem uma nova fórmula componente, $S_0(\bar{k}', \bar{n}')$, e a última fórmula componente é $\forall x (\bar{k}' < x \rightarrow S_0(x, \bar{n}'))$. Temos de verificar que essas duas **sentenças** também são implicadas.

Já temos $\forall x (\bar{k} < x \rightarrow S_0(x, \bar{n}'))$. Em particular, isso nos dá $\bar{k} < \bar{k}' \rightarrow S_0(\bar{k}', \bar{n}')$. Do axioma $\forall x x < x'$ obtemos $\bar{k} < \bar{k}'$. Por modus ponens, segue-se $S_0(\bar{k}', \bar{n}')$.

Além disso, como $\tau(M, w) \vdash \bar{k} < \bar{k}'$, o axioma da transitividade de $<$ nos dá $\forall x (\bar{k}' < x \rightarrow S_0(x, \bar{n}'))$. (Deixamos a verificação disso como exercício.)

2. Suponha que haja uma instrução da forma (2). Então, por **Definição 32.9(3b)**,

$$\begin{aligned} & \forall x \forall y ((Q_q(x', y) \wedge S_\sigma(x', y)) \rightarrow \\ & \quad (Q_{q'}(x, y') \wedge S_{\sigma'}(x', y') \wedge \varphi(x, y))) \wedge \\ & \forall y ((Q_{q_i}(\bar{0}, y) \wedge S_\sigma(\bar{0}, y)) \rightarrow \\ & \quad (Q_{q_j}(\bar{0}, y') \wedge S_{\sigma'}(\bar{0}, y') \wedge \varphi(\bar{0}, y))) \end{aligned}$$

é uma fórmula componente de $\tau(M, w)$. Se $m > 0$, então seja $l = m - 1$ (isto é, $m = l + 1$). A primeira fórmula componente da **sentença** acima implica o seguinte:

$$\begin{aligned} & (Q_q(\bar{l}', \bar{n}) \wedge S_\sigma(\bar{l}', \bar{n})) \rightarrow \\ & \quad (Q_{q'}(\bar{l}, \bar{n}') \wedge S_{\sigma'}(\bar{l}', \bar{n}') \wedge \varphi(\bar{l}, \bar{n})) \end{aligned}$$

Caso contrário, seja $l = m = 0$ e considere a seguinte **sentença** implicada pela segunda fórmula componente:

$$\begin{aligned} & ((Q_{q_i}(\bar{0}, \bar{n}) \wedge S_\sigma(\bar{0}, \bar{n})) \rightarrow \\ & \quad (Q_{q_j}(\bar{0}, \bar{n}') \wedge S_{\sigma'}(\bar{0}, \bar{n}') \wedge \varphi(\bar{0}, \bar{n}))) \end{aligned}$$

Qualquer das sentenças implica

$$\begin{aligned} & Q_{q'}(\bar{l}, \bar{n}') \wedge S_{\sigma'}(\bar{m}, \bar{n}') \wedge \\ & \quad S_{\sigma_0}(\bar{0}, \bar{n}') \wedge \cdots \wedge S_{\sigma_k}(\bar{k}, \bar{n}') \wedge \\ & \quad \forall x (\bar{k} < x \rightarrow S_0(x, \bar{n}')) \end{aligned}$$

como antes. (Note que no primeiro caso, $\bar{l}' \equiv \bar{l} + 1 \equiv \bar{m}$ e no segundo caso $\bar{l} \equiv \bar{0}$.) Mas isto é justamente $\chi(M, w, n + 1)$.

32.7. VERIFICANDO A REPRESENTAÇÃO

3. O caso (3) é deixado como exercício.

Mostramos que para qualquer n , $\tau(M, w) \models \chi(M, w, n)$. $\square$

Problema 32.7. Complete o caso (3) da prova de Lema 32.13.

Problema 32.8. Dê a derivação de $S_{\sigma_i}(\bar{i}, \bar{n}')$ a partir de $S_{\sigma_i}(\bar{i}, \bar{n})$ e $\varphi(m, n)$ (supondo $i \neq m$, isto é, $i < m$ ou $m < i$).

Problema 32.9. Dê a derivação de $\forall x (\bar{k}' < x \rightarrow S_0(x, \bar{n}'))$ a partir de $\forall x (\bar{k} < x \rightarrow S_0(x, \bar{n}'))$, $\forall x x < x'$ e $\forall x \forall y \forall z ((x < y \wedge y < z) \rightarrow x < z)$.

*tur:und:ver:
lem:valid-if-halt*

Lema 32.14. Se M para na entrada w , então $\tau(M, w) \rightarrow \alpha(M, w)$ é válida.

Prova. Por Lema 32.13, sabemos que, para qualquer instante n , a descrição $\chi(M, w, n)$ da configuração de M no instante n é implicada por $\tau(M, w)$. Suponha que M pare após k passos. Nesse ponto, ela estará examinando o quadrado m , para algum $m \in \mathbb{N}$. Então $\chi(M, w, k)$ descreve uma configuração de parada de M , isto é, contém como fórmulas componentes tanto $Q_q(\bar{m}, \bar{k})$ quanto $S_\sigma(\bar{m}, \bar{k})$ com $\delta(q, \sigma)$ indefinida. Assim, por Lema 32.12, $\chi(M, w, k) \models \alpha(M, w)$. Mas como $\tau(M, w) \models \chi(M, w, k)$, temos $\tau(M, w) \models \alpha(M, w)$ e portanto $\tau(M, w) \rightarrow \alpha(M, w)$ é válida. $\square$

Para completar a verificação de nossa afirmação, temos também de estabelecer a direção reversa: se $\tau(M, w) \rightarrow \alpha(M, w)$ é válida, então M de fato para quando iniciada na entrada w . explanation

*tur:und:ver:
lem:halt-if-valid*

Lema 32.15. Se $\tau(M, w) \rightarrow \alpha(M, w)$, então M para na entrada w .

Prova. Considere a $\mathcal{L}_M$ -estrutura $\mathfrak{M}$ com domínio $\mathbb{N}$ que interpreta o como 0, $'$ como a função sucessor e $<$ como a relação menor-que, e os predicados Q_q e S_σ como segue:

$$\begin{aligned} Q_q^{\mathfrak{M}} &= \{ \langle m, n \rangle : \begin{array}{l} \text{iniciada em } w, \text{ após } n \text{ passos,} \\ M \text{ está no estado } q \text{ examinando o quadrado } m \end{array} \} \\ S_\sigma^{\mathfrak{M}} &= \{ \langle m, n \rangle : \begin{array}{l} \text{iniciada em } w, \text{ após } n \text{ passos,} \\ \text{o quadrado } m \text{ de } M \text{ contém o símbolo } \sigma \end{array} \} \end{aligned}$$

Em outras palavras, construímos a estrutura $\mathfrak{M}$ de modo que ela descreva o que M iniciada na entrada w de fato faz, passo a passo. Claramente, $\mathfrak{M} \models \tau(M, w)$. Se $\tau(M, w) \rightarrow \alpha(M, w)$, então também $\mathfrak{M} \models \alpha(M, w)$, isto é,

$$\mathfrak{M} \models \exists x \exists y \left(\bigvee_{\langle q, \sigma \rangle \in X} (Q_q(x, y) \wedge S_\sigma(x, y)) \right).$$

Como $|\mathfrak{M}| = \mathbb{N}$, deve haver $m, n \in \mathbb{N}$ tais que $\mathfrak{M} \models Q_q(\bar{m}, \bar{n}) \wedge S_\sigma(\bar{m}, \bar{n})$ para algum q e σ tais que $\delta(q, \sigma)$ é indefinida. Pela definição de $\mathfrak{M}$, isso significa que M iniciada na entrada w após n passos está no estado q e lendo o símbolo σ , e a função de transição é indefinida, isto é, M parou. $\square$

32.8 O Problema da Decisão é Insolúvel

Teorema 32.16. *O problema da decisão é insolúvel: Não há máquina de Turing D tal que, quando iniciada em uma fita que contém como entrada a sentença ψ da lógica de primeira ordem, D eventualmente para e produz 1 se, e somente se, ψ é válida, e 0 caso contrário.*

tur:und:uns:
sec
tur:und:uns:
thm:decision-prob

Prova. Suponha que o problema da decisão fosse solúvel, isto é, suponha que houvesse uma máquina de Turing D . Então poderíamos resolver o problema da parada como segue. Construimos uma máquina de Turing E que, dado como entrada o número e da máquina de Turing M_e e a entrada w , computa a sentença correspondente $\tau(M_e, w) \rightarrow \alpha(M_e, w)$ e para, examinando o quadrado mais à esquerda na fita. A máquina $E \cap D$ então, dada a entrada e e w , primeiro computaria $\tau(M_e, w) \rightarrow \alpha(M_e, w)$ e depois executaria a máquina do problema da decisão D nessa entrada. D para com saída 1 se, e somente se, $\tau(M_e, w) \rightarrow \alpha(M_e, w)$ é válida, e produz 0 caso contrário. Por Lema 32.15 e Lema 32.14, $\tau(M_e, w) \rightarrow \alpha(M_e, w)$ é válida se, e somente se, M_e para na entrada w . Assim, $E \cap D$, dada a entrada e e w , para com saída 1 se, e somente se, M_e para na entrada w , e para com saída 0 caso contrário. Em outras palavras, $E \cap D$ resolveria o problema da parada. Mas sabemos, por Teorema 32.8, que não pode existir tal máquina de Turing. $\square$

Corolário 32.17. *É indecidível se uma sentença arbitrária da lógica de primeira ordem é satisfatível.*

tur:und:uns:
cor:undecidable-sat

Prova. Suponha que a satisfatibilidade fosse decidível por uma máquina de Turing S . Então poderíamos resolver o problema da decisão como segue: Dada como entrada a sentença B , mova ψ um quadrado para a direita. Volte ao quadrado 1 e escreva o símbolo $\neg$.

Agora execute a máquina de Turing S . Ela eventualmente para com saída 1 (se $\neg\psi$ é satisfatível) ou 0 (se $\neg\psi$ é insatisfatível) na fita. Se há um 1 no quadrado 1, apague-o; se o quadrado 1 está vazio, escreva um 1 e então pare.

Essa máquina de Turing sempre para, e sua saída é 1 se, e somente se, $\neg\psi$ é insatisfatível, e 0 caso contrário. Como ψ é válida se, e somente se, $\neg\psi$ é insatisfatível, a máquina produz 1 se, e somente se, ψ é válida, e 0 caso contrário, isto é, ela resolveria o problema da decisão. $\square$

explanation

Portanto, não há máquina de Turing que sempre dê uma resposta “sim” ou “não” correta à pergunta “ ψ é uma sentença válida da lógica de primeira ordem?” No entanto, há uma máquina de Turing que sempre dá uma resposta “sim” correta—mas que simplesmente não para se a resposta é “não”. Isso decorre do teorema da correção e completude da lógica de primeira ordem, e do fato de que derivações podem ser efetivamente enumeradas.

Teorema 32.18. *A validade de sentenças de primeira ordem é semidecidível: Há uma máquina de Turing E que, quando iniciada em uma fita que contém*

tur:und:uns:
thm:valid-ce

32.9. TEOREMA DE TRAKHTENBROT

como entrada a *sentença* ψ da lógica de primeira ordem, E eventualmente para e produz 1 se, e somente se, ψ é válida, mas não para caso contrário.

Prova. Todas as *derivações* possíveis da lógica de primeira ordem podem ser geradas, uma após a outra, por um algoritmo efetivo. A máquina E faz isso e, quando encontra a *derivação* que mostra que $\vdash \psi$, ela para com saída 1. Pelo teorema da correção, se E para com saída 1, é porque $\models \psi$. Pelo teorema da completude, se $\models \psi$, há a *derivação* que mostra que $\vdash \psi$. Como E gera sistematicamente todas as *derivações* possíveis, ela eventualmente encontrará uma que mostra $\vdash \psi$, de modo que eventualmente parará com saída 1. $\square$

content/turing-machines/undecidability/trakhtenbrot.tex

32.9 Teorema de Trakhtenbrot

tur:und:tra:
sec

Na [Seção 32.6](#) definimos as *sentenças* $\tau(M, w)$ e $\alpha(M, w)$ para uma máquina de Turing M e uma cadeia de entrada w . Em seguida, mostramos em [Lema 32.14](#) e [Lema 32.15](#) que $\tau(M, w) \rightarrow \alpha(M, w)$ é válida se, e somente se, M , iniciada na entrada w , eventualmente para. Como o Problema da Parada é indecidível, isso implica que a validade e a satisfatibilidade de *sentenças* da lógica de primeira ordem são indecidíveis ([Teorema 32.16](#) e [Corolário 32.17](#)).

explanation

Mas a validade e a satisfatibilidade de sentenças são definidas para *estruturas* arbitrárias, finitas ou infinitas. Você poderia suspeitar que é mais fácil decidir se a *sentença* é satisfatível em uma *estrutura* finita (ou válida em todas as *estruturas* finitas). Podemos adaptar a prova da insolubilidade do problema da decisão de modo que ela mostre que não é esse o caso.

Primeiro, se você voltar à prova de [Lema 32.15](#), verá que o que fizemos ali foi produzir um modelo $\mathfrak{M}$ de $\tau(M, w)$ que descreve exatamente o que a máquina M faz quando iniciada na entrada w . O domínio daquele modelo era $\mathbb{N}$, isto é, infinito. Mas se M de fato para na entrada w , podemos construir um modelo finito $\mathfrak{M}'$ da mesma maneira. Suponha que M iniciada na entrada w pare após k passos. Tome como domínio $|\mathfrak{M}'|$ o conjunto $\{0, \dots, n\}$, em que n é o maior entre k e o comprimento de w , e seja

$$\iota^{\mathfrak{M}'}(x) = \begin{cases} x + 1 & \text{se } x < n \\ n & \text{caso contrário,} \end{cases}$$

e $\langle x, y \rangle \in <^{\mathfrak{M}'}$ se, e somente se, $x < y$ ou $x = y = n$. Quanto ao resto, $\mathfrak{M}'$ é definida tal como $\mathfrak{M}$. Pela definição de $\mathfrak{M}'$, tal como na prova de [Lema 32.15](#), $\mathfrak{M}' \models \tau(M, w)$. E como supusemos que M para na entrada w , $\mathfrak{M}' \models \alpha(M, w)$. Assim, $\mathfrak{M}'$ é um modelo finito de $\tau(M, w) \wedge \alpha(M, w)$ (note que substituímos $\rightarrow$ por $\wedge$).

Estamos a meio caminho de uma prova: mostramos que se M para na entrada w , então $\tau(M, w) \wedge \alpha(M, w)$ tem um modelo finito. Infelizmente, a recíproca disso não vale, isto é, há máquinas de Turing que não param em

alguma entrada w , mas $\tau(M, w) \wedge \alpha(M, w)$ ainda assim tem um modelo finito. Por exemplo, considere a máquina M com o único estado q_0 e a instrução $\delta(q_0, 0) = \langle q_0, 0, N \rangle$. Iniciada na entrada vazia $w = \Lambda$, essa máquina nunca para: está em um laço infinito, mas não altera a fita nem move o cabeçote. Todas as configurações são as mesmas (mesmo estado, mesma posição do cabeçote, mesmo conteúdo da fita). Podemos definir uma **estrutura** finita $\mathfrak{M}''$ que satisfaz $\tau(M, \Lambda) \wedge \alpha(M, \Lambda)$ (exercício). Podemos, no entanto, modificar $\tau(M, w)$ de modo adequado para que tais **estruturas** sejam descartadas.

Problema 32.10. Seja M uma máquina de Turing com o único estado q_0 e a única instrução $\delta(q_0, 0) = \langle q, 0, N \rangle$. Seja $|\mathfrak{M}''| = \{0, 1, 2\}$, $\iota^{\mathfrak{M}''}(0) = \iota^{\mathfrak{M}''}(1) = 1$ e $\iota^{\mathfrak{M}''}(2) = 2$, e $<^{\mathfrak{M}''} = \{\langle 0, 1 \rangle, \langle 1, 1 \rangle, \langle 2, 2 \rangle\}$. Defina $Q_{q_0}^{\mathfrak{M}''}$, $S_0^{\mathfrak{M}''}$ e $S_{\triangleright}^{\mathfrak{M}''}$ de modo que $\tau(M, \Lambda)$ e $\alpha(M, \Lambda)$ se tornem verdadeiras e explique por que o são. Dica: Observe que $\delta(q_0, \triangleright)$ é indefinida. Garanta que

$$\begin{aligned} Q_{q_0}(\bar{1}, \bar{n}) \wedge S_{\triangleright}(\bar{0}, \bar{n}) \wedge \forall x (\bar{0} < x \rightarrow S_0(x, \bar{n})) \quad \text{para todo } n \in \mathbb{N} \\ \exists y (Q_{q_0}(\bar{0}, y) \wedge S_{\triangleright}(\bar{0}, y)) \end{aligned}$$

são ambas verdadeiras em $\mathfrak{M}''$.

Considere as **sentenças** que descrevem a operação da máquina de Turing M na entrada $w = \sigma_{i_1} \dots \sigma_{i_k}$:

1. Axiomas que descrevem números e $<$ (tal como na definição de $\tau(M, w)$ na **Seção 32.6**).
2. Axiomas que descrevem a configuração de entrada: tal como na definição de $\tau(M, w)$.
3. Axiomas que descrevem a transição de uma configuração para a seguinte:
Para o que segue, seja $\varphi(x, y)$ como antes, e seja

$$\psi(y) \equiv \forall x (x < y \rightarrow x \neq y).$$

- a) Para toda instrução $\delta(q_i, \sigma) = \langle q_j, \sigma', R \rangle$, a **sentença**:

tur:und:tra:
rep-right

$$\begin{aligned} \forall x \forall y ((Q_{q_i}(x, y) \wedge S_{\sigma}(x, y)) \rightarrow \\ (Q_{q_j}(x', y') \wedge S_{\sigma'}(x, y') \wedge \varphi(x, y) \wedge \psi(y'))) \end{aligned}$$

- b) Para toda instrução $\delta(q_i, \sigma) = \langle q_j, \sigma', L \rangle$, a **sentença**

tur:und:tra:
rep-left

$$\begin{aligned} \forall x \forall y ((Q_{q_i}(x', y) \wedge S_{\sigma}(x', y)) \rightarrow \\ (Q_{q_j}(x, y') \wedge S_{\sigma'}(x', y') \wedge \varphi(x, y))) \wedge \\ \forall y ((Q_{q_i}(\bar{0}, y) \wedge S_{\sigma}(\bar{0}, y)) \rightarrow \\ (Q_{q_j}(\bar{0}, y') \wedge S_{\sigma'}(\bar{0}, y') \wedge \varphi(\bar{0}, y) \wedge \psi(y'))) \end{aligned}$$

32.9. TEOREMA DE TRAKHTENBROT

tur:und:tra:
rep-stay

c) Para toda instrução $\delta(q_i, \sigma) = \langle q_j, \sigma', N \rangle$, a **sentença**:

$$\forall x \forall y ((Q_{q_i}(x, y) \wedge S_\sigma(x, y)) \rightarrow (Q_{q_j}(x, y') \wedge S_{\sigma'}(x, y') \wedge \varphi(x, y) \wedge \psi(y')))$$

Como você pode ver, as **sentenças** que descrevem as transições de M são as mesmas que a **sentença** correspondente em $\tau(M, w)$, exceto que adicionamos $\psi(y')$ ao final. $\psi(y')$ garante que o número y' da “próxima” configuração é diferente de todos os números anteriores $0, 0', \dots$.

Seja $\tau'(M, w)$ a conjunção de todas as **sentenças** acima para a máquina de Turing M e a entrada w .

tur:und:tra:
lem:halts-sat

Lema 32.19. *Se M iniciada na entrada w para, então $\tau'(M, w) \wedge \alpha(M, w)$ tem um modelo finito.*

Prova. Seja $\mathfrak{M}'$ como na prova de **Lema 32.15**, exceto que

$$\begin{aligned} |\mathfrak{M}'| &= \{0, \dots, n\}, \\ r^{\mathfrak{M}'}(x) &= \begin{cases} x+1 & \text{se } x < n \\ n & \text{caso contrário,} \end{cases} \\ \langle x, y \rangle &\in <^{\mathfrak{M}'} \text{ sse } x < y \text{ ou } x = y = n, \end{aligned}$$

em que $n = \max(k, \text{len}(w))$ e k é o menor número tal que M iniciada na entrada w parou após k passos. Deixamos como exercício a verificação de que $\mathfrak{M}' \models \tau'(M, w) \wedge E(M, w)$. $\square$

Problema 32.11. Complete a prova de **Lema 32.19** provando que $\mathfrak{M}' \models \tau(M, w) \wedge E(M, w)$.

tur:und:tra:
lem:sat-halts

Lema 32.20. *Se $\tau'(M, w) \wedge \alpha(M, w)$ tem um modelo finito, então M iniciada na entrada w para.*

Prova. Mostramos a contrapositiva. Suponha que M iniciada em w não pare. Se $\tau'(M, w) \wedge \alpha(M, w)$ não tem modelo algum, terminamos. Então suponha que $\mathfrak{M}$ seja um modelo de $\tau(M, w) \wedge \alpha(M, w)$. Temos de mostrar que ele não pode ser finito.

Podemos provar, tal como em **Lema 32.13**, que se M , iniciada na entrada w , não parou após n passos, então $\tau'(M, w) \models \chi(M, w, n) \wedge \psi(\bar{n})$. Como M iniciada na entrada w não para, $\tau'(M, w) \models \chi(M, w, n) \wedge \psi(\bar{n})$ para todo $n \in \mathbb{N}$. Note que, por **Proposição 32.10**, $\tau'(M, w) \models \bar{k} < \bar{n}$ para todo $k < n$. Além disso, $\psi(\bar{n}) \models \bar{k} < \bar{n} \rightarrow \bar{k} \neq \bar{n}$. Assim, $\mathfrak{M} \models \bar{k} \neq \bar{n}$ para todo $k < n$, isto é, os infinitos termos $\bar{k}$ devem ter todos valores diferentes em $\mathfrak{M}$. Mas isso requer que $|\mathfrak{M}|$ seja infinito, de modo que $\mathfrak{M}$ não pode ser um modelo finito de $\tau'(M, w) \wedge \alpha(M, w)$. $\square$

Problema 32.12. Complete a prova de **Lema 32.20** provando que se M , iniciada na entrada w , não parou após n passos, então $\tau'(M, w) \models \psi(\bar{n})$.

Teorema 32.21 (Teorema de Trakhtenbrot). *É indecidível se uma **sentença** arbitrária da lógica de primeira ordem tem um modelo finito (isto é, é finitamente satisfatível).* tur:und:tra:
thm:trakhtenbrodt

Prova. Suponha que houvesse uma máquina de Turing F que decide o problema da satisfatibilidade finita. Então, dada qualquer máquina de Turing M e entrada w , poderíamos computar a sentença $\tau'(M, w) \wedge \alpha(M, w)$ e usar F para decidir se ela tem um modelo finito. Por **Lemas 32.19** e **32.20**, ela tem se, e somente se, M iniciada na entrada w para. Assim, poderíamos usar F para resolver o problema da parada, que sabemos ser insolúvel. $\square$

Corolário 32.22. *Não pode haver um sistema de **derivação** que seja correto e completo para a validade finita, isto é, um sistema de **derivação** que tenha $\vdash \psi$ se, e somente se, $\mathfrak{M} \models \psi$ para toda **estrutura** finita $\mathfrak{M}$.* tur:und:tra:
cor:fproof-incomp

Prova. Exercício. $\square$

Problema 32.13. Prove **Corolário 32.22**. Observe que ψ é satisfeita em toda **estrutura** finita se, e somente se, $\neg\psi$ não é finitamente satisfatível. Explique por que a satisfatibilidade finita é semidecidível no sentido de **Teorema 32.18**. Use isso para argumentar que se houvesse um sistema de **derivação** para a validade finita, então a satisfatibilidade finita seria decidível.

Parte VII

Incompleteness

Material in this part covers the incompleteness theorems. It depends on material in the parts on first-order logic (esp., the proof system), the material on recursive functions (in the computability part). It is based on Jeremy Avigad's notes with revisions by Richard Zach.

Capítulo 33

Introduction to Incompleteness

-NoValue-

`content/incompleteness/introduction/historical-background.tex`

33.1 Historical Background

inc:int:bgr:
sec In this section, we will briefly discuss historical developments that will help put the incompleteness theorems in context. In particular, we will give a very sketchy overview of the history of mathematical logic; and then say a few words about the history of the foundations of mathematics.

The phrase “mathematical logic” is ambiguous. One can interpret the word “mathematical” as describing the subject matter, as in, “the logic of mathematics,” denoting the principles of mathematical reasoning; or as describing the methods, as in “the mathematics of logic,” denoting a mathematical study of the principles of reasoning. The account that follows involves mathematical logic in both senses, often at the same time. digression

The study of logic began, essentially, with Aristotle, who lived approximately 384–322 BCE. His *Categories*, *Prior analytics*, and *Posterior analytics* include systematic studies of the principles of scientific reasoning, including a thorough and systematic study of the syllogism.

CAPÍTULO 33. INTRODUCTION TO INCOMPLETENESS

Aristotle's logic dominated scholastic philosophy through the middle ages; indeed, as late as the eighteenth century, Kant maintained that Aristotle's logic was perfect and in no need of revision. But the theory of the syllogism is far too limited to model anything but the most superficial aspects of mathematical reasoning. A century earlier, Leibniz, a contemporary of Newton's, imagined a complete "calculus" for logical reasoning, and made some rudimentary steps towards designing such a calculus, essentially describing a version of propositional logic.

The nineteenth century was a watershed for logic. In 1854 George Boole wrote *The Laws of Thought*, with a thorough algebraic study of propositional logic that is not far from modern presentations. In 1879 Gottlob Frege published his *Begriffsschrift* (Concept writing) which extends propositional logic with quantifiers and relations, and thus includes first-order logic. In fact, Frege's logical systems included higher-order logic as well, and more. In his *Basic Laws of Arithmetic*, Frege set out to show that all of arithmetic could be derived in his *Begriffsschrift* from purely logical assumption. Unfortunately, these assumptions turned out to be inconsistent, as Russell showed in 1902. But setting aside the inconsistent axiom, Frege more or less invented modern logic singlehandedly, a startling achievement. Quantificational logic was also developed independently by algebraically-minded thinkers after Boole, including Peirce and Schröder.

Let us now turn to developments in the foundations of mathematics. Of course, since logic plays an important role in mathematics, there is a good deal of interaction with the developments just described. For example, Frege developed his logic with the explicit purpose of showing that all of mathematics could be based solely on his logical framework; in particular, he wished to show that mathematics consists of a priori *analytic* truths instead of, as Kant had maintained, a priori *synthetic* ones.

Many take the birth of mathematics proper to have occurred with the Greeks. Euclid's *Elements*, written around 300 B.C., is already a mature representative of Greek mathematics, with its emphasis on rigor and precision. The definitions and proofs in Euclid's *Elements* survive more or less intact in high school geometry textbooks today (to the extent that geometry is still taught in high schools). This model of mathematical reasoning has been held to be a paradigm for rigorous argumentation not only in mathematics but in branches of philosophy as well. (Spinoza even presented moral and religious arguments in the Euclidean style, which is strange to see!)

Calculus was invented by Newton and Leibniz in the seventeenth century. (A fierce priority dispute raged for centuries, but most scholars today hold that the two developments were for the most part independent.) Calculus involves reasoning about, for example, infinite sums of infinitely small quantities; these features fueled criticism by Bishop Berkeley, who argued that belief in God was no less rational than the mathematics of his time. The methods of calculus were widely used in the eighteenth century, for example by Leonhard Euler, who used calculations involving infinite sums with dramatic results.

In the nineteenth century, mathematicians tried to address Berkeley's cri-

33.1. HISTORICAL BACKGROUND

ticisms by putting calculus on a firmer foundation. Efforts by Cauchy, Weierstrass, Bolzano, and others led to our contemporary definitions of limits, continuity, differentiation, and integration in terms of “epsilon and delta,” in other words, devoid of any reference to infinitesimals. Later in the century, mathematicians tried to push further, and explain all aspects of calculus, including the real numbers themselves, in terms of the natural numbers. (Kronecker: “God created the whole numbers, all else is the work of man.”) In 1872, Dedekind wrote “Continuity and the irrational numbers,” where he showed how to “construct” the real numbers as sets of rational numbers (which, as you know, can be viewed as pairs of natural numbers); in 1888 he wrote “Was sind und was sollen die Zahlen” (roughly, “What are the natural numbers, and what should they be?”) which aimed to explain the natural numbers in purely “logical” terms. In 1887 Kronecker wrote “Über den Zahlbegriff” (“On the concept of number”) where he spoke of representing all mathematical object in terms of the integers; in 1889 Giuseppe Peano gave formal, symbolic axioms for the natural numbers.

The end of the nineteenth century also brought a new boldness in dealing with the infinite. Before then, infinitary objects and structures (like the set of natural numbers) were treated gingerly; “infinitely many” was understood as “as many as you want,” and “approaches in the limit” was understood as “gets as close as you want.” But Georg Cantor showed that it was possible to take the infinite at face value. Work by Cantor, Dedekind, and others help to introduce the general set-theoretic understanding of mathematics that is now widely accepted.

This brings us to twentieth century developments in logic and foundations. In 1902 Russell discovered the paradox in Frege’s logical system. In 1904 Zermelo proved Cantor’s well-ordering principle, using the so-called “axiom of choice”; the legitimacy of this axiom prompted a good deal of debate. Between 1910 and 1913 the three volumes of Russell and Whitehead’s *Principia Mathematica* appeared, extending the Fregean program of establishing mathematics on logical grounds. Unfortunately, Russell and Whitehead were forced to adopt two principles that seemed hard to justify as purely logical: an axiom of infinity and an axiom of “reducibility.” In the 1900’s Poincaré criticized the use of “impredicative definitions” in mathematics, and in the 1910’s Brouwer began proposing to refound all of mathematics in an “intuitionistic” basis, which avoided the use of the law of the excluded middle ($\varphi \vee \neg\varphi$).

Strange days indeed! The program of reducing all of mathematics to logic is now referred to as “logicism,” and is commonly viewed as having failed, due to the difficulties mentioned above. The program of developing mathematics in terms of intuitionistic mental constructions is called “intuitionism,” and is viewed as posing overly severe restrictions on everyday mathematics. Around the turn of the century, David Hilbert, one of the most influential mathematicians of all time, was a strong supporter of the new, abstract methods introduced by Cantor and Dedekind: “no one will drive us from the paradise that Cantor has created for us.” At the same time, he was sensitive to foundational criticisms of these new methods (oddly enough, now called “classical”). He proposed a

CAPÍTULO 33. INTRODUCTION TO INCOMPLETENESS

way of having one's cake and eating it too:

1. Represent classical methods with formal axioms and rules; represent mathematical questions as *fórmulas* in an axiomatic system.
2. Use safe, “finitary” methods to prove that these formal deductive systems are consistent.

Hilbert's work went a long way toward accomplishing the first goal. In 1899, he had done this for geometry in his celebrated book *Foundations of geometry*. In subsequent years, he and a number of his students and collaborators worked on other areas of mathematics to do what Hilbert had done for geometry. Hilbert himself gave axiom systems for arithmetic and analysis. Zermelo gave an axiomatization of set theory, which was expanded on by Fraenkel, Skolem, von Neumann, and others. By the mid-1920s, there were two approaches that laid claim to the title of an axiomatization of “all” of mathematics, the *Principia mathematica* of Russell and Whitehead, and what came to be known as Zermelo–Fraenkel set theory.

In 1921, Hilbert set out on a research project to establish the goal of proving these systems to be consistent. He was aided in this project by several of his students, in particular Bernays, Ackermann, and later Gentzen. The basic idea for accomplishing this goal was to cast the question of the possibility of *derivação* of an inconsistency in mathematics as a combinatorial problem about possible sequences of symbols, namely possible sequences of sentences which meet the criterion of being a correct *derivação* of, say, $\varphi \wedge \neg\varphi$ from the axioms of an axiom system for arithmetic, analysis, or set theory. A proof of the impossibility of such a sequence of symbols would—since it is itself a mathematical proof—be formalizable in these axiomatic systems. In other words, there would be some sentence Con which states that, say, arithmetic is consistent. Moreover, this sentence should be provable in the systems in question, especially if its proof requires only very restricted, “finitary” means.

The second aim, that the axiom systems developed would settle every mathematical question, can be made precise in two ways. In one way, we can formulate it as follows: For any sentence φ in the language of an axiom system for mathematics, either φ or $\neg\varphi$ is provable from the axioms. If this were true, then there would be no sentences which can neither be proved nor refuted on the basis of the axioms, no questions which the axioms do not settle. An axiom system with this property is called *complete*. Of course, for any given sentence it might still be a difficult task to determine which of the two alternatives holds. But in principle there should be a method to do so. In fact, for the axiom and *derivação* systems considered by Hilbert, completeness would imply that such a method exists—although Hilbert did not realize this. The second way to interpret the question would be this stronger requirement: that there be a mechanical, computational method which would determine, for a given sentence φ , whether it is derivable from the axioms or not.

In 1931, Gödel proved the two “incompleteness theorems,” which showed that this program could not succeed. There is no axiom system for mathematics

33.2. DEFINITIONS

which is complete, specifically, the sentence that expresses the consistency of the axioms is a sentence which can neither be proved nor refuted.

This struck a lethal blow to Hilbert's original program. However, as is so often the case in mathematics, it also opened up exciting new avenues for research. If there is no one, all-encompassing formal system of mathematics, it makes sense to develop more circumscribed systems and investigate what can be proved in them. It also makes sense to develop less restricted methods of proof for establishing the consistency of these systems, and to find ways to measure how hard it is to prove their consistency. Since Gödel showed that (almost) every formal system has questions it cannot settle, it makes sense to look for "interesting" questions a given formal system cannot settle, and to figure out how strong a formal system has to be to settle them. To the present day, logicians have been pursuing these questions in a new mathematical discipline, the theory of proofs.

-NoValue-

content/incompleteness/introduction/definitions.tex

33.2 Definitions

inc:int:def:sec In order to carry out Hilbert's project of formalizing mathematics and showing that such a formalization is consistent and complete, the first order of business would be that of picking a language, logical framework, and a system of axioms. For our purposes, let us suppose that mathematics can be formalized in a first-order language, O.s., that there is some set of *símbolos de constante*, *símbolos de função*, and *símbolos de predicado* which, together with the connectives and quantifiers of first-order logic, allow us to express the claims of mathematics. Most people agree that such a language exists: the language of set theory, in which $\in$ is the only non-logical symbol. That such a simple language is so expressive is of course a very implausible claim at first sight, and it took a lot of work to establish that practically all of mathematics can be expressed in this very austere vocabulary. To keep things simple, for now, let's restrict our discussion to arithmetic, so the part of mathematics that just deals with the natural numbers $\mathbb{N}$. The natural language in which to express facts of arithmetic is $\mathcal{L}_A$. $\mathcal{L}_A$ contains a single two-place *símbolo de predicado* $<$, a single *símbolo de constante* 0 , one one-place *símbolo de função* $!$, and two two-place *símbolos de função* $+$ and $\times$.

Definição 33.1. A set of *sentenças* Γ is a *theory* if it is closed under entailment, O.s., if $\Gamma = \{\varphi : \Gamma \models \varphi\}$.

There are two easy ways to specify theories. One is as the set of *sentenças* true in some *estrutura*. For instance, consider the *estrutura* for $\mathcal{L}_A$ in which the *domínio* is $\mathbb{N}$ and all non-logical symbols are interpreted as you would expect.

inc:int:def:def:standard-model **Definição 33.2.** The *standard model of arithmetic* is the *estrutura* $\mathfrak{N}$ defined as follows:

1. $|\mathfrak{N}| = \mathbb{N}$
2. $0^{\mathfrak{N}} = 0$
3. $1^{\mathfrak{N}}(n) = n + 1$ for all $n \in \mathbb{N}$
4. $+\mathfrak{N}(n, m) = n + m$ for all $n, m \in \mathbb{N}$
5. $\times^{\mathfrak{N}}(n, m) = n \cdot m$ for all $n, m \in \mathbb{N}$
6. $<^{\mathfrak{N}} = \{\langle n, m \rangle : n \in \mathbb{N}, m \in \mathbb{N}, n < m\}$

Note the difference between ‘ $\times$ ’ and ‘ $\cdot$ ’: $\times$ is a symbol in the language of arithmetic. Of course, we’ve chosen it to remind us of multiplication, but $\times$ is not the multiplication operation but a two-place function symbol (officially, f_1^2). By contrast, $\cdot$ is the ordinary multiplication function. When you see something like $n \cdot m$, we mean the product of the numbers n and m ; when you see something like $x \times y$ we are talking about a term in the language of arithmetic. In the standard model, the function symbol $\times$ is interpreted as the function $\cdot$ on the natural numbers. For addition, we use $+$ as both the function symbol of the language of arithmetic, and the addition function on the natural numbers. Here you have to use the context to determine what is meant.

Definição 33.3. The theory of *true arithmetic* is the set of **sentenças** satisfied in the standard model of arithmetic, O.s.,

$$\mathbf{TA} = \{\varphi : \mathfrak{N} \models \varphi\}.$$

$\mathbf{TA}$ is a theory, for whenever $\mathbf{TA} \models \varphi$, φ is satisfied in every **estrutura** which satisfies $\mathbf{TA}$. Since $\mathfrak{N} \models \mathbf{TA}$, we have that $\mathfrak{N} \models \varphi$, and so $\varphi \in \mathbf{TA}$.

The other way to specify a theory Γ is as the set of **sentenças** entailed by some set of sentences Γ_0 . In that case, Γ is the “closure” of Γ_0 under entailment. Specifying a theory this way is only interesting if Γ_0 is explicitly specified, p.ex., if the **membros** of Γ_0 are listed. At the very least, Γ_0 has to be decidable, O.s., there has to be a computable test for when a **sentença** counts as an element of Γ_0 or not. We call the **sentenças** in Γ_0 *axioms* for Γ , and Γ *axiomatized* by Γ_0 .

Definição 33.4. A theory Γ is *axiomatized* by Γ_0 iff

$$\Gamma = \{\varphi : \Gamma_0 \models \varphi\}$$

33.2. DEFINITIONS

Definição 33.5. The theory **Q** axiomatized by the following sentences is known as “Robinson’s **Q**” and is a very simple theory of arithmetic.

$$\forall x \forall y (x' = y' \rightarrow x = y) \quad (Q_1)$$

$$\forall x \, 0 \neq x' \quad (Q_2)$$

$$\forall x (x = 0 \vee \exists y \, x = y') \quad (Q_3)$$

$$\forall x (x + 0) = x \quad (Q_4)$$

$$\forall x \forall y (x + y') = (x + y)' \quad (Q_5)$$

$$\forall x (x \times 0) = 0 \quad (Q_6)$$

$$\forall x \forall y (x \times y') = ((x \times y) + x) \quad (Q_7)$$

$$\forall x \forall y (x < y \leftrightarrow \exists z (z' + x) = y) \quad (Q_8)$$

The set of **sentenças** $\{Q_1, \dots, Q_8\}$ are the axioms of **Q**, so **Q** consists of all **sentenças** entailed by them:

$$\mathbf{Q} = \{\varphi : \{Q_1, \dots, Q_8\} \models \varphi\}.$$

Definição 33.6. Suppose $\varphi(x)$ is a **fórmula** in $\mathcal{L}_A$ with free variables x and $y_1, \dots, y_n$. Then any **sentença** of the form

$$\forall y_1 \dots \forall y_n ((\varphi(0) \wedge \forall x (\varphi(x) \rightarrow \varphi(x')))) \rightarrow \forall x \varphi(x))$$

is an instance of the *induction schema*.

Peano arithmetic **PA** is the theory axiomatized by the axioms of **Q** together with all instances of the induction schema.

Every instance of the induction schema is true in $\mathfrak{N}$. This is easiest to see [explanation](#) if the **fórmula** φ only has one free **variável** x . Then $\varphi(x)$ defines a subset X_φ of $\mathbb{N}$ in $\mathfrak{N}$. X_φ is the set of all $n \in \mathbb{N}$ such that $\mathfrak{N}, s \models \varphi(x)$ when $s(x) = n$. The corresponding instance of the induction schema is

$$((\varphi(0) \wedge \forall x (\varphi(x) \rightarrow \varphi(x')))) \rightarrow \forall x \varphi(x)).$$

If its antecedent is true in $\mathfrak{N}$, then $0 \in X_\varphi$ and, whenever $n \in X_\varphi$, so is $n + 1$. Since $0 \in X_\varphi$, we get $1 \in X_\varphi$. With $1 \in X_\varphi$ we get $2 \in X_\varphi$. And so on. So for every $n \in \mathbb{N}$, $n \in X_\varphi$. But this means that $\forall x \varphi(x)$ is satisfied in $\mathfrak{N}$.

Both **Q** and **PA** are axiomatized theories. The big question is, how strong are they? For instance, can **PA** prove all the truths about $\mathbb{N}$ that can be expressed in $\mathcal{L}_A$? Specifically, do the axioms of **PA** settle all the questions that can be formulated in $\mathcal{L}_A$?

Another way to put this is to ask: Is **PA** = **TA**? **TA** obviously does prove (O.s., it includes) all the truths about $\mathbb{N}$, and it settles all the questions that can be formulated in $\mathcal{L}_A$, since if φ is a **sentença** in $\mathcal{L}_A$, then either $\mathfrak{N} \models \varphi$ or $\mathfrak{N} \models \neg\varphi$, and so either **TA** $\models \varphi$ or **TA** $\models \neg\varphi$. Call such a theory **completo**.

Definição 33.7. A theory Γ is **completo** iff for every **sentença** φ in its language, either $\Gamma \models \varphi$ or $\Gamma \models \neg\varphi$.

explanation By the Completeness Theorem, $\Gamma \models \varphi$ iff $\Gamma \vdash \varphi$, so Γ is complete iff for every **sentença** φ in its language, either $\Gamma \vdash \varphi$ or $\Gamma \vdash \neg\varphi$.

Another question we are led to ask is this: Is there a computational procedure we can use to test if a **sentença** is in **TA**, in **PA**, or even just in **Q**? We can make this more precise by defining when a set (p.ex., a set of **sentenças**) is **decidível**.

Definição 33.8. A set X is **decidível** iff there is a computational procedure which on input x returns 1 if $x \in X$ and 0 otherwise.

So our question becomes: Is **TA** (**PA**, **Q**) **decidível**?

The answer to all these questions will be: no. None of these theories are decidable. However, this phenomenon is not specific to these particular theories. In fact, *any* theory that satisfies certain conditions is subject to the same results. One of these conditions, which **Q** and **PA** satisfy, is that they are axiomatized by a **decidível** set of axioms.

Definição 33.9. A theory is **axiomatizável** if it is axiomatized by a **decidível** set of axioms.

Exemplo 33.10. Any theory axiomatized by a finite set of **sentenças** is **axiomatizável**, since any finite set is **decidível**. Thus, **Q**, for instance, is **axiomatizável**.

Schematically axiomatized theories like **PA** are also **axiomatizável**. For to test if ψ is among the axioms of **PA**, O.s., to compute the function χ_X where $\chi_X(\psi) = 1$ if ψ is an axiom of **PA** and $= 0$ otherwise, we can do the following: First, check if ψ is one of the axioms of **Q**. If it is, the answer is “yes” and the value of $\chi_X(\psi) = 1$. If not, test if it is an instance of the induction schema. This can be done systematically; in this case, perhaps it’s easiest to see that it can be done as follows: Any instance of the induction schema begins with a number of universal quantifiers, and then a sub-**fórmula** that is a conditional. The consequent of that conditional is $\forall x \varphi(x, y_1, \dots, y_n)$ where x and $y_1, \dots, y_n$ are all the free variables of φ and the initial quantifiers of ψ bind the variables $y_1, \dots, y_n$. Once we have extracted this φ and checked that its free variables match the variables bound by the universal quantifiers at the front and $\forall x$, we go on to check that the antecedent of the conditional matches

$$\varphi(0, y_1, \dots, y_n) \wedge \forall x (\varphi(x, y_1, \dots, y_n) \rightarrow \varphi(x', y_1, \dots, y_n))$$

Again, if it does, ψ is an instance of the induction schema, and if it doesn’t, ψ isn’t.

In answering this question—and the more general question of which theories are complete or decidable—it will be useful to consider also the following definition. Recall that a set X is **enumerável** iff it is empty or if there is **uma sobrejetiva** function $f: \mathbb{N} \rightarrow X$. Such a function is called an enumeration of X .

33.2. DEFINITIONS

Definição 33.11. A set X is called *computavelmente enumerável* (c.e. for short) iff it is empty or it has a computable enumeration.

In addition to *axiomatizabilidade*, another condition on theories to which the incompleteness theorems apply will be that they are strong enough to prove basic facts about computable functions and *decidível* relations. By “basic facts,” we mean *sentenças* which express what the values of computable functions are for each of their arguments. And by “strong enough” we mean that the theories in question count these sentences among its theorems. For instance, consider a prototypical computable function: addition. The value of $+$ for arguments 2 and 3 is 5, O.s., $2 + 3 = 5$. A sentence in the language of arithmetic that expresses that the value of $+$ for arguments 2 and 3 is 5 is: $(\overline{2} + \overline{3}) = \overline{5}$. And, p.ex., $\mathbf{Q}$ proves this sentence. More generally, we would like there to be, for each computable function $f(x_1, x_2)$ a *fórmula* $\varphi_f(x_1, x_2, y)$ in $\mathcal{L}_A$ such that $\mathbf{Q} \vdash \varphi_f(\overline{n_1}, \overline{n_2}, \overline{m})$ whenever $f(n_1, n_2) = m$. In this way, $\mathbf{Q}$ proves that the value of f for arguments n_1, n_2 is m . In fact, we require that it proves a bit more, namely that no other number is the value of f for arguments n_1, n_2 . And the same goes for *decidível* relations. This is made precise in the following two definitions.

Definição 33.12. A *fórmula* $\varphi(x_1, \dots, x_k, y)$ *representa* the function $f: \mathbb{N}^k \rightarrow \mathbb{N}$ in Γ iff whenever $f(n_1, \dots, n_k) = m$, then

1. $\Gamma \vdash \varphi(\overline{n_1}, \dots, \overline{n_k}, \overline{m})$, and
2. $\Gamma \vdash \forall y(\varphi(\overline{n_1}, \dots, \overline{n_k}, y) \rightarrow y = \overline{m})$.

Definição 33.13. A *fórmula* $\varphi(x_1, \dots, x_k)$ *representa* the relation $R \subseteq \mathbb{N}^k$ iff,

1. whenever $R(n_1, \dots, n_k)$, $\Gamma \vdash \varphi(\overline{n_1}, \dots, \overline{n_k})$, and
2. whenever not $R(n_1, \dots, n_k)$, $\Gamma \vdash \neg\varphi(\overline{n_1}, \dots, \overline{n_k})$.

A theory is “strong enough” for the incompleteness theorems to apply if it *representa* all computable functions and all *decidível* relations. $\mathbf{Q}$ and its extensions satisfy this condition, but it will take us a while to establish this—it’s a non-trivial fact about the kinds of things $\mathbf{Q}$ can prove, and it’s hard to show because $\mathbf{Q}$ has only a few axioms from which we’ll have to prove all these facts. However, $\mathbf{Q}$ is a very weak theory. So although it’s hard to prove that $\mathbf{Q}$ represents all computable functions, most interesting theories are stronger than $\mathbf{Q}$, O.s., prove more than $\mathbf{Q}$ does. And if $\mathbf{Q}$ proves something, any stronger theory does; since $\mathbf{Q}$ represents all computable functions, every stronger theory does. This means that many interesting theories meet this condition of the incompleteness theorems. So our hard work will pay off, since it shows that the incompleteness theorems apply to a wide range of theories. Certainly, any theory aiming to formalize “all of mathematics” must prove everything that $\mathbf{Q}$ proves, since it should at the very least be able to capture the results of elementary computations. So any theory that is a candidate

for a theory of “all of mathematics” will be one to which the incompleteness theorems apply.

-NoValue-

content/incompleteness/introduction/overview.tex

33.3 Overview of Incompleteness Results

Hilbert expected that mathematics could be formalized in an **axiomatizável** theory which it would be possible to prove **completo** and **decidível**. Moreover, he aimed to prove the consistency of this theory with very weak, “finitary,” means, which would defend classical mathematics against the challenges of intuitionism. Gödel’s incompleteness theorems showed that these goals cannot be achieved.

Gödel’s first incompleteness theorem showed that a version of Russell and Whitehead’s *Principia Mathematica* is not **completo**. But the proof was actually very general and applies to a wide variety of theories. This means that it wasn’t just that *Principia Mathematica* did not manage to completely capture mathematics, but that *no* acceptable theory does. It took a while to isolate the features of theories that suffice for the incompleteness theorems to apply, and to generalize Gödel’s proof to apply make it depend only on these features. But we are now in a position to state a very general version of the first incompleteness theorem for theories in the language $\mathcal{L}_A$ of arithmetic.

Teorema 33.14. *If Γ is a consistent and **axiomatizável** theory in $\mathcal{L}_A$ which **representa** all computable functions and **decidível** relations, then Γ is not **completo**.*

To say that Γ is not **completo** is to say that for at least one **sentença** φ , $\Gamma \not\vdash \varphi$ and $\Gamma \not\vdash \neg\varphi$. Such a **sentença** is called *independent* (of Γ). We can in fact relatively quickly prove that there must be independent sentences. But the power of Gödel’s proof of the theorem lies in the fact that it exhibits a *specific example* of such an independent **sentença**. The intriguing construction produces a **sentença** γ_Γ , called a *Gödel sentence* for Γ , which is unprovable because in Γ , γ_Γ is equivalent to the claim that γ_Γ is unprovable in Γ . It does so *constructively*, O.s., given an axiomatization of Γ and a description of the **derivação** system, the proof gives a method for actually writing down γ_Γ .

The construction in Gödel’s proof requires that we find a way to express in $\mathcal{L}_A$ the properties of and operations on terms and **fórmulas** of $\mathcal{L}_A$ itself. These include properties such as “ φ is a **sentença**,” “ δ is a **derivação** of φ ,” and operations such as $\varphi[t/x]$. This way must (a) express these properties and relations via a “coding” of symbols and sequences thereof (which is what terms, **fórmulas**, **derivações**, etc. are) as natural numbers (which is what $\mathcal{L}_A$ can talk about). It must (b) do this in such a way that Γ will prove the relevant facts, so we must show that these properties are coded by **decidível** properties

33.4. UNDECIDABILITY AND INCOMPLETENESS

of natural numbers and the operations correspond to computable functions on natural numbers. This is called “arithmetization of syntax.”

Before we investigate how syntax can be arithmetized, however, we will consider the condition that Γ is “strong enough,” O.s., **representa** all computable functions and **decidível** relations. This requires that we give a precise definition of “computable.” This can be done in a number of ways, p.ex., via the model of Turing machines, or as those functions computable by programs in some general-purpose programming language. Since our aim is to **representam** these functions and relations in a theory in the language $\mathcal{L}_A$, however, it is best to pick a simple definition of computability of just numerical functions. This is the notion of *recursive function*. So we will first discuss the recursive functions. We will then show that **Q** already **representa** all recursive functions and relations. This will allow us to apply the incompleteness theorem to specific theories such as **Q** and **PA**, since we will have established that these are examples of theories that are “strong enough.”

The end result of the arithmetization of syntax is a **fórmula** $\text{Prov}_\Gamma(x)$ which, via the coding of **fórmulas** as numbers, expresses provability from the axioms of Γ . Specifically, if φ is coded by the number n , and $\Gamma \vdash \varphi$, then $\Gamma \vdash \text{Prov}_\Gamma(\bar{n})$. This “provability predicate” for Γ allows us also to express, in a certain sense, the consistency of Γ as a **sentença** of $\mathcal{L}_A$: let the “consistency statement” for Γ be the **sentença** $\neg \text{Prov}_\Gamma(\bar{n})$, where we take n to be the code of a contradiction, p.ex., of $\perp$. The second incompleteness theorem states that consistent **axiomatizável** theories also do not prove their own consistency statements. The conditions required for this theorem to apply are a bit more stringent than just that the theory represents all computable functions and **decidível** relations, but we will show that **PA** satisfies them.

-NoValue-

content/incompleteness/introduction/undecidability.tex

33.4 Undecidability and Incompleteness

inc:int:dec:sec Gödel’s proof of the incompleteness theorems require arithmetization of syntax. But even without that we can obtain some nice results just on the assumption that a theory **representa** all **decidível** relations. The proof is a diagonal argument similar to the proof of the undecidability of the halting problem.

Teorema 33.15. *If Γ is a consistent theory that **representa** every **decidível** relation, then Γ is not **decidível**.*

Prova. Suppose Γ were **decidível**. We show that if Γ **representa** every **decidível** relation, it must be inconsistent.

Decidível properties (one-place relations) are represented by **fórmulas** with one free variable. Let $\varphi_0(x), \varphi_1(x), \dots$, be a computable enumeration of all such **fórmulas**. Now consider the following set $D \subseteq \mathbb{N}$:

$$D = \{n : \Gamma \vdash \neg \varphi_n(\bar{n})\}$$

The set D is **decidível**, since we can test if $n \in D$ by first computing $\varphi_n(x)$, and from this $\neg\varphi_n(\bar{n})$. Obviously, substituting the term $\bar{n}$ for every free occurrence of x in $\varphi_n(x)$ and prefixing $\varphi(\bar{n})$ by $\neg$ is a mechanical matter. By assumption, Γ is **decidível**, so we can test if $\neg\varphi(\bar{n}) \in \Gamma$. If it is, $n \in D$, and if it isn't, $n \notin D$. So D is likewise **decidível**.

Since Γ **representa** all **decidível** properties, it **representa** D . And the **fórmulas** which **representam** D in Γ are all among $\varphi_0(x), \varphi_1(x), \dots$. So let d be a number such that $\varphi_d(x)$ **representa** D in Γ . If $d \notin D$, then, since $\varphi_d(x)$ **representa** D , $\Gamma \vdash \neg\varphi_d(\bar{d})$. But that means that d meets the defining condition of D , and so $d \in D$. This contradicts $d \notin D$. So by indirect proof, $d \in D$.

Since $d \in D$, by the definition of D , $\Gamma \vdash \neg\varphi_d(\bar{d})$. On the other hand, since $\varphi_d(x)$ **representa** D in Γ , $\Gamma \vdash \varphi_d(\bar{d})$. Hence, Γ is inconsistent. $\square$

explanation

The preceding theorem shows that no consistent theory that **representa** all **decidível** relations can be **decidível**. We will show that **Q** does **representam** all **decidível** relations; this means that all theories that include **Q**, such as **PA** and **TA**, also do, and hence also are not **decidível**. (Since all these theories are true in the standard model, they are all consistent.)

We can also use this result to obtain a weak version of the first incompleteness theorem. Any theory that is **axiomatizável** and **completo** is **decidível**. Consistent theories that are **axiomatizável** and **representam** all **decidível** properties then cannot be **completo**.

Teorema 33.16. *If Γ is **axiomatizável** and **completo** it is **decidível**.*

Prova. Any inconsistent theory is **decidível**, since inconsistent theories contain all **sentenças**, so the answer to the question “is $\varphi \in \Gamma$ ” is always “yes,” O.s., can be decided.

So suppose Γ is consistent, and furthermore is **axiomatizável**, and **completo**. Since Γ is **axiomatizável**, it is **computavelmente enumerável**. For we can enumerate all the correct **derivações** from the axioms of Γ by a computable function. From a correct **derivação** we can compute the **sentença** it **deriva**, and so together there is a computable function that enumerates all theorems of Γ . A **sentença** is a theorem of Γ iff $\neg\varphi$ is not a theorem, since Γ is consistent and **completo**. We can therefore decide if $\varphi \in \Gamma$ as follows. Enumerate all theorems of Γ . When φ appears on this list, we know that $\Gamma \vdash \varphi$. When $\neg\varphi$ appears on this list, we know that $\Gamma \not\vdash \varphi$. Since Γ is **completo**, one of these cases eventually obtains, so the procedure eventually produces an answer. $\square$

Corolário 33.17. *If Γ is consistent, **axiomatizável**, and **representa** every **decidível** property, it is not **completo**.*

inc:int:dec:
cor:incompleteness

Prova. If Γ were **completo**, it would be **decidível** by the previous theorem (since it is **axiomatizável** and consistent). But since Γ **representa** every **decidível** property, it is not **decidível**, by the first theorem. $\square$

Problema 33.1. Show that **TA** = $\{\varphi : \mathfrak{N} \models \varphi\}$ is not **axiomatizável**. You may assume that **TA** represents all decidable properties.

Once we have established that, p.ex., $\mathbf{Q}$, *representa* all *decidível* properties, the corollary tells us that $\mathbf{Q}$ must be incomplete. However, its proof does not provide an example of an independent *sentença*; it merely shows that such a *sentença* must exist. For this, we have to arithmetize syntax and follow Gödel’s original proof idea. And of course, we still have to show the first claim, namely that $\mathbf{Q}$ does, in fact, *representam* all *decidível* properties.

It should be noted that not every *interesting* theory is incomplete or undecidable. There are many theories that are sufficiently strong to describe interesting mathematical facts that do not satisfy the conditions of Gödel’s result. For instance, $\mathbf{Pres} = \{\varphi \in \mathcal{L}_{A^+} : \mathfrak{N} \models \varphi\}$, the set of *sentenças* of the language of arithmetic without $\times$ true in the standard model, is both complete and decidable. This theory is called Presburger arithmetic, and proves all the truths about natural numbers that can be formulated just with 0 , $!$, and $+$.

Capítulo 34

Arithmetization of Syntax

Note that arithmetization for signed tableaux is not yet available.

-NoValue- _____

[content/incompleteness/arithmetization-syntax/introduction.tex](#)

34.1 Introduction

inc:art:int:
sec

In order to connect computability and logic, we need a way to talk about the objects of logic (symbols, terms, *fórmulas*, *derivações*), operations on them, and their properties and relations, in a way amenable to computational treatment. We can do this directly, by considering computable functions and relations on symbols, sequences of symbols, and other objects built from them. Since the objects of logical syntax are all finite and built from a *enumerável* sets of symbols, this is possible for some models of computation. But other models of computation—such as the recursive functions—are restricted to numbers, their relations and functions. Moreover, ultimately we also want to be able to deal with syntax within certain theories, specifically, in theories formulated in

the language of arithmetic. In these cases it is necessary to *arithmetize* syntax, O.s., to represent syntactic objects, operations on them, and their relations, as numbers, arithmetical functions, and arithmetical relations, respectively. The idea, which goes back to Leibniz, is to assign numbers to syntactic objects.

It is relatively straightforward to assign numbers to symbols as their “codes.” Some symbols pose a bit of a challenge, since, p.ex., there are infinitely many *variáveis*, and even infinitely many *símbolos de função* of each arity n . But of course it’s possible to assign numbers to symbols systematically in such a way that, say, v_2 and v_3 are assigned different codes. Sequences of symbols (such as terms and *fórmulas*) are a bigger challenge. But if we can deal with sequences of numbers purely arithmetically (p.ex., by the powers-of-primes coding of sequences), we can extend the coding of individual symbols to coding of sequences of symbols, and then further to sequences or other arrangements of *fórmulas*, such as *derivações*. This extended coding is called “Gödel numbering.” Every term, *fórmula*, and *derivação* is assigned a Gödel number.

By coding sequences of symbols as sequences of their codes, and by choosing a system of coding sequences that can be dealt with using computable functions, we can then also deal with Gödel numbers using computable functions. In practice, all the relevant functions will be primitive recursive. For instance, computing the length of a sequence and computing the i -th element of a sequence from the code of the sequence are both primitive recursive. If the number coding the sequence is, p.ex., the Gödel number of a *fórmula* φ , we immediately see that the length of a *fórmula* and the (code of the) i -th symbol in a *fórmula* can also be computed from the Gödel number of φ . It is a bit harder to prove that, p.ex., the property of being the Gödel number of a correctly formed term or of a correct *derivação* is primitive recursive. It is nevertheless possible, because the sequences of interest (terms, *fórmulas*, *derivações*) are inductively defined.

As an example, consider the operation of substitution. If φ is a formula, x a variable, and t a term, then $\varphi[t/x]$ is the result of replacing every free occurrence of x in φ by t . Now suppose we have assigned Gödel numbers to φ , x , t —say, k , l , and m , respectively. The same scheme assigns a Gödel number to $\varphi[t/x]$, say, n . This mapping—of k , l , and m to n —is the arithmetical analog of the substitution operation. When the substitution operation maps φ , x , t to $\varphi[t/x]$, the arithmetized substitution function maps the Gödel numbers k , l , m to the Gödel number n . We will see that this function is primitive recursive.

Arithmetization of syntax is not just of abstract interest, although it was originally a non-trivial insight that languages like the language of arithmetic, which do not come with mechanisms for “talking about” languages can, after all, formalize complex properties of expressions. It is then just a small step to ask what a theory in this language, such as Peano arithmetic, can *prove* about its own language (including, p.ex., whether *sentenças* are provable or true). This leads us to the famous limitative theorems of Gödel (about unprovability) and Tarski (the undefinability of truth). But the trick of arithmetizing syntax is also important in order to prove some important results in computability theory, p.ex., about the computational power of theories or the relationship

34.2. CODING SYMBOLS

between different models of computability. The arithmetization of syntax serves as a model for arithmetizing other objects and properties. For instance, it is similarly possible to arithmetize configurations and computations (say, of Turing machines). This makes it possible to simulate computations in one model (p.ex., Turing machines) in another (p.ex., recursive functions).

-NoValue-

[content/incompleteness/arithmetization-syntax/coding-symbols.tex](#)

34.2 Coding Symbols

inc:art:cod:
sec

The basic language $\mathcal{L}$ of first order logic makes use of the symbols

$$\perp \quad \neg \quad \vee \quad \wedge \quad \rightarrow \quad \forall \quad \exists \quad = \quad (\quad) \quad ,$$

together with **enumerável** sets of variables and **símbolos de constante**, and **enumerável** sets of **símbolos de função** and **símbolos de predicado** of arbitrary arity. We can assign *codes* to each of these symbols in such a way that every symbol is assigned a unique number as its code, and no two different symbols are assigned the same number. We know that this is possible since the set of all symbols is **enumerável** and so there is **uma bijeção** between it and the set of natural numbers. But we want to make sure that we can recover the symbol (as well as some information about it, p.ex., the arity of **a símbolo de função**) from its code in a computable way. There are many possible ways of doing this, of course. Here is one such way, which uses primitive recursive functions. (Recall that $\langle n_0, \dots, n_k \rangle$ is the number coding the sequence of numbers $n_0, \dots, n_k$.)

Definição 34.1. If s is a symbol of $\mathcal{L}$, let the *symbol code* c_s be defined as follows:

1. If s is among the logical symbols, c_s is given by the following table:

$\perp$	$\neg$	$\vee$	$\wedge$	$\rightarrow$	$\forall$
$\langle 0, 0 \rangle$	$\langle 0, 1 \rangle$	$\langle 0, 2 \rangle$	$\langle 0, 3 \rangle$	$\langle 0, 4 \rangle$	$\langle 0, 5 \rangle$
$\exists$	$=$	$($	$)$	$,$	
$\langle 0, 6 \rangle$	$\langle 0, 7 \rangle$	$\langle 0, 8 \rangle$	$\langle 0, 9 \rangle$	$\langle 0, 10 \rangle$	

2. If s is the i -th variable v_i , then $c_s = \langle 1, i \rangle$.
3. If s is the i -th **símbolo de constante** c_i , then $c_s = \langle 2, i \rangle$.
4. If s is the i -th n -ary **símbolo de função** f_i^n , then $c_s = \langle 3, n, i \rangle$.
5. If s is the i -th n -ary **símbolo de predicado** P_i^n , then $c_s = \langle 4, n, i \rangle$.

Proposição 34.2. *The following relations are primitive recursive:*

1. $\text{Fn}(x, n)$ iff x is the code of f_i^n for some i , O.s., x is the code of an n -ary **símbolo de função**.

2. $\text{Pred}(x, n)$ iff x is the code of P_i^n for some i or x is the code of $=$ and $n = 2$, O.s., x is the code of an n -ary *símbolo de predicado*.

Definição 34.3. If $s_0, \dots, s_{n-1}$ is a sequence of symbols, its *Gödel number* is $\langle c_{s_0}, \dots, c_{s_{n-1}} \rangle$.

explanation

Note that *codes* and *Gödel numbers* are different things. For instance, the variable v_5 has a code $c_{v_5} = \langle 1, 5 \rangle = 2^2 \cdot 3^6$. But the variable v_5 considered as a term is also a sequence of symbols (of length 1). The *Gödel number* $\#v_5\#$ of the *term* v_5 is $\langle c_{v_5} \rangle = 2^{c_{v_5}+1} = 2^{2^2 \cdot 3^6 + 1}$.

Exemplo 34.4. Recall that if $k_0, \dots, k_{n-1}$ is a sequence of numbers, then the code of the sequence $\langle k_0, \dots, k_{n-1} \rangle$ in the power-of-primes coding is

$$2^{k_0+1} \cdot 3^{k_1+1} \cdot \dots \cdot p_{n-1}^{k_{n-1}+1},$$

where p_i is the i -th prime (starting with $p_0 = 2$). So for instance, the formula $v_0 = 0$, or, more explicitly, $\langle v_0, c_0 \rangle$, has the Gödel number

$$\langle c_=, c_{\langle, c_{v_0}, c_0, \rangle} \rangle.$$

Here, $c_=$ is $\langle 0, 7 \rangle = 2^{0+1} \cdot 3^{7+1}$, c_{v_0} is $\langle 1, 0 \rangle = 2^{1+1} \cdot 3^{0+1}$, etc. So $\#v_0 = \langle v_0, c_0 \rangle\#$ is

$$\begin{aligned} 2^{c_=+1} \cdot 3^{c_{\langle, c_{v_0}, c_0, \rangle}+1} &= \\ 2^{2^1 \cdot 3^8 + 1} \cdot 3^{2^1 \cdot 3^9 + 1} \cdot 5^{2^2 \cdot 3^1 + 1} \cdot 7^{2^1 \cdot 3^{11} + 1} \cdot 11^{2^3 \cdot 3^1 + 1} \cdot 13^{2^1 \cdot 3^{10} + 1} &= \\ 2^{13 \cdot 123} \cdot 3^{39 \cdot 367} \cdot 5^{13} \cdot 7^{354 \cdot 295} \cdot 11^{25} \cdot 13^{118 \cdot 099}. \end{aligned}$$

-NoValue-

[content/incompleteness/arithmetization-syntax/coding-terms.tex](#)

34.3 Coding Terms

explanation

A term is simply a certain kind of sequence of symbols: it is built up inductively from constants and variables according to the formation rules for terms. Since sequences of symbols can be coded as numbers—using a coding scheme for the symbols plus a way to code sequences of numbers—assigning Gödel numbers to terms is not difficult. The challenge is rather to show that the property a number has if it is the Gödel number of a correctly formed term is computable, or in fact primitive recursive.

inc:art:trm:
sec

Variáveis and *símbolos de constante* are the simplest terms, and testing whether x is the Gödel number of such a term is easy: $\text{Var}(x)$ holds if x is $\#v_i\#$ for some i . In other words, x is a sequence of length 1 and its single element $(x)_0$ is the code of some *variável* v_i , O.s., x is $\langle \langle 1, i \rangle \rangle$ for some i . Similarly, $\text{Const}(x)$ holds if x is $\#c_i\#$ for some i . Both of these relations are primitive recursive, since if such an i exists, it must be $< x$:

$$\begin{aligned} \text{Var}(x) &\Leftrightarrow (\exists i < x) x = \langle \langle 1, i \rangle \rangle \\ \text{Const}(x) &\Leftrightarrow (\exists i < x) x = \langle \langle 2, i \rangle \rangle \end{aligned}$$

34.3. CODING TERMS

*inc:art:trm:
prop:term-primrec*

Proposição 34.5. *The relations $\text{Term}(x)$ and $\text{ClTerm}(x)$ which hold iff x is the Gödel number of a term or a closed term, respectively, are primitive recursive.*

Prova. A sequence of symbols s is a term iff there is a sequence $s_0, \dots, s_{k-1} = s$ of terms which records how the term s was formed from **símbolos de constante** and **variáveis** according to the formation rules for terms. To express that such a putative formation sequence follows the formation rules it has to be the case that, for each $i < k$, either

1. s_i is a **variável** v_j , or
2. s_i is a **símbolo de constante** c_j , or
3. s_i is built from n terms $t_1, \dots, t_n$ occurring prior to place i using an n -place **símbolo de função** f_j^n .

To show that the corresponding relation on Gödel numbers is primitive recursive, we have to express this condition primitive recursively, O.s., using primitive recursive functions, relations, and bounded quantification.

Suppose y is the number that codes the sequence $s_0, \dots, s_{k-1}$, O.s., $y = \langle \#s_0\#, \dots, \#s_{k-1}\# \rangle$. It codes a formation sequence for the term with Gödel number x iff for all $i < k$:

1. $\text{Var}((y)_i)$, or
2. $\text{Const}((y)_i)$, or
3. there is an n and a number $z = \langle z_1, \dots, z_n \rangle$ such that each z_l is equal to some $(y)_{i'}$ for $i' < i$ and

$$(y)_i = \#f_j^n(\# \frown \text{flatten}(z) \frown \#)^\#,$$

and moreover $(y)_{k-1} = x$. (The function $\text{flatten}(z)$ turns the sequence $\langle \#t_1\#, \dots, \#t_n\# \rangle$ into $\#t_1, \dots, t_n\#$ and is primitive recursive.)

The indices j, n , the Gödel numbers z_l of the terms t_l , and the code z of the sequence $\langle z_1, \dots, z_n \rangle$, in (3) are all less than y . We can replace k above with $\text{len}(y)$. Hence we can express “ y is the code of a formation sequence of the term with Gödel number x ” in a way that shows that this relation is primitive recursive.

We now just have to convince ourselves that there is a primitive recursive bound on y . But if x is the Gödel number of a term, it must have a formation sequence with at most $\text{len}(x)$ terms (since every term in the formation sequence of s must start at some place in s , and no two subterms can start at the same place). The Gödel number of each subterm of s is of course $\leq x$. Hence, there always is a formation sequence with code $\leq p_{k-1}^{k(x+1)}$, where $k = \text{len}(x)$.

For ClTerm , simply leave out the clause for **variáveis**. □

Problema 34.1. Show that the function $\text{flatten}(z)$, which turns the sequence $\langle \#t_1, \dots, \#t_n \rangle$ into $\#t_1, \dots, \#t_n$, is primitive recursive.

Proposição 34.6. *The function $\text{num}(n) = \#n$ is primitive recursive.*

*inc:art:trm:
prop:num-primrec*

Prova. We define $\text{num}(n)$ by primitive recursion:

$$\begin{aligned} \text{num}(0) &= \#0 \\ \text{num}(n+1) &= \#(\# \frown \text{num}(n) \frown \#). \end{aligned} \quad \square$$

-NoValue- _____
content/incompleteness/arithmeticization-syntax/coding-formulas.tex

34.4 Coding Fórmulas

Once we have defined the relation $\text{Term}(x)$ primitive recursively, we can use it to define the corresponding relation for **fórmulas**, $\text{Frm}(x)$ primitive recursively.

*inc:art:frm:
sec*

Proposição 34.7. *The relation $\text{Atom}(x)$ which holds iff x is the Gödel number of an atomic **fórmula**, is primitive recursive.*

Prova. The number x is the Gödel number of an atomic **fórmula** iff one of the following holds:

1. There are $n, j < x$, and $z < x$ such that for each $i < n$, $\text{Term}((z)_i)$ and $x =$

$$\#P_j^n(\# \frown \text{flatten}(z) \frown \#).$$

2. There are $z_1, z_2 < x$ such that $\text{Term}(z_1)$, $\text{Term}(z_2)$, and $x =$

$$\#(\# \frown z_1 \frown \#, \# \frown z_2 \frown \#).$$

3. $x = \# \perp \#$.

4. $x = \# \top \#$.

□

Proposição 34.8. *The relation $\text{Frm}(x)$ which holds iff x is the Gödel number of a **fórmula** is primitive recursive.*

*inc:art:frm:
prop:frm-primrec*

Prova. A sequence of symbols s is a **fórmula** iff there is formation sequence $s_0, \dots, s_{k-1} = s$ of **fórmula** which records how s was formed from atomic **fórmulas** according to the formation rules. The code for each s_i (and indeed of the code of the sequence $\langle s_0, \dots, s_{k-1} \rangle$) is less than the code x of s . □

Problema 34.2. Give a detailed proof of **Proposição 34.8** along the lines of the first proof of **Proposição 34.5**.

34.5. SUBSTITUTION

*inc:art:frm:
prop:freeocc-primrec*

Proposição 34.9. *The relation $\text{FreeOcc}(x, z, i)$, which holds iff the i -th symbol of the formula with Gödel number x is a free occurrence of the variable with Gödel number z , is primitive recursive.*

Prova. Exercise. □

Problema 34.3. Prove **Proposição 34.9**. You may make use of the fact that any substring of a fórmula which is a fórmula is a sub-fórmula of it.

Proposição 34.10. *The property $\text{Sent}(x)$ which holds iff x is the Gödel number of a sentença is primitive recursive.*

Prova. A sentença is a fórmula without free occurrences of variáveis. So $\text{Sent}(x)$ holds iff

$$(\forall i < \text{len}(x)) (\forall z < x) ((\exists j < z) z = \#v_j^\# \rightarrow \neg \text{FreeOcc}(x, z, i)). \quad \square$$

-NoValue-

[content/incompleteness/arithmeticization-syntax/substitution.tex](#)

34.5 Substitution

*inc:art:sub:
sec*

Recall that substitution is the operation of replacing all free occurrences of a variável u in a fórmula φ by a term t , written $\varphi[t/u]$. This operation, when carried out on Gödel numbers of variáveis, fórmulas, and terms, is primitive recursive.

*inc:art:sub:
prop:subst-primrec*

Proposição 34.11. *There is a primitive recursive function $\text{Subst}(x, y, z)$ with the property that*

$$\text{Subst}(\# \varphi^\#, \# t^\#, \# u^\#) = \# \varphi[t/u]^\#.$$

Prova. We can then define a function hSubst by primitive recursion as follows:

$$\begin{aligned} \text{hSubst}(x, y, z, 0) &= \Lambda \\ \text{hSubst}(x, y, z, i + 1) &= \begin{cases} \text{hSubst}(x, y, z, i) \frown y & \text{if } \text{FreeOcc}(x, z, i) \\ \text{append}(\text{hSubst}(x, y, z, i), (x)_i) & \text{otherwise.} \end{cases} \end{aligned}$$

$\text{Subst}(x, y, z)$ can now be defined as $\text{hSubst}(x, y, z, \text{len}(x))$. □

*inc:art:sub:
prop:free-for*

Proposição 34.12. *The relation $\text{FreeFor}(x, y, z)$, which holds iff the term with Gödel number y is livre para the variable with Gödel number z in the formula with Gödel number x , is primitive recursive.*

Prova. Exercise. □

Problema 34.4. Prove [Proposição 34.12](#)

-NoValue-

[content/incompleteness/arithmetization-syntax/proofs-in-lk.tex](#)

34.6 Derivações in LK

explanation In order to arithmetize [derivações](#), we must represent [derivações](#) as numbers. inc:art:plk:sec Since [derivações](#) are trees of sequents where each inference carries also a label, a recursive representation is the most obvious approach: we represent a [derivação](#) as a tuple, the components of which are the end-sequent, the label, and the representations of the sub-[derivações](#) leading to the premises of the last inference.

Definição 34.13. If Γ is a finite sequence of [sentenças](#), $\Gamma = \langle \varphi_1, \dots, \varphi_n \rangle$, then $\# \Gamma = \langle \# \varphi_1, \dots, \# \varphi_n \rangle$.

If $\Gamma \Rightarrow \Delta$ is a sequent, then a Gödel number of $\Gamma \Rightarrow \Delta$ is

$$\# \Gamma \Rightarrow \Delta = \langle \# \Gamma, \# \Delta \rangle$$

If π is a [derivação](#) in **LK**, then $\# \pi$ is defined as follows:

1. If π consists only of the initial sequent $\Gamma \Rightarrow \Delta$, then $\# \pi$ is

$$\langle 0, \# \Gamma \Rightarrow \Delta \rangle.$$

2. If π ends in an inference with one or two premises, has $\Gamma \Rightarrow \Delta$ as its conclusion, and π_1 and π_2 are the immediate subproof ending in the premise of the last inference, then $\# \pi$ is

$$\langle 1, \# \pi_1, \# \Gamma \Rightarrow \Delta, k \rangle \text{ or } \langle 2, \# \pi_1, \# \pi_2, \# \Gamma \Rightarrow \Delta, k \rangle,$$

respectively, where k is given by the following table according to which rule was used in the last inference:

Rule:	WL	WR	CL	CR	XL	XR
k :	1	2	3	4	5	6
Rule:	$\neg$ L	$\neg$ R	$\wedge$ L	$\wedge$ R	$\vee$ L	$\vee$ R
k :	7	8	9	10	11	12
Rule:	$\rightarrow$ L	$\rightarrow$ R	$\forall$ L	$\forall$ R	$\exists$ L	$\exists$ R
k :	13	14	15	16	17	18
Rule:	Cut	=				
k :	19	20				

Exemplo 34.14. Consider the very simple [derivação](#)

34.6. DERIVAÇÕES IN LK

$$\frac{\frac{\varphi \Rightarrow \varphi}{\varphi \wedge \psi \Rightarrow \varphi} \wedge L}{\Rightarrow (\varphi \wedge \psi) \rightarrow \varphi} \rightarrow R$$

The Gödel number of the initial sequent would be $p_0 = \langle 0, \# \varphi \Rightarrow \varphi^\# \rangle$. The Gödel number of the *derivação* ending in the conclusion of $\wedge L$ would be $p_1 = \langle 1, p_0, \# \varphi \wedge \psi \Rightarrow \varphi^\#, 9 \rangle$ (1 since $\wedge L$ has one premise, the Gödel number of the conclusion $\varphi \wedge \psi \Rightarrow \varphi$, and 9 is the number coding $\wedge L$). The Gödel number of the entire *derivação* then is $\langle 1, p_1, \# \Rightarrow (\varphi \wedge \psi) \rightarrow \varphi^\#, 14 \rangle$, O.s.,

$$\langle 1, \langle 1, \langle 0, \# \varphi \Rightarrow \varphi^\# \rangle, \# \varphi \wedge \psi \Rightarrow \varphi^\#, 9 \rangle, \# \Rightarrow (\varphi \wedge \psi) \rightarrow \varphi^\#, 14 \rangle.$$

Having settled on a representation of *derivações*, we must also show that we can manipulate such *derivações* primitive recursively, and express their essential properties and relations so. Some operations are simple: p.ex., given a Gödel number p of a *derivação*, $\text{EndSeq}(p) = (p)_{(p)_0+1}$ gives us the Gödel number of its end-sequent and $\text{LastRule}(p) = (p)_{(p)_0+2}$ the code of its last rule. The property $\text{Sequent}(s)$ defined by

$$\text{len}(s) = 2 \wedge (\forall i < \text{len}((s)_0) + \text{len}((s)_1)) \text{Sent}(((s)_0 \frown (s)_1)_i)$$

holds of s iff s is the Gödel number of a sequent consisting of *sentenças*. Some are much harder. We'll at least sketch how to do this. The goal is to show that the relation “ π is a *derivação* of φ from Γ ” is a primitive recursive relation of the Gödel numbers of π and φ .

inc:art:plk: prop:followsby **Proposição 34.15.** *The property $\text{Correct}(p)$ which holds iff the last inference in the *derivação* π with Gödel number p is correct, is primitive recursive.*

Prova. $\Gamma \Rightarrow \Delta$ is an initial sequent if either there is a *sentença* φ such that $\Gamma \Rightarrow \Delta$ is $\varphi \Rightarrow \varphi$, or there is a term t such that $\Gamma \Rightarrow \Delta$ is $\emptyset \Rightarrow t = t$. In terms of Gödel numbers, $\text{InitSeq}(s)$ holds iff

$$\begin{aligned} & (\exists x < s) (\text{Sent}(x) \wedge s = \langle \langle x \rangle, \langle x \rangle \rangle) \vee \\ & (\exists t < s) (\text{Term}(t) \wedge s = \langle 0, \langle \# = (\# \frown t \frown \#, \# \frown t \frown \#) \# \rangle \rangle). \end{aligned}$$

We also have to show that for each rule of inference R the relation $\text{FollowsBy}_R(p)$ is primitive recursive, where $\text{FollowsBy}_R(p)$ holds iff p is the Gödel number of *derivação* π , and the end-sequent of π follows by a correct application of R from the immediate sub-*derivações* of π .

A simple case is that of the $\wedge R$ rule. If π ends in a correct $\wedge R$ inference, it looks like this:

$$\frac{\begin{array}{c} \vdots \\ \pi_1 \\ \vdots \\ \Gamma \Rightarrow \Delta, \varphi \end{array} \quad \begin{array}{c} \vdots \\ \pi_2 \\ \vdots \\ \Gamma \Rightarrow \Delta, \psi \end{array}}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi} \wedge R$$

So, the last inference in the *derivação* π is a correct application of $\wedge R$ iff there are sequences of *sentenças* Γ and Δ as well as two *sentenças* φ and ψ such that the end-sequent of π_1 is $\Gamma \Rightarrow \Delta, \varphi$, the end-sequent of π_2 is $\Gamma \Rightarrow \Delta, \psi$, and the end-sequent of π is $\Gamma \Rightarrow \Delta, \varphi \wedge \psi$. We just have to translate this into Gödel numbers. If $s = \# \Gamma \Rightarrow \Delta^\#$ then $(s)_0 = \# \Gamma^\#$ and $(s)_1 = \# \Delta^\#$. So, $\text{FollowsBy}_{\wedge R}(p)$ holds iff

$$\begin{aligned} & (\exists g < p) (\exists d < p) (\exists a < p) (\exists b < p) \\ & \text{EndSequent}(p) = \langle g, d \frown \langle \#(\# \frown a \frown \# \wedge^\# \frown b \frown \#)^\# \rangle \rangle \wedge \\ & \text{EndSequent}((p)_1) = \langle g, d \frown \langle a \rangle \rangle \wedge \\ & \text{EndSequent}((p)_2) = \langle g, d \frown \langle b \rangle \rangle \wedge \\ & (p)_0 = 2 \wedge \text{LastRule}(p) = 10. \end{aligned}$$

The individual lines express, respectively, “there is a sequence (Γ) with Gödel number g , there is a sequence (Δ) with Gödel number d , a *fórmula* (φ) with Gödel number a , and a *fórmula* (ψ) with Gödel number b ,” such that “the end-sequent of π is $\Gamma \Rightarrow \Delta, \varphi \wedge \psi$,” “the end-sequent of π_1 is $\Gamma \Rightarrow \Delta, \varphi$,” “the end-sequent of π_2 is $\Gamma \Rightarrow \Delta, \psi$,” and “ π has two immediate subderivations and the last inference rule is $\wedge R$ (with number 10).”

The last inference in π is a correct application of $\exists R$ iff there are sequences Γ and Δ , a *fórmula* φ , a variable x , and a term t , such that the end-sequent of π is $\Gamma \Rightarrow \Delta, \exists x \varphi$ and the end-sequent of π_1 is $\Gamma \Rightarrow \Delta, \varphi[t/x]$. So in terms of Gödel numbers, we have $\text{FollowsBy}_{\exists R}(p)$ iff

$$\begin{aligned} & (\exists g < p) (\exists d < p) (\exists a < p) (\exists x < p) (\exists t < p) \\ & \text{EndSequent}(p) = \langle g, d \frown \langle \# \exists^\# \frown x \frown a \rangle \rangle \wedge \\ & \text{EndSequent}((p)_1) = \langle g, d \frown \langle \text{Subst}(a, t, x) \rangle \rangle \wedge \\ & (p)_0 = 1 \wedge \text{LastRule}(p) = 18. \end{aligned}$$

We then define $\text{Correct}(p)$ as

$$\begin{aligned} & \text{Sequent}(\text{EndSequent}(p)) \wedge \\ & [(\text{LastRule}(p) = 1 \wedge \text{FollowsBy}_{\text{WL}}(p)) \vee \dots \vee \\ & (\text{LastRule}(p) = 20 \wedge \text{FollowsBy}_=(p)) \vee \\ & (p)_0 = 0 \wedge \text{InitialSeq}(\text{EndSequent}(p))] \end{aligned}$$

The first line ensures that the end-sequent of d is actually a sequent consisting of *sentenças*. The last line covers the case where p is just an initial sequent. $\square$

Problema 34.5. Define the following properties as in **Proposição 34.15**:

1. $\text{FollowsBy}_{\text{Cut}}(p)$,
2. $\text{FollowsBy}_{\rightarrow L}(p)$,
3. $\text{FollowsBy}_=(p)$,

34.7. DERIVAÇÕES IN NATURAL DEDUCTION

4. FollowsBy $_{\forall R}(p)$.

For the last one, you will have to also show that you can test primitive recursively if the last inference of the **derivação** with Gödel number p satisfies the eigenvariable condition, O.s., the eigenvariable a of the $\forall R$ does not occur in the end-sequent.

inc:art:plk: prop:deriv **Proposição 34.16.** *The relation $\text{Deriv}(p)$ which holds if p is the Gödel number of a correct **derivação** π , is primitive recursive.*

Prova. A **derivação** π is correct if every one of its inferences is a correct application of a rule, O.s., if every one of its sub-**derivações** ends in a correct inference. So, $\text{Deriv}(d)$ iff

$$(\forall i < \text{len}(\text{SubtreeSeq}(p))) \text{Correct}((\text{SubtreeSeq}(p))_i). \quad \square$$

Proposição 34.17. *Suppose Γ is a primitive recursive set of **sentenças**. Then the relation $\text{Prf}_\Gamma(x, y)$ expressing “ x is the code of a **derivação** π of $\Gamma_0 \Rightarrow \varphi$ for some finite $\Gamma_0 \subseteq \Gamma$ and y is the Gödel number of φ ” is primitive recursive.*

Prova. Suppose “ $y \in \Gamma$ ” is given by the primitive recursive predicate $R_\Gamma(y)$. We have to show that $\text{Prf}_\Gamma(x, y)$ which holds iff y is the Gödel number of a sentence φ and x is the code of an **LK-derivação** with end-sequent $\Gamma_0 \Rightarrow \varphi$ is primitive recursive.

By the previous proposition, the property $\text{Deriv}(x)$ which holds iff x is the code of a correct **derivação** π in **LK** is primitive recursive. If x is such a code, then $\text{EndSequent}(x)$ is the code of the end-sequent of π , and so $(\text{EndSequent}(x))_0$ is the code of the left side of the end sequent and $(\text{EndSequent}(x))_1$ the right side. So we can express “the right side of the end-sequent of π is φ ” as $\text{len}((\text{EndSequent}(x))_1) = 1 \wedge ((\text{EndSequent}(x))_1)_0 = x$. The left side of the end-sequent of π is of course automatically finite, we just have to express that every sentence in it is in Γ . Thus we can define $\text{Prf}_\Gamma(x, y)$ by

$$\begin{aligned} \text{Prf}_\Gamma(x, y) \Leftrightarrow & \text{Deriv}(x) \wedge \\ & (\forall i < \text{len}((\text{EndSequent}(x))_0)) R_\Gamma(((\text{EndSequent}(x))_0)_i) \wedge \\ & \text{len}((\text{EndSequent}(x))_1) = 1 \wedge ((\text{EndSequent}(x))_1)_0 = y. \quad \square \end{aligned}$$

-NoValue-

[content/incompleteness/arithmeticization-syntax/proofs-in-nd.tex](#)

34.7 Derivações in Natural Deduction

inc:art:pnd: sec In order to arithmetize **derivações**, we must represent **derivações** as numbers. *explanation* Since **derivações** are trees of **fórmulas** where each inference carries one or two labels, a recursive representation is the most obvious approach: we represent a **derivação** as a tuple, the components of which are the number of immediate

sub-*derivações* leading to the premises of the last inference, the representations of these sub-*derivações*, and the end-*fórmula*, the discharge label of the last inference, and a number indicating the type of the last inference.

Definição 34.18. If δ is a *derivação* in natural deduction, then $\# \delta^\#$ is defined inductively as follows:

1. If δ consists only of the assumption φ , then $\# \delta^\#$ is $\langle 0, \# \varphi^\#, n \rangle$. The number n is 0 if it is an *não descarregada* assumption, and the numerical label otherwise.
2. If δ ends in an inference with zero, one, two, or three premises, then $\# \delta^\#$ is

$$\begin{aligned} &\langle 0, \# \varphi^\#, n, k \rangle, \\ &\langle 1, \# \delta_1^\#, \# \varphi^\#, n, k \rangle, \\ &\langle 2, \# \delta_1^\#, \# \delta_2^\#, \# \varphi^\#, n, k \rangle, \text{ or} \\ &\langle 3, \# \delta_1^\#, \# \delta_2^\#, \# \delta_3^\#, \# \varphi^\#, n, k \rangle, \end{aligned}$$

respectively. Here $\delta_1, \delta_2, \delta_3$ are the sub-*derivações* ending in the premise(s) of the last inference in δ , φ is the conclusion of the last inference in δ , n is the discharge label of the last inference (0 if the inference does not discharge any assumptions), and k is given by the following table according to which rule was used in the last inference.

Rule:	$\wedge$ Intro	$\wedge$ Elim	$\vee$ Intro	$\vee$ Elim
k :	1	2	3	4
Rule:	$\rightarrow$ Intro	$\rightarrow$ Elim	$\neg$ Intro	$\neg$ Elim
k :	5	6	7	8
Rule:	$\perp_I$	$\perp_C$	$\forall$ Intro	$\forall$ Elim
k :	9	10	11	12
Rule:	$\exists$ Intro	$\exists$ Elim	$=$ Intro	$=$ Elim
k :	13	14	15	16

Exemplo 34.19. Consider the very simple *derivação*

$$\frac{\frac{[\varphi \wedge \psi]^1}{\varphi} \wedge \text{Elim}}{(\varphi \wedge \psi) \rightarrow \varphi} \rightarrow \text{Intro}$$

The Gödel number of the assumption would be $d_0 = \langle 0, \# \varphi \wedge \psi^\#, 1 \rangle$. The Gödel number of the *derivação* ending in the conclusion of $\wedge$ Elim would be $d_1 = \langle 1, d_0, \# \varphi^\#, 0, 2 \rangle$ (1 since $\wedge$ Elim has one premise, the Gödel number of conclusion φ , 0 because no assumption is discharged, and 2 is the number coding $\wedge$ Elim). The Gödel number of the entire *derivação* then is $\langle 1, d_1, \# ((\varphi \wedge \psi) \rightarrow \varphi)^\#, 1, 5 \rangle$, O.s.,

$$\langle 1, \langle 1, \langle 0, \# (\varphi \wedge \psi)^\#, 1 \rangle, \# \varphi^\#, 0, 2 \rangle, \# ((\varphi \wedge \psi) \rightarrow \varphi)^\#, 1, 5 \rangle.$$

34.7. DERIVAÇÕES IN NATURAL DEDUCTION

Having settled on a representation of *derivações*, we must also show that explanation we can manipulate Gödel numbers of such *derivações* primitive recursively, and express their essential properties and relations. Some operations are simple: p.ex., given a Gödel number d of a *derivação*, $\text{EndFmla}(d) = (d)_{(d)_0+1}$ gives us the Gödel number of its end-*fórmula*, $\text{DischargeLabel}(d) = (d)_{(d)_0+2}$ gives us the discharge label and $\text{LastRule}(d) = (d)_{(d)_0+3}$ the number indicating the type of the last inference. Some are much harder. We'll at least sketch how to do this. The goal is to show that the relation “ δ is a *derivação* of φ from Γ ” is a primitive recursive relation of the Gödel numbers of δ and φ .

Proposição 34.20. *The following relations are primitive recursive:*

1. φ occurs as an assumption in δ with label n .
2. All assumptions in δ with label n are of the form φ (O.s., we can *descarregar* the assumption φ using label n in δ).

Prova. We have to show that the corresponding relations between Gödel numbers of *fórmulas* and Gödel numbers of *derivações* are primitive recursive.

1. We want to show that $\text{Assum}(x, d, n)$, which holds if x is the Gödel number of an assumption of the *derivação* with Gödel number d labelled n , is primitive recursive. This is the case if the *derivação* with Gödel number $\langle 0, x, n \rangle$ is a sub-*derivação* of d . Note that the way we code *derivações* is a special case of the coding of trees introduced in *Seção 29.12*, so the primitive recursive function $\text{SubtreeSeq}(d)$ gives a sequence of Gödel numbers of all sub-*derivações* of d (of length at most d). So we can define

$$\text{Assum}(x, d, n) \Leftrightarrow (\exists i < d) (\text{SubtreeSeq}(d))_i = \langle 0, x, n \rangle.$$

2. We want to show that $\text{Discharge}(x, d, n)$, which holds if all assumptions with label n in the *derivação* with Gödel number d all are the *fórmula* with Gödel number x . But this relation holds iff $(\forall y < d) (\text{Assum}(y, d, n) \rightarrow y = x)$. $\square$

inc:art:pnd: prop:followsby **Proposição 34.21.** *The property $\text{Correct}(d)$ which holds iff the last inference in the *derivação* δ with Gödel number d is correct, is primitive recursive.*

Prova. Here we have to show that for each rule of inference R the relation $\text{FollowsBy}_R(d)$ is primitive recursive, where $\text{FollowsBy}_R(d)$ holds iff d is the Gödel number of *derivação* δ , and the end-*fórmula* of δ follows by a correct application of R from the immediate sub-*derivações* of δ .

A simple case is that of the $\wedge$ Intro rule. If δ ends in a correct $\wedge$ Intro inference, it looks like this:

$$\frac{\begin{array}{c} \vdots \\ \delta_1 \\ \vdots \\ \varphi \end{array} \quad \begin{array}{c} \vdots \\ \delta_2 \\ \vdots \\ \psi \end{array}}{\varphi \wedge \psi} \wedge \text{Intro}$$

Then the Gödel number d of δ is $\langle 2, d_1, d_2, \#(\varphi \wedge \psi)^\#, 0, k \rangle$ where $\text{EndFmla}(d_1) = \# \varphi^\#, \text{EndFmla}(d_2) = \# \psi^\#, n = 0$, and $k = 1$. So we can define $\text{FollowsBy}_{\wedge \text{Intro}}(d)$ as

$$\begin{aligned} (d)_0 &= 2 \wedge \text{DischargeLabel}(d) = 0 \wedge \text{LastRule}(d) = 1 \wedge \\ &\quad \text{EndFmla}(d) = \\ &\quad \#(\# \frown \text{EndFmla}((d)_1) \frown \# \wedge \# \frown \text{EndFmla}((d)_2) \frown \#)^\#. \end{aligned}$$

Another simple example is the $=\text{Intro}$ rule. This has no premises, so $(d)_0 = 0$, like assumptions. It also has no discharge label, O.s., $n = 0$. However, φ must be of the form $t = t$, for a closed term t . Here, a primitive recursive definition is

$$\begin{aligned} (d)_0 &= 0 \wedge \text{DischargeLabel}(d) = 0 \wedge \\ &\quad (\exists t < d) (\text{CTerm}(t) \wedge \text{EndFmla}(d) = \\ &\quad \quad \#(=\# \frown t \frown \#, \# \frown t \frown \#)^\#). \end{aligned}$$

For a more complicated example, $\text{FollowsBy}_{\rightarrow \text{Intro}}(d)$ holds iff the end-*fórmula* of δ is of the form $(\varphi \rightarrow \psi)$, where the end-*fórmula* of δ_1 is ψ , and any assumption in δ labelled n is of the form φ . We can express this primitive recursively by

$$\begin{aligned} (d)_0 &= 1 \wedge \\ &\quad (\exists a < d) (\text{Discharge}(a, (d)_1, \text{DischargeLabel}(d)) \wedge \\ &\quad \text{EndFmla}(d) = (\#(\# \frown a \frown \# \rightarrow \# \frown \text{EndFmla}((d)_1) \frown \#)^\#)) \end{aligned}$$

(Think of a as the Gödel number of φ).

For another example, consider $\exists \text{Intro}$. Here, the last inference in δ is correct iff there is a *fórmula* φ , a closed term t and a *variável* x such that $\varphi[t/x]$ is the end-*fórmula* of the *derivação* δ_1 and $\exists x \varphi$ is the conclusion of the last inference. So, $\text{FollowsBy}_{\exists \text{Intro}}(d)$ holds iff

$$\begin{aligned} (d)_0 &= 1 \wedge \text{DischargeLabel}(d) = 0 \wedge \\ &\quad (\exists a < d) (\exists x < d) (\exists t < d) (\text{CTerm}(t) \wedge \text{Var}(x) \wedge \\ &\quad \text{Subst}(a, t, x) = \text{EndFmla}((d)_1) \wedge \text{EndFmla}(d) = (\# \exists \# \frown x \frown a)^\#). \end{aligned}$$

34.7. DERIVAÇÕES IN NATURAL DEDUCTION

We then define $\text{Correct}(d)$ as

$$\begin{aligned} & \text{Sent}(\text{EndFmla}(d)) \wedge \\ & (\text{LastRule}(d) = 1 \wedge \text{FollowsBy}_{\wedge\text{Intro}}(d)) \vee \dots \vee \\ & (\text{LastRule}(d) = 16 \wedge \text{FollowsBy}_{=\text{Elim}}(d)) \vee \\ & (\exists n < d) (\exists x < d) (d = \langle 0, x, n \rangle). \end{aligned}$$

The first line ensures that the end-fórmula of d is a sentence. The last line covers the case where d is just an assumption. $\square$

Problema 34.6. Define the following properties as in **Proposição 34.21**:

1. $\text{FollowsBy}_{\rightarrow\text{Elim}}(d)$,
2. $\text{FollowsBy}_{=\text{Elim}}(d)$,
3. $\text{FollowsBy}_{\vee\text{Elim}}(d)$,
4. $\text{FollowsBy}_{\forall\text{Intro}}(d)$.

For the last one, you will have to also show that you can test primitive recursively if the last inference of the derivação with Gödel number d satisfies the eigenvariable condition, O.s., the eigenvariable a of the $\forall\text{Intro}$ inference occurs neither in the end-fórmula of d nor in an open assumption of d . You may use the primitive recursive predicate OpenAssum from **Proposição 34.23** for this.

inc:art:pnd: prop:deriv **Proposição 34.22.** *The relation $\text{Deriv}(d)$ which holds if d is the Gödel number of a correct derivação δ , is primitive recursive.*

Prova. A derivação δ is correct if every one of its inferences is a correct application of a rule, O.s., if every one of its sub-derivações ends in a correct inference. So, $\text{Deriv}(d)$ iff

$$(\forall i < \text{len}(\text{SubtreeSeq}(d))) \text{Correct}((\text{SubtreeSeq}(d))_i) \quad \square$$

inc:art:pnd: prop:openassum **Proposição 34.23.** *The relation $\text{OpenAssum}(z, d)$ that holds if z is the Gödel number of a não descarregada assumption φ of the derivação δ with Gödel number d , is primitive recursive.*

Prova. An occurrence of an assumption is **descarregada** if it occurs with label n in a sub-derivação of δ that ends in a rule with discharge label n . So φ is a **não descarregada** assumption of δ if at least one of its occurrences is not **descarregada** in δ . We must be careful: δ may contain both **descarregada** and **não descarregada** occurrences of φ .

Consider a sequence $\delta_0, \dots, \delta_k$ where $\delta_0 = \delta$, δ_k is the assumption $[\varphi]^n$ (for some n), and δ_{i+1} is an immediate sub-derivação of δ_i . If such a sequence exists in which no δ_i ends in an inference with discharge label n , then φ is a **não descarregada** assumption of δ .

The primitive recursive function $\text{SubtreeSeq}(d)$ provides us with a sequence of Gödel numbers of all sub-*derivações* of δ . Any sequence of Gödel numbers of sub-*derivações* of δ is a subsequence of it. Being a subsequence of is a primitive recursive relation: $\text{Subseq}(s, s')$ holds iff $(\forall i < \text{len}(s)) \exists j < \text{len}(s') (s)_i = (s')_j$. Being an immediate sub-*derivação* is as well: $\text{Subderiv}(d, d')$ iff $(\exists j < (d')_0) d = (d')_j$. So we can define $\text{OpenAssum}(z, d)$ by

$$\begin{aligned} (\exists s < \text{SubtreeSeq}(d)) (\text{Subseq}(s, \text{SubtreeSeq}(d)) \wedge (s)_0 = d \wedge \\ (\exists n < d) ((s)_{\text{len}(s)-1} = \langle 0, z, n \rangle \wedge \\ (\forall i < (\text{len}(s) - 1)) (\text{Subderiv}((s)_{i+1}, (s)_i) \wedge \\ \text{DischargeLabel}((s)_i) \neq n))). \quad \square \end{aligned}$$

Proposição 34.24. *Suppose Γ is a primitive recursive set of *sentenças*. Then the relation $\text{Prf}_\Gamma(x, y)$ expressing “ x is the code of a *derivação* δ of φ from *não descarregada* assumptions in Γ and y is the Gödel number of φ ” is primitive recursive.* inc:art:pnd:
prop:prf-prim-rec

Prova. Suppose “ $y \in \Gamma$ ” is given by the primitive recursive predicate $R_\Gamma(y)$. We have to show that $\text{Prf}_\Gamma(x, y)$ which holds iff y is the Gödel number of a sentence φ and x is the code of a natural deduction *derivação* with end fórmula φ and all *não descarregada* assumptions in Γ is primitive recursive.

By **Proposição 34.22**, the property $\text{Deriv}(x)$ which holds iff x is the Gödel number of a correct *derivação* δ in natural deduction is primitive recursive. Thus we can define $\text{Prf}_\Gamma(x, y)$ by

$$\begin{aligned} \text{Prf}_\Gamma(x, y) \Leftrightarrow \text{Deriv}(x) \wedge \text{EndFmla}(x) = y \wedge \\ (\forall z < x) (\text{OpenAssum}(z, x) \rightarrow R_\Gamma(z)). \quad \square \end{aligned}$$

-NoValue-

 <content/incompleteness/arithmetization-syntax/proofs-in-ax.tex>

34.8 Axiomatic Derivações

explanation In order to arithmetize axiomatic *derivações*, we must represent *derivações* as numbers. Since *derivações* are simply sequences of *fórmulas*, the obvious approach is to code every *derivação* as the code of the sequence of codes of *fórmulas* in it. inc:art:pax:
sec

Definição 34.25. If δ is an axiomatic *derivação* consisting of *fórmulas* $\varphi_1, \dots, \varphi_n$, then $\# \delta \#$ is

$$\langle \# \varphi_1 \#, \dots, \# \varphi_n \# \rangle.$$

Exemplo 34.26. Consider the very simple *derivação*:

34.8. AXIOMATIC DERIVAÇÕES

1. $\psi \rightarrow (\psi \vee \varphi)$
2. $(\psi \rightarrow (\psi \vee \varphi)) \rightarrow (\varphi \rightarrow (\psi \rightarrow (\psi \vee \varphi)))$
3. $\varphi \rightarrow (\psi \rightarrow (\psi \vee \varphi))$

The Gödel number of this **derivação** would be

$$\langle \# \psi \rightarrow (\psi \vee \varphi)^\#, \\ \# (\psi \rightarrow (\psi \vee \varphi)) \rightarrow (\varphi \rightarrow (\psi \rightarrow (\psi \vee \varphi)))^\#, \\ \# \varphi \rightarrow (\psi \rightarrow (\psi \vee \varphi))^\# \rangle.$$

Having settled on a representation of **derivações**, we must also show that we can manipulate such **derivações** primitive recursively, and express their essential properties and relations so. Some operations are simple: p.ex., given a Gödel number d of a **derivação**, $(d)_{\text{len}(d)-1}$ gives us the Gödel number of its end-**fórmula**. Some are much harder. We'll at least sketch how to do this. The goal is to show that the relation “ δ is a **derivação** of φ from I ” is primitive recursive in the Gödel numbers of δ and φ .

explanation

inc:art:paz:
prop:followsby

Proposição 34.27. *The following relations are primitive recursive:*

1. φ is an axiom.
2. The i -th line in δ is justified by *modus ponens*
3. The i -th line in δ is justified by QR.
4. δ is a correct **derivação**.

Prova. We have to show that the corresponding relations between Gödel numbers of **fórmulas** and Gödel numbers of **derivações** are primitive recursive.

1. We have a given list of axiom schemas, and φ is an axiom if it is of the form given by one of these schemas. Since the list of schemas is finite, it suffices to show that we can test primitive recursively, for each axiom schema, if φ is of that form. For instance, consider the axiom schema

$$\psi \rightarrow (\chi \rightarrow \psi).$$

φ is an instance of this axiom schema if there are **fórmulas** ψ and χ such that we obtain φ when we concatenate ‘(’ with ψ with ‘ $\rightarrow$ ’ with ‘(’ with χ with ‘ $\rightarrow$ ’ with ψ and with ‘)’. We can test the corresponding property of the Gödel number n of φ , since concatenation of sequences is primitive recursive and the Gödel numbers of ψ and χ must be smaller than the Gödel number of φ , since when the relation holds, both ψ and χ are sub-**fórmulas** of φ . Hence, we can define:

$$\text{IsAx}_{\psi \rightarrow (\chi \rightarrow \psi)}(n) \Leftrightarrow (\exists b < n) (\exists c < n) (\text{Sent}(b) \wedge \text{Sent}(c) \wedge \\ n = \#(\# \frown b \frown \# \rightarrow \# \frown \#(\# \frown c \frown \# \rightarrow \# \frown b \frown \#))^\#).$$

If we have such a definition for each axiom schema, their disjunction defines the property $\text{IsAx}(n)$, “ n is the Gödel number of an axiom.”

2. The i -th line in δ is justified by modus ponens iff there are lines j and $k < i$ where the **sentença** on line j is some formula φ , the sentence on line k is $\varphi \rightarrow \psi$, and the sentence on line i is ψ .

$$\begin{aligned} \text{MP}(d, i) &\Leftrightarrow (\exists j < i) (\exists k < i) \\ (d)_k &= \#(\# \frown (d)_j \frown \# \rightarrow \# \frown (d)_i \frown \#) \# \end{aligned}$$

Since bounded quantification, concatenation, and $=$ are primitive recursive, this defines a primitive recursive relation.

3. A line in δ is justified by QR if it is of the form $\psi \rightarrow \forall x \varphi(x)$, a preceding line is $\psi \rightarrow \varphi(c)$ for some **símbolo de constante** c , and c does not occur in ψ . This is the case iff
- a) there is a **sentença** ψ and
 - b) a **fórmula** $\varphi(x)$ with a single variable x free so that
 - c) line i contains $\psi \rightarrow \forall x \varphi(x)$
 - d) some line $j < i$ contains $\psi \rightarrow \varphi[c/x]$ for a constant c
 - e) which does not occur in ψ .

All of these can be tested primitive recursively, since the Gödel numbers of ψ , $\varphi(x)$, and x are less than the Gödel number of the formula on line i , and that of a less than the Gödel number of the formula on line j :

$$\begin{aligned} \text{QR}_1(d, i) &\Leftrightarrow (\exists b < (d)_i) (\exists x < (d)_i) (\exists a < (d)_i) (\exists c < (d)_j) (\\ &\quad \text{Var}(x) \wedge \text{Const}(c) \wedge \\ (d)_i &= \#(\# \frown b \frown \# \rightarrow \# \frown \# \forall \# \frown x \frown a \frown \#) \# \wedge \\ (d)_j &= \#(\# \frown b \frown \# \rightarrow \# \frown \text{Subst}(a, c, x) \frown \#) \# \wedge \\ &\quad \text{Sent}(b) \wedge \text{Sent}(\text{Subst}(a, c, x)) \wedge (\forall k < \text{len}(b)) (b)_k \neq (c)_0 \end{aligned}$$

Here we assume that c and x are the Gödel numbers of the variable and constant considered as terms (O.s., not their symbol codes). We test that x is the only free variable of $\varphi(x)$ by testing if $\varphi(x)[c/x]$ is a **sentença**, and ensure that c does not occur in ψ by requiring that every symbol of ψ is different from c .

We leave the other version of QR as an exercise.

4. d is the Gödel number of a correct **derivação** iff every line in it is an axiom, or justified by modus ponens or QR. Hence:

$$\text{Deriv}(d) \Leftrightarrow (\forall i < \text{len}(d)) (\text{IsAx}((d)_i) \vee \text{MP}(d, i) \vee \text{QR}(d, i)) \quad \square$$

Problema 34.7. Define the following relations as in **Proposição 34.27**:

1. $\text{IsAx}_{\varphi \rightarrow (\psi \rightarrow (\varphi \wedge \psi))}(n)$,

2. $\text{IsAx}_{\forall x \varphi(x) \rightarrow \varphi(t)}(n)$,
3. $\text{QR}_2(d, i)$ (for the other version of QR).

Proposição 34.28. *Suppose Γ is a primitive recursive set of *sentenças*. Then the relation $\text{Prf}_\Gamma(x, y)$ expressing “ x is the code of a *derivação* δ of φ from Γ and y is the Gödel number of φ ” is primitive recursive.*

Prova. Suppose “ $y \in \Gamma$ ” is given by the primitive recursive predicate $R_\Gamma(y)$. We have to show that the relation $\text{Prf}_\Gamma(x, y)$ is primitive recursive, where $\text{Prf}_\Gamma(x, y)$ holds iff y is the Gödel number of a *sentença* φ and x is the code of a *derivação* of φ from Γ .

By the previous proposition, the property $\text{Deriv}(x)$ which holds iff x is the code of a correct *derivação* δ is primitive recursive. However, that definition did not take into account the set Γ as an additional way to justify lines in the *derivação*. Our primitive recursive test of whether a line is justified by QR also left out of consideration the requirement that the constant c is not allowed to occur in Γ . It is possible to amend our definition so that it takes into account Γ directly, but it is easier to use Deriv and the deduction theorem. $\Gamma \vdash \varphi$ iff there is some finite list of *sentenças* $\psi_1, \dots, \psi_n \in \Gamma$ such that $\{\psi_1, \dots, \psi_n\} \vdash \varphi$. And by the deduction theorem, this is the case if $\vdash (\psi_1 \rightarrow (\psi_2 \rightarrow \dots (\psi_n \rightarrow \varphi) \dots))$. Whether a *sentença* with Gödel number z is of this form can be tested primitive recursively. So, instead of considering x as the Gödel number of a *derivação* of the *sentença* with Gödel number y from Γ , we consider x as the Gödel number of a *derivação* of a nested conditional of the above form from $\emptyset$.

First, if we have a sequence of *sentenças*, we can primitive recursively form the conditional with all these sentences as antecedents and given *sentença* as consequent:

$$\begin{aligned} \text{hCond}(s, y, 0) &= y \\ \text{hCond}(s, y, n+1) &= \#(\# \frown (s)_n \frown \# \rightarrow \# \frown \text{Cond}(s, y, n) \frown \#) \# \\ \text{Cond}(s, y) &= \text{hCond}(s, y, \text{len}(s)) \end{aligned}$$

So we can define $\text{Prf}_\Gamma(x, y)$ by

$$\begin{aligned} \text{Prf}_\Gamma(x, y) \Leftrightarrow & (\exists s < \text{sequenceBound}(x, x)) (\\ & (x)_{\text{len}(x)-1} = \text{Cond}(s, y) \wedge \\ & (\forall i < \text{len}(s)) (s)_i \in \Gamma \wedge \\ & \text{Deriv}(x)). \end{aligned}$$

The bound on s is given by considering that each $(s)_i$ is the Gödel number of a sub-*fórmula* of the last line of the *derivação*, O.s., is less than $(x)_{\text{len}(x)-1}$. The number of antecedents $\psi \in \Gamma$, O.s., the length of s , is less than the length of the last line of x . $\square$

Capítulo 35

Representability in $\mathbf{Q}$

-NoValue-

 content/incompleteness/representability-in-q/introduction.tex

35.1 Introduction

The incompleteness theorems apply to theories in which basic facts about computable functions can be expressed and proved. We will describe a very minimal such theory called “ $\mathbf{Q}$ ” (or, sometimes, “Robinson’s Q ,” after Raphael Robinson). We will say what it means for a function to be *representable* in $\mathbf{Q}$, and then we will prove the following:

inc:req:int:
sec

A function is representable in $\mathbf{Q}$ if and only if it is computable.

For one thing, this provides us with another model of computability. But we will also use it to show that the set $\{\varphi : \mathbf{Q} \vdash \varphi\}$ is not decidable, by reducing the halting problem to it. By the time we are done, we will have proved much stronger things than this.

The language of $\mathbf{Q}$ is the language of arithmetic; $\mathbf{Q}$ consists of the following axioms (to be used in conjunction with the other axioms and rules of first-order logic with *predicado de identidade*):

$$\forall x \forall y (x' = y' \rightarrow x = y) \quad (Q_1)$$

$$\forall x \, 0 \neq x' \quad (Q_2)$$

$$\forall x (x = 0 \vee \exists y \, x = y') \quad (Q_3)$$

$$\forall x (x + 0) = x \quad (Q_4)$$

$$\forall x \forall y (x + y') = (x + y)' \quad (Q_5)$$

$$\forall x (x \times 0) = 0 \quad (Q_6)$$

$$\forall x \forall y (x \times y') = ((x \times y) + x) \quad (Q_7)$$

$$\forall x \forall y (x < y \leftrightarrow \exists z (z' + x) = y) \quad (Q_8)$$

35.1. INTRODUCTION

For each natural number n , define the numeral $\bar{n}$ to be the term $o''\dots'$ where there are n tick marks in all. So, $\bar{0}$ is the **símbolo de constante** o by itself, $\bar{1}$ is o' , $\bar{2}$ is o'' , etc.

As a theory of arithmetic, $\mathbf{Q}$ is *extremely* weak; for example, you can't even prove very simple facts like $\forall x x \neq x'$ or $\forall x \forall y (x + y) = (y + x)$. But we will see that much of the reason that $\mathbf{Q}$ is so interesting is *because* it is so weak. In fact, it is just barely strong enough for the incompleteness theorem to hold. Another reason $\mathbf{Q}$ is interesting is because it has a *finite* set of axioms.

A stronger theory than $\mathbf{Q}$ (called *Peano arithmetic* $\mathbf{PA}$) is obtained by adding a schema of induction to $\mathbf{Q}$:

$$(\varphi(o) \wedge \forall x (\varphi(x) \rightarrow \varphi(x'))) \rightarrow \forall x \varphi(x)$$

where $\varphi(x)$ is any formula. If $\varphi(x)$ contains free **variáveis** other than x , we add universal quantifiers to the front to bind all of them (so that the corresponding instance of the induction schema is a **sentença**). For instance, if $\varphi(x, y)$ also contains the **variável** y free, the corresponding instance is

$$\forall y ((\varphi(o) \wedge \forall x (\varphi(x) \rightarrow \varphi(x'))) \rightarrow \forall x \varphi(x))$$

Using instances of the induction schema, one can prove much more from the axioms of $\mathbf{PA}$ than from those of $\mathbf{Q}$. In fact, it takes a good deal of work to find “natural” statements about the natural numbers that can't be proved in Peano arithmetic!

inc:req:int:
defn:representable-fn

Definição 35.1. A function $f(x_0, \dots, x_k)$ from the natural numbers to the natural numbers is said to be *representable in $\mathbf{Q}$* if there is a formula $\varphi_f(x_0, \dots, x_k, y)$ such that whenever $f(n_0, \dots, n_k) = m$, $\mathbf{Q}$ proves

inc:req:int:
defn:rep:a
inc:req:int:
defn:rep:b

1. $\varphi_f(\bar{n}_0, \dots, \bar{n}_k, \bar{m})$
2. $\forall y (\varphi_f(\bar{n}_0, \dots, \bar{n}_k, y) \rightarrow \bar{m} = y)$.

There are other ways of stating the definition; for example, we could equivalently require that $\mathbf{Q}$ proves $\forall y (\varphi_f(\bar{n}_0, \dots, \bar{n}_k, y) \leftrightarrow y = \bar{m})$.

inc:req:int:
thm:representable-iff-comp

Teorema 35.2. A function is representable in $\mathbf{Q}$ if and only if it is computable.

There are two directions to proving the theorem. The left-to-right direction is fairly straightforward once arithmetization of syntax is in place. The other direction requires more work. Here is the basic idea: we pick “general recursive” as a way of making “computable” precise, and show that every general recursive function is representable in $\mathbf{Q}$. Recall that a function is general recursive if it can be defined from zero, the successor function succ , and the projection functions P_i^n , using composition, primitive recursion, and regular minimization. So one way of showing that every general recursive function is representable in $\mathbf{Q}$ is to show that the basic functions are representable, and

whenever some functions are representable, then so are the functions defined from them using composition, primitive recursion, and regular minimization. In other words, we might show that the basic functions are representable, and that the representable functions are “closed under” composition, primitive recursion, and regular minimization. This guarantees that every general recursive function is representable.

It turns out that the step where we would show that representable functions are closed under primitive recursion is hard. In order to avoid this step, we show first that in fact we can do without primitive recursion. That is, we show that every general recursive function can be defined from basic functions using composition and regular minimization alone. To do this, we show that primitive recursion can actually be done by a specific regular minimization. However, for this to work, we have to add some additional basic functions: addition, multiplication, and the characteristic function of the identity relation $\chi_=$. Then, we can prove the theorem by showing that all of *these* basic functions are representable in $\mathbf{Q}$, and the representable functions are closed under composition and regular minimization.

-NoValue-

[content/incompleteness/representability-in-q/representable-comp.tex](#)

35.2 Functions Representable in $\mathbf{Q}$ are Computable

We’ll prove that every function that is representable in $\mathbf{Q}$ is computable. We first have to establish a lemma about functions representable in $\mathbf{Q}$. inc:req:rpc:
sec

Lema 35.3. *If $f(x_0, \dots, x_k)$ is representable in $\mathbf{Q}$, there is a fórmula $\varphi(x_0, \dots, x_k, y)$ such that* (rpc:
lem:rep-q

$$\mathbf{Q} \vdash \varphi_f(\overline{n_0}, \dots, \overline{n_k}, \overline{m}) \quad \text{iff} \quad m = f(n_0, \dots, n_k).$$

Prova. The “if” part is [Definição 35.1\(1\)](#). The “only if” part is seen as follows: Suppose $\mathbf{Q} \vdash \varphi_f(\overline{n_0}, \dots, \overline{n_k}, \overline{m})$ but $m \neq f(n_0, \dots, n_k)$. Let $l = f(n_0, \dots, n_k)$. By [Definição 35.1\(1\)](#), $\mathbf{Q} \vdash \varphi_f(\overline{n_0}, \dots, \overline{n_k}, \overline{l})$. By [Definição 35.1\(2\)](#), $\forall y (\varphi_f(\overline{n_0}, \dots, \overline{n_k}, y) \rightarrow \overline{l} = y)$. Using logic and the assumption that $\mathbf{Q} \vdash \varphi_f(\overline{n_0}, \dots, \overline{n_k}, \overline{m})$, we get that $\mathbf{Q} \vdash \overline{l} = \overline{m}$. On the other hand, by [Lema 35.14](#), $\mathbf{Q} \vdash \overline{l} \neq \overline{m}$. So $\mathbf{Q}$ is inconsistent. But that is impossible, since $\mathbf{Q}$ is satisfied by the standard model (see [Definição 33.2](#)), $\mathfrak{N} \models \mathbf{Q}$, and satisfiable theories are always consistent by the Soundness Theorem ([Corolários 19.31, 20.29, 21.31 e 22.38](#)). □

Lema 35.4. *Every function that is representable in $\mathbf{Q}$ is computable.*

Prova. Let’s first give the intuitive idea for why this is true. To compute f , we do the following. List all the possible derivações δ in the language of arithmetic. This is possible to do mechanically. For each one, check if it is a derivação of a fórmula of the form $\varphi_f(\overline{n_0}, \dots, \overline{n_k}, \overline{m})$ (the fórmula representing f in $\mathbf{Q}$ from [Lema 35.3](#)). If it is, $m = f(n_0, \dots, n_k)$ by [Lema 35.3](#), and we’ve found the value of f . The search terminates because $\mathbf{Q} \vdash \varphi_f(\overline{n_0}, \dots, \overline{n_k}, \overline{f(n_0, \dots, n_k)})$, so eventually we find a δ of the right sort.

35.3. THE BETA FUNCTION LEMMA

This is not quite precise because our procedure operates on *derivações* and *fórmulas* instead of just on numbers, and we haven't explained exactly why "listing all possible *derivações*" is mechanically possible. But as we've seen, it is possible to code terms, *fórmulas*, and *derivações* by Gödel numbers. We've also introduced a precise model of computation, the general recursive functions. And we've seen that the relation $\text{Prf}_{\mathbf{Q}}(d, y)$, which holds iff d is the Gödel number of a *derivação* of the *fórmula* with Gödel number y from the axioms of $\mathbf{Q}$, is (primitive) recursive. Other primitive recursive functions we'll need are *num* (Proposição 34.6) and *Subst* (Proposição 34.11). From these, it is possible to define f by minimization; thus, f is recursive.

First, define

$$A(n_0, \dots, n_k, m) = \text{Subst}(\text{Subst}(\dots \text{Subst}(\ulcorner \varphi_f \urcorner, \text{num}(n_0), \ulcorner x_0 \urcorner), \dots), \text{num}(n_k), \ulcorner x_k \urcorner), \text{num}(m), \ulcorner y \urcorner)$$

This looks complicated, but it's just the function $A(n_0, \dots, n_k, m) = \ulcorner \varphi_f(\overline{n_0}, \dots, \overline{n_k}, \overline{m}) \urcorner$.

Now, consider the relation $R(n_0, \dots, n_k, s)$ which holds if $(s)_0$ is the Gödel number of a *derivação* from $\mathbf{Q}$ of $\varphi_f(\overline{n_0}, \dots, \overline{n_k}, (s)_1)$:

$$R(n_0, \dots, n_k, s) \quad \text{iff} \quad \text{Prf}_{\mathbf{Q}}((s)_0, A(n_0, \dots, n_k, (s)_1))$$

If we can find an s such that $R(n_0, \dots, n_k, s)$ holds, we have found a pair of numbers— $(s)_0$ and $(s)_1$ —such that $(s)_0$ is the Gödel number of a *derivação* of $A_f(\overline{n_0}, \dots, \overline{n_k}, (s)_1)$. So looking for s is like looking for the pair d and m in the informal proof. And a computable function that "looks for" such an s can be defined by regular minimization. Note that R is regular: for every $n_0, \dots, n_k$, there is a *derivação* δ of $\mathbf{Q} \vdash \varphi_f(\overline{n_0}, \dots, \overline{n_k}, f(n_0, \dots, n_k))$, so $R(n_0, \dots, n_k, s)$ holds for $s = \langle \ulcorner \delta \urcorner, f(n_0, \dots, n_k) \rangle$. So, we can write f as

$$f(n_0, \dots, n_k) = (\mu s R(n_0, \dots, n_k, s))_1. \quad \square$$

-NoValue-

content/incompleteness/representability-in-q/beta-function.tex

35.3 The Beta Function Lemma

inc:req:bet:
sec

In order to show that we can carry out primitive recursion if addition, multiplication, and $\chi_=_$ are available, we need to develop functions that handle sequences. (If we had exponentiation as well, our task would be easier.) When we had primitive recursion, we could define things like the " n -th prime," and pick a fairly straightforward coding. But here we do not have primitive recursion—in fact we want to show that we can do primitive recursion using minimization—so we need to be more clever.

Lema 35.5. *There is a function $\beta(d, i)$ such that for every sequence $a_0, \dots, a_n$ there is a number d , such that for every $i \leq n$, $\beta(d, i) = a_i$. Moreover, β can be defined from the basic functions using just composition and regular minimization.* inc:req:bet:
lem:beta

Think of d as coding the sequence $\langle a_0, \dots, a_n \rangle$, and $\beta(d, i)$ returning the i -th element. (Note that this “coding” does *not* use the power-of-primes coding we’re already familiar with!). The lemma is fairly minimal; it doesn’t say we can concatenate sequences or append elements, or even that we can *compute* d from $a_0, \dots, a_n$ using functions definable by composition and regular minimization. All it says is that there is a “decoding” function such that every sequence is “coded.”

The use of the notation β is Gödel’s. To repeat, the hard part of proving the lemma is defining a suitable β using the seemingly restricted resources, O.s., using just composition and minimization—however, we’re allowed to use addition, multiplication, and $\chi_{=}$. There are various ways to prove this lemma, but one of the cleanest is still Gödel’s original method, which used a number-theoretic fact called Sunzi’s Theorem (traditionally, the “Chinese Remainder Theorem”).

Definição 35.6. Two natural numbers a and b are *relatively prime* iff their greatest common divisor is 1; in other words, they have no other divisors in common.

Definição 35.7. Natural numbers a and b are *congruent modulo* c , $a \equiv b \pmod{c}$, iff $c \mid (a - b)$, O.s., a and b have the same remainder when divided by c .

Here is Sunzi’s Theorem:

Teorema 35.8. *Suppose $x_0, \dots, x_n$ are (pairwise) relatively prime. Let $y_0, \dots, y_n$ be any numbers. Then there is a number z such that*

$$\begin{aligned} z &\equiv y_0 \pmod{x_0} \\ z &\equiv y_1 \pmod{x_1} \\ &\vdots \\ z &\equiv y_n \pmod{x_n}. \end{aligned}$$

Here is how we will use Sunzi’s Theorem: if $x_0, \dots, x_n$ are bigger than $y_0, \dots, y_n$ respectively, then we can take z to code the sequence $\langle y_0, \dots, y_n \rangle$. To recover y_i , we need only divide z by x_i and take the remainder. To use this coding, we will need to find suitable values for $x_0, \dots, x_n$.

A couple of observations will help us in this regard. Given $y_0, \dots, y_n$, let

$$\begin{aligned} j &= \max(n, y_0 + 1, \dots, y_n + 1), \\ m &= \text{lcm}(1, \dots, j), \end{aligned}$$

35.3. THE BETA FUNCTION LEMMA

and let

$$\begin{aligned} x_0 &= 1 + m \\ x_1 &= 1 + 2 \cdot m \\ x_2 &= 1 + 3 \cdot m \\ &\vdots \\ x_n &= 1 + (n + 1) \cdot m \end{aligned}$$

Then two things are true:

- 1. $x_0, \dots, x_n$ are relatively prime.
- 2. For each i , $y_i < x_i$.

To see that (1) is true, note that if p is a prime number and $p \mid x_i$ and $p \mid x_k$, then $p \mid 1 + (i + 1)m$ and $p \mid 1 + (k + 1)m$. But then p divides their difference,

$$(1 + (i + 1)m) - (1 + (k + 1)m) = (i - k)m.$$

Since p divides $1 + (i + 1)m$, it can't divide m as well (otherwise, the first division would leave a remainder of 1). So p divides $i - k$, since p divides $(i - k)m$. But $|i - k|$ is at most n , and we have chosen $j \geq n$, so this implies that $p \mid m$, again a contradiction. So there is no prime number dividing both x_i and x_k . Clause (2) is easy: we have $y_i < j \leq m < x_i$.

Now let us prove the β function lemma. Remember that we can use 0, successor, plus, times, $\chi_=$, projections, and any function defined from them using composition and minimization applied to regular functions. We can also use a relation if its characteristic function is so definable. As before we can show that these relations are closed under Boolean combinations and bounded quantification; for example:

$$\begin{aligned} \text{not}(x) &= \chi_=(x, 0) \\ (\min x \leq z) R(x, y) &= \mu x (R(x, y) \vee x = z) \\ (\exists x \leq z) R(x, y) &\Leftrightarrow R((\min x \leq z) R(x, y), y) \end{aligned}$$

We can then show that all of the following are also definable without primitive recursion:

- 1. The pairing function, $J(x, y) = \frac{1}{2}[(x + y)(x + y + 1)] + x$;
- 2. the projection functions

$$\begin{aligned} K(z) &= (\min x \leq z) (\exists y \leq z) z = J(x, y), \\ L(z) &= (\min y \leq z) (\exists x \leq z) z = J(x, y); \end{aligned}$$

- 3. the less-than relation $x < y$;

4. the divisibility relation $x \mid y$;
5. the function $\text{rem}(x, y)$ which returns the remainder when y is divided by x .

Now define

$$\begin{aligned}\beta^*(d_0, d_1, i) &= \text{rem}(1 + (i + 1)d_1, d_0) \text{ and} \\ \beta(d, i) &= \beta^*(K(d), L(d), i).\end{aligned}$$

This is the function we want. Given $a_0, \dots, a_n$ as above, let

$$j = \max(n, a_0 + 1, \dots, a_n + 1),$$

and let $d_1 = \text{lcm}(1, \dots, j)$. By (1) above, we know that $1 + d_1, 1 + 2d_1, \dots, 1 + (n+1)d_1$ are relatively prime, and by (2) that all are greater than $a_0, \dots, a_n$. By Sunzi's Theorem there is a value d_0 such that for each i ,

$$d_0 \equiv a_i \pmod{1 + (i + 1)d_1}$$

and so (because d_1 is greater than a_i),

$$a_i = \text{rem}(1 + (i + 1)d_1, d_0).$$

Let $d = J(d_0, d_1)$. Then for each $i \leq n$, we have

$$\begin{aligned}\beta(d, i) &= \beta^*(d_0, d_1, i) \\ &= \text{rem}(1 + (i + 1)d_1, d_0) \\ &= a_i\end{aligned}$$

which is what we need. This completes the proof of the β -function lemma.

Problema 35.1. Show that the relations $x < y$, $x \mid y$, and the function $\text{rem}(x, y)$ can be defined without primitive recursion. You may use 0, successor, plus, times, $\chi=$, projections, and bounded minimization and quantification.

-NoValue-

 [content/incompleteness/representability-in-q/prim-rec.tex](#)

35.4 Simulating Primitive Recursion

Now we can show that definition by primitive recursion can be “simulated” by regular minimization using the beta function. Suppose we have $f(\vec{x})$ and $g(\vec{x}, y, z)$. Then the function $h(x, \vec{z})$ defined from f and g by primitive recursion is

$$\begin{aligned}h(\vec{x}, 0) &= f(\vec{x}) \\ h(\vec{x}, y + 1) &= g(\vec{x}, y, h(\vec{x}, y)).\end{aligned}$$

We need to show that h can be defined from f and g using just composition and regular minimization, using the basic functions and functions defined from them using composition and regular minimization (such as β).

35.5. BASIC FUNCTIONS ARE REPRESENTABLE IN $\mathbf{Q}$

*inc:req:pri:
lem:prim-rec* **Lema 35.9.** *If h can be defined from f and g using primitive recursion, it can be defined from f , g , the functions zero, succ, P_i^n , add, mult, $\chi_=$, using composition and regular minimization.*

Prova. First, define an auxiliary function $\hat{h}(\vec{x}, y)$ which returns the least number d such that d codes a sequence which satisfies

1. $(d)_0 = f(\vec{x})$, and
2. for each $i < y$, $(d)_{i+1} = g(\vec{x}, i, (d)_i)$,

where now $(d)_i$ is short for $\beta(d, i)$. In other words, $\hat{h}$ returns the sequence $\langle h(\vec{x}, 0), h(\vec{x}, 1), \dots, h(\vec{x}, y) \rangle$. We can write $\hat{h}$ as

$$\hat{h}(\vec{x}, y) = \mu d (\beta(d, 0) = f(\vec{x}) \wedge (\forall i < y) \beta(d, i+1) = g(\vec{x}, i, \beta(d, i))).$$

Note: no primitive recursion is needed here, just minimization. The function we minimize is regular because of the beta function lemma **Lema 35.5**.

But now we have

$$h(\vec{x}, y) = \beta(\hat{h}(\vec{x}, y), y),$$

so h can be defined from the basic functions using just composition and regular minimization. $\square$

-NoValue-
content/incompleteness/representability-in-q/basic-representable.tex

35.5 Basic Functions are Representable in $\mathbf{Q}$

*inc:req:bre:
sec* First we have to show that all the basic functions are representable in $\mathbf{Q}$. In the end, we need to show how to assign to each k -ary basic function $f(x_0, \dots, x_{k-1})$ a fórmula $\varphi_f(x_0, \dots, x_{k-1}, y)$ that represents it.

We will be able to represent zero, successor, plus, times, the characteristic function for equality, and projections. In each case, the appropriate representing function is entirely straightforward; for example, zero is represented by the formula $y = 0$, successor is represented by the fórmula $x'_0 = y$, and addition is represented by the fórmula $(x_0 + x_1) = y$. The work involves showing that $\mathbf{Q}$ can prove the relevant sentenças; for example, saying that addition is represented by the fórmula above involves showing that for every pair of natural numbers m and n , $\mathbf{Q}$ proves

$$\begin{aligned} \bar{n} + \bar{m} &= \overline{n + m} \text{ and} \\ \forall y ((\bar{n} + \bar{m}) = y &\rightarrow y = \overline{n + m}). \end{aligned}$$

*inc:req:bre:
prop:rep-zero* **Proposição 35.10.** *The zero function $\text{zero}(x) = 0$ is represented in $\mathbf{Q}$ by $\varphi_{\text{zero}}(x, y) \equiv y = 0$.*

*inc:req:bre:
prop:rep-succ* **Proposição 35.11.** *The successor function $\text{succ}(x) = x + 1$ is represented in $\mathbf{Q}$ by $\varphi_{\text{succ}}(x, y) \equiv y = x'$.*

Proposição 35.12. *The projection function $P_i^n(x_0, \dots, x_{n-1}) = x_i$ is represented in $\mathbf{Q}$ by*

*inc:req:bre:
prop:rep-proj*

$$\varphi_{P_i^n}(x_0, \dots, x_{n-1}, y) \equiv y = x_i.$$

Problema 35.2. Prove that $y = 0$, $y = x'$, and $y = x_i$ represent zero, succ, and P_i^n , respectively.

Proposição 35.13. *The characteristic function of $=$,*

*inc:req:bre:
prop:rep-id*

$$\chi_{=}(x_0, x_1) = \begin{cases} 1 & \text{if } x_0 = x_1 \\ 0 & \text{otherwise} \end{cases}$$

is represented in $\mathbf{Q}$ by

$$\varphi_{\chi_{=}}(x_0, x_1, y) \equiv (x_0 = x_1 \wedge y = \bar{1}) \vee (x_0 \neq x_1 \wedge y = \bar{0}).$$

The proof requires the following lemma.

Lema 35.14. *Given natural numbers n and m , if $n \neq m$, then $\mathbf{Q} \vdash \bar{n} \neq \bar{m}$.*

*inc:req:bre:
lem:q-proves-neq*

Prova. Use induction on n to show that for every m , if $n \neq m$, then $\mathbf{Q} \vdash \bar{n} \neq \bar{m}$.

In the base case, $n = 0$. If m is not equal to 0, then $m = k + 1$ for some natural number k . We have an axiom that says $\forall x \, 0 \neq x'$. By a quantifier axiom, replacing x by $\bar{k}$, we can conclude $0 \neq \bar{k}'$. But $\bar{k}'$ is just $\bar{m}$.

In the induction step, we can assume the claim is true for n , and consider $n + 1$. Let m be any natural number. There are two possibilities: either $m = 0$ or for some k we have $m = k + 1$. The first case is handled as above. In the second case, suppose $n + 1 \neq k + 1$. Then $n \neq k$. By the induction hypothesis for n we have $\mathbf{Q} \vdash \bar{n} \neq \bar{k}$. We have an axiom that says $\forall x \, \forall y \, x' = y' \rightarrow x = y$. Using a quantifier axiom, we have $\bar{n}' = \bar{k}' \rightarrow \bar{n} = \bar{k}$. Using propositional logic, we can conclude, in $\mathbf{Q}$, $\bar{n} \neq \bar{k} \rightarrow \bar{n}' \neq \bar{k}'$. Using modus ponens, we can conclude $\bar{n}' \neq \bar{k}'$, which is what we want, since $\bar{k}'$ is $\bar{m}$. $\square$

explanation

Note that the lemma does not say much: in essence it says that $\mathbf{Q}$ can prove that different numerals denote different objects. For example, $\mathbf{Q}$ proves $0'' \neq 0'''$. But showing that this holds in general requires some care. Note also that although we are using induction, it is induction *outside* of $\mathbf{Q}$.

Prova de Proposição 35.13. If $n = m$, then $\bar{n}$ and $\bar{m}$ are the same term, and $\chi_{=}(n, m) = 1$. But $\mathbf{Q} \vdash (\bar{n} = \bar{m} \wedge \bar{1} = \bar{1})$, so it proves $\varphi_{=}(n, m, \bar{1})$. If $n \neq m$, then $\chi_{=}(n, m) = 0$. By **Lema 35.14**, $\mathbf{Q} \vdash \bar{n} \neq \bar{m}$ and so also $(\bar{n} \neq \bar{m} \wedge 0 = 0)$. Thus $\mathbf{Q} \vdash \varphi_{=}(n, m, \bar{0})$.

For the second part, we also have two cases. If $n = m$, we have to show that $\mathbf{Q} \vdash \forall y \, (\varphi_{=}(n, m, y) \rightarrow y = \bar{1})$. Arguing informally, suppose $\varphi_{=}(n, m, y)$, O.s.,

$$(\bar{n} = \bar{n} \wedge y = \bar{1}) \vee (\bar{n} \neq \bar{n} \wedge y = \bar{0})$$

35.5. BASIC FUNCTIONS ARE REPRESENTABLE IN $\mathbf{Q}$

The left disjunct implies $y = \bar{1}$ by logic; the right contradicts $\bar{n} = \bar{n}$ which is provable by logic.

Suppose, on the other hand, that $n \neq m$. Then $\varphi_{=}(\bar{n}, \bar{m}, y)$ is

$$(\bar{n} = \bar{m} \wedge y = \bar{1}) \vee (\bar{n} \neq \bar{m} \wedge y = \bar{0})$$

Here, the left disjunct contradicts $\bar{n} \neq \bar{m}$, which is provable in $\mathbf{Q}$ by [Lema 35.14](#); the right disjunct entails $y = \bar{0}$. $\square$

inc:req:bre: prop:rep-add **Proposição 35.15.** *The addition function $\text{add}(x_0, x_1) = x_0 + x_1$ is represented in $\mathbf{Q}$ by*

$$\varphi_{\text{add}}(x_0, x_1, y) \equiv y = (x_0 + x_1).$$

inc:req:bre: lem:q-proves-add **Lema 35.16.** $\mathbf{Q} \vdash (\bar{n} + \bar{m}) = \overline{n + m}$

Prova. We prove this by induction on m . If $m = 0$, the claim is that $\mathbf{Q} \vdash (\bar{n} + \bar{0}) = \bar{n}$. This follows by axiom Q_4 . Now suppose the claim for m ; let's prove the claim for $m + 1$, O.s., prove that $\mathbf{Q} \vdash (\bar{n} + \bar{m} + \bar{1}) = \overline{n + m + 1}$. Note that $\bar{m} + \bar{1}$ is just $\bar{m}'$, and $\overline{n + m + 1}$ is just $\overline{n + m'}$. By axiom Q_5 , $\mathbf{Q} \vdash (\bar{n} + \bar{m}') = \overline{(n + m')}$. By induction hypothesis, $\mathbf{Q} \vdash (\bar{n} + \bar{m}) = \overline{n + m}$. So $\mathbf{Q} \vdash (\bar{n} + \bar{m}') = \overline{n + m'}$. $\square$

Prova de Proposição 35.15. The fórmula $\varphi_{\text{add}}(x_0, x_1, y)$ representing add is $y = (x_0 + x_1)$. First we show that if $\text{add}(n, m) = k$, then $\mathbf{Q} \vdash \varphi_{\text{add}}(\bar{n}, \bar{m}, \bar{k})$, O.s., $\mathbf{Q} \vdash \bar{k} = (\bar{n} + \bar{m})$. But since $k = n + m$, $\bar{k}$ just is $\overline{n + m}$, and we've shown in [Lema 35.16](#) that $\mathbf{Q} \vdash (\bar{n} + \bar{m}) = \overline{n + m}$.

We also have to show that if $\text{add}(n, m) = k$, then

$$\mathbf{Q} \vdash \forall y (\varphi_{\text{add}}(\bar{n}, \bar{m}, y) \rightarrow y = \bar{k}).$$

Suppose we have $(\bar{n} + \bar{m}) = y$. Since

$$\mathbf{Q} \vdash (\bar{n} + \bar{m}) = \overline{n + m},$$

we can replace the left side with $\overline{n + m}$ and get $\overline{n + m} = y$, for arbitrary y . $\square$

inc:req:bre: prop:rep-mult **Proposição 35.17.** *The multiplication function $\text{mult}(x_0, x_1) = x_0 \cdot x_1$ is represented in $\mathbf{Q}$ by*

$$\varphi_{\text{mult}}(x_0, x_1, y) \equiv y = (x_0 \times x_1).$$

Prova. Exercise. $\square$

inc:req:bre: lem:q-proves-mult **Lema 35.18.** $\mathbf{Q} \vdash (\bar{n} \times \bar{m}) = \overline{n \cdot m}$

Prova. Exercise. $\square$

Problema 35.3. Prove [Lema 35.18](#).

Problema 35.4. Use [Lema 35.18](#) to prove [Proposição 35.17](#).

explanation

Recall that we use $\times$ for the function symbol of the language of arithmetic, and $\cdot$ for the ordinary multiplication operation on numbers. So $\cdot$ can appear between expressions for numbers (such as in $m \cdot n$) while $\times$ appears only between terms of the language of arithmetic (such as in $(\overline{m} \times \overline{n})$). Even more confusingly, $+$ is used for both the **símbolo de função** and the addition operation. When it appears between terms—p.ex., in $(\overline{n} + \overline{m})$ —it is the 2-place **símbolo de função** of the language of arithmetic, and when it appears between numbers—p.ex., in $n + m$ —it is the addition operation. This includes the case $\overline{n} + \overline{m}$: this is the standard numeral corresponding to the number $n + m$.

-NoValue-

 content/incompleteness/representability-in-q/composition-representable.tex

35.6 Composition is Representable in $\mathbf{Q}$

Suppose h is defined by

inc:req:cmp:
sec

$$h(x_0, \dots, x_{l-1}) = f(g_0(x_0, \dots, x_{l-1}), \dots, g_{k-1}(x_0, \dots, x_{l-1})).$$

where we have already found **fórmulas** $\varphi_f, \varphi_{g_0}, \dots, \varphi_{g_{k-1}}$ representing the functions f , and $g_0, \dots, g_{k-1}$, respectively. We have to find a **fórmula** φ_h representing h .

Let's start with a simple case, where all functions are 1-place, O.s., consider $h(x) = f(g(x))$. If $\varphi_f(y, z)$ represents f , and $\varphi_g(x, y)$ represents g , we need a **fórmula** $\varphi_h(x, z)$ that represents h . Note that $h(x) = z$ iff there is a y such that both $z = f(y)$ and $y = g(x)$. (If $h(x) = z$, then $g(x)$ is such a y ; if such a y exists, then since $y = g(x)$ and $z = f(y)$, $z = f(g(x))$.) This suggests that $\exists y (\varphi_g(x, y) \wedge \varphi_f(y, z))$ is a good candidate for $\varphi_h(x, z)$. We just have to verify that $\mathbf{Q}$ proves the relevant **fórmulas**.

Proposição 35.19. *If $h(n) = m$, then $\mathbf{Q} \vdash \varphi_h(\overline{n}, \overline{m})$.*

inc:req:cmp:
prop:rep1

Prova. Suppose $h(n) = m$, O.s., $f(g(n)) = m$. Let $k = g(n)$. Then

$$\mathbf{Q} \vdash \varphi_g(\overline{n}, \overline{k})$$

since φ_g represents g , and

$$\mathbf{Q} \vdash \varphi_f(\overline{k}, \overline{m})$$

since φ_f represents f . Thus,

$$\mathbf{Q} \vdash \varphi_g(\overline{n}, \overline{k}) \wedge \varphi_f(\overline{k}, \overline{m})$$

and consequently also

$$\mathbf{Q} \vdash \exists y (\varphi_g(\overline{n}, y) \wedge \varphi_f(y, \overline{m})),$$

O.s., $\mathbf{Q} \vdash \varphi_h(\overline{n}, \overline{m})$. □

35.7. REGULAR MINIMIZATION IS REPRESENTABLE IN $\mathbf{Q}$

inc:req:cmp: prop:rep2 **Proposição 35.20.** *If $h(n) = m$, then $\mathbf{Q} \vdash \forall z (\varphi_h(\bar{n}, z) \rightarrow z = \bar{m})$.*

Prova. Suppose $h(n) = m$, O.s., $f(g(n)) = m$. Let $k = g(n)$. Then

$$\mathbf{Q} \vdash \forall y (\varphi_g(\bar{n}, y) \rightarrow y = \bar{k})$$

since φ_g represents g , and

$$\mathbf{Q} \vdash \forall z (\varphi_f(\bar{k}, z) \rightarrow z = \bar{m})$$

since φ_f represents f . Using just a little bit of logic, we can show that also

$$\mathbf{Q} \vdash \forall z (\exists y (\varphi_g(\bar{n}, y) \wedge \varphi_f(y, z)) \rightarrow z = \bar{m}).$$

O.s., $\mathbf{Q} \vdash \forall y (\varphi_h(\bar{n}, y) \rightarrow y = \bar{m})$. □

The same idea works in the more complex case where f and g_i have arity greater than 1.

inc:req:cmp: prop:rep-composition **Proposição 35.21.** *If $\varphi_f(y_0, \dots, y_{k-1}, z)$ represents $f(y_0, \dots, y_{k-1})$ in $\mathbf{Q}$, and $\varphi_{g_i}(x_0, \dots, x_{l-1}, y)$ represents $g_i(x_0, \dots, x_{l-1})$ in $\mathbf{Q}$, then*

$$\begin{aligned} \exists y_0 \dots \exists y_{k-1} (\varphi_{g_0}(x_0, \dots, x_{l-1}, y_0) \wedge \dots \wedge \\ \varphi_{g_{k-1}}(x_0, \dots, x_{l-1}, y_{k-1}) \wedge \varphi_f(y_0, \dots, y_{k-1}, z)) \end{aligned}$$

represents

$$h(x_0, \dots, x_{l-1}) = f(g_0(x_0, \dots, x_{l-1}), \dots, g_{k-1}(x_0, \dots, x_{l-1})).$$

Prova. Exercise. □

Problema 35.5. Using the proofs of [Proposição 35.20](#) and [Proposição 35.20](#) as a guide, carry out the proof of [Proposição 35.21](#) in detail.

-NoValue- _____
content/incompleteness/representability-in-q/minimization-representable.tex

35.7 Regular Minimization is Representable in $\mathbf{Q}$

inc:req:min: sec Let's consider unbounded search. Suppose $g(x, z)$ is regular and representable in $\mathbf{Q}$, say by the [fórmula](#) $\varphi_g(x, z, y)$. Let f be defined by $f(z) = \mu x [g(x, z) = 0]$. We would like to find a [fórmula](#) $\varphi_f(z, y)$ representing f . The value of $f(z)$ is that number x which (a) satisfies $g(x, z) = 0$ and (b) is the least such, O.s., for any $w < x$, $g(w, z) \neq 0$. So the following is a natural choice:

$$\varphi_f(z, y) \equiv \varphi_g(y, z, 0) \wedge \forall w (w < y \rightarrow \neg \varphi_g(w, z, 0)).$$

In the general case, of course, we would have to replace z with $z_0, \dots, z_k$.

The proof, again, will involve some lemmas about things $\mathbf{Q}$ is strong enough to prove.

CAPÍTULO 35. REPRESENTABILITY IN $\mathbf{Q}$

Lema 35.22. For every *símbolo de constante* a and every natural number n , *inc:req:min:*
lem:succ

$$\mathbf{Q} \vdash (a' + \bar{n}) = (a + \bar{n})'.$$

Prova. The proof is, as usual, by induction on n . In the base case, $n = 0$, we need to show that $\mathbf{Q}$ proves $(a' + 0) = (a + 0)'$. But we have:

$$\mathbf{Q} \vdash (a' + 0) = a' \quad \text{by axiom } Q_4 \quad (35.1) \quad \text{inc:req:min:}$$

$$\mathbf{Q} \vdash (a + 0) = a \quad \text{by axiom } Q_4 \quad (35.2) \quad \text{step1 inc:req:min:}$$

$$\mathbf{Q} \vdash (a + 0)' = a' \quad \text{by Eq. (35.2)} \quad (35.3) \quad \text{step2 inc:req:min:}$$

$$\mathbf{Q} \vdash (a' + 0) = (a + 0)' \quad \text{by Eq. (35.1) and Eq. (35.3)} \quad \text{step3}$$

In the induction step, we can assume that we have shown that $\mathbf{Q} \vdash (a' + \bar{n}) = (a + \bar{n})'$. Since $\overline{n+1}$ is $\bar{n}'$, we need to show that $\mathbf{Q}$ proves $(a' + \bar{n}') = (a + \bar{n}')'$. We have:

$$\mathbf{Q} \vdash (a' + \bar{n}') = (a' + \bar{n})' \quad \text{by axiom } Q_5 \quad (35.4) \quad \text{inc:req:min:}$$

$$\mathbf{Q} \vdash (a' + \bar{n}') = (a + \bar{n}')' \quad \text{inductive hypothesis} \quad (35.5) \quad \text{step5 inc:req:min:}$$

$$\mathbf{Q} \vdash (a' + \bar{n}') = (a + \bar{n}')' \quad \text{by Eq. (35.4) and Eq. (35.5).} \quad \square \quad \text{step6}$$

It is again worth mentioning that this is weaker than saying that $\mathbf{Q}$ proves $\forall x \forall y (x' + y) = (x + y)'$. Although this *sentença* is true in $\mathfrak{N}$, $\mathbf{Q}$ does not prove it.

Lema 35.23. $\mathbf{Q} \vdash \forall x \neg x < 0$. *inc:req:min:*
lem:less-zero

Prova. We give the proof informally (O.s., only giving hints as to how to construct the formal *derivação*).

We have to prove $\neg a < 0$ for an arbitrary a . By the definition of $<$, we need to prove $\neg \exists y (y' + a) = 0$ in $\mathbf{Q}$. We'll assume $\exists y (y' + a) = 0$ and prove a contradiction. Suppose $(b' + a) = 0$. Using Q_3 , we have that $a = 0 \vee \exists y a = y'$. We distinguish cases.

Case 1: $a = 0$ holds. From $(b' + a) = 0$, we have $(b' + 0) = 0$. By axiom Q_4 of $\mathbf{Q}$, we have $(b' + 0) = b'$, and hence $b' = 0$. But by axiom Q_2 we also have $b' \neq 0$, a contradiction.

Case 2: For some c , $a = c'$. But then we have $(b' + c') = 0$. By axiom Q_5 , we have $(b' + c)' = 0$, again contradicting axiom Q_2 . $\square$

Lema 35.24. For every natural number n , *inc:req:min:*
lem:less-nsucc

$$\mathbf{Q} \vdash \forall x (x < \overline{n+1} \rightarrow (x = 0 \vee \dots \vee x = \bar{n})).$$

Prova. We use induction on n . Let us consider the base case, when $n = 0$. In that case, we need to show $a < \bar{1} \rightarrow a = 0$, for arbitrary a . Suppose $a < \bar{1}$. Then by the defining axiom for $<$, we have $\exists y (y' + a) = 0'$ (since $\bar{1} \equiv 0'$).

Suppose b has that property, O.s., we have $(b' + a) = 0'$. We need to show $a = 0$. By axiom Q_3 , we have either $a = 0$ or that there is a c such that $a = c'$.

35.7. REGULAR MINIMIZATION IS REPRESENTABLE IN $\mathbf{Q}$

In the former case, there is nothing to show. So suppose $a = c'$. Then we have $(b' + c') = o'$. By axiom Q_5 of $\mathbf{Q}$, we have $(b' + c)' = o'$. By axiom Q_1 , we have $(b' + c) = o$. But this means, by axiom Q_8 , that $c < o$, contradicting **Lema 35.23**.

Now for the inductive step. We prove the case for $n + 1$, assuming the case for n . So suppose $a < \overline{n + 2}$. Again using Q_3 we can distinguish two cases: $a = o$ and for some b , $a = b'$. In the first case, $a = o \vee \dots \vee a = \overline{n + 1}$ follows trivially. In the second case, we have $b' < \overline{n + 2}$, O.s., $b' < \overline{n + 1}'$. By axiom Q_8 , for some c , $(c' + b') = \overline{n + 1}'$. By axiom Q_5 , $(c' + b)' = \overline{n + 1}'$. By axiom Q_1 , $(c' + b) = \overline{n + 1}$, and so $b < \overline{n + 1}$ by axiom Q_8 . By inductive hypothesis, $b = o \vee \dots \vee b = \overline{n}$. From this, we get $b' = o' \vee \dots \vee b' = \overline{n}'$ by logic, and so $a = \overline{1} \vee \dots \vee a = \overline{n + 1}$ since $a = b'$. $\square$

inc:req:min: **Lema 35.25.** *For every natural number m ,*
lem:trichotomy

$$\mathbf{Q} \vdash \forall y ((y < \overline{m} \vee \overline{m} < y) \vee y = \overline{m}).$$

Prova. By induction on m . First, consider the case $m = 0$. $\mathbf{Q} \vdash \forall y (y = o \vee \exists z y = z')$ by Q_3 . Let a be arbitrary. Then either $a = o$ or for some b , $a = b'$. In the former case, we also have $(a < o \vee o < a) \vee a = o$. But if $a = b'$, then $(b' + o) = (a + o)$ by the logic of $=$. By Q_4 , $(a + o) = a$, so we have $(b' + o) = a$, and hence $\exists z (z' + o) = a$. By the definition of $<$ in Q_8 , $o < a$. If $o < a$, then also $(o < a \vee a < o) \vee a = o$.

Now suppose we have

$$\mathbf{Q} \vdash \forall y ((y < \overline{m} \vee \overline{m} < y) \vee y = \overline{m})$$

and we want to show

$$\mathbf{Q} \vdash \forall y ((y < \overline{m + 1} \vee \overline{m + 1} < y) \vee y = \overline{m + 1})$$

Let a be arbitrary. By Q_3 , either $a = o$ or for some b , $a = b'$. In the first case, we have $\overline{m}' + a = \overline{m + 1}$ by Q_4 , and so $a < \overline{m + 1}$ by Q_8 .

Now consider the second case, $a = b'$. By the induction hypothesis, $(b < \overline{m} \vee \overline{m} < b) \vee b = \overline{m}$.

The first disjunct $b < \overline{m}$ is equivalent (by Q_8) to $\exists z (z' + b) = \overline{m}$. Suppose c has this property. If $(c' + b) = \overline{m}$, then also $(c' + b)' = \overline{m}'$. By Q_5 , $(c' + b)' = (c' + b')$. Hence, $(c' + b') = \overline{m}'$. We get $\exists u (u' + b') = \overline{m + 1}$ by existentially generalizing on c' and keeping in mind that $\overline{m}' \equiv \overline{m + 1}$. Hence, if $b < \overline{m}$ then $b' < \overline{m + 1}$ and so $a < \overline{m + 1}$.

Now suppose $\overline{m} < b$, O.s., $\exists z (z' + \overline{m}) = b$. Suppose c is such a z , O.s., $(c' + \overline{m}) = b$. By logic, $(c' + \overline{m})' = b'$. By Q_5 , $(c' + \overline{m}') = b'$. Since $a = b'$ and $\overline{m}' \equiv \overline{m + 1}$, $(c' + \overline{m + 1}) = a$. By Q_8 , $\overline{m + 1} < a$.

Finally, assume $b = \overline{m}$. Then, by logic, $b' = \overline{m}'$, and so $a = \overline{m + 1}$.

Hence, from each disjunct of the case for m and b , we can obtain the corresponding disjunct for $m + 1$ and a . $\square$

Proposição 35.26. *If $\varphi_g(x, z, y)$ represents $g(x, z)$ in $\mathbf{Q}$, then*

*inc:req:min:
prop:rep-minimization*

$$\varphi_f(z, y) \equiv \varphi_g(y, z, 0) \wedge \forall w (w < y \rightarrow \neg \varphi_g(w, z, 0))$$

represents $f(z) = \mu x [g(x, z) = 0]$.

Prova. First we show that if $f(n) = m$, then $\mathbf{Q} \vdash \varphi_f(\bar{n}, \bar{m})$, O.s.,

$$\mathbf{Q} \vdash \varphi_g(\bar{m}, \bar{n}, 0) \wedge \forall w (w < \bar{m} \rightarrow \neg \varphi_g(w, \bar{n}, 0)).$$

Since $\varphi_g(x, z, y)$ represents $g(x, z)$ and $g(m, n) = 0$ if $f(n) = m$, we have

$$\mathbf{Q} \vdash \varphi_g(\bar{m}, \bar{n}, 0).$$

If $f(n) = m$, then for every $k < m$, $g(k, n) \neq 0$. So

$$\mathbf{Q} \vdash \neg \varphi_g(\bar{k}, \bar{n}, 0).$$

We get that

$$\mathbf{Q} \vdash \forall w (w < \bar{m} \rightarrow \neg \varphi_g(w, \bar{n}, 0)). \quad (35.6) \quad \text{inc:req:min: rep-less}$$

by **Lema 35.23** in case $m = 0$ and by **Lema 35.24** otherwise.

Now let's show that if $f(n) = m$, then $\mathbf{Q} \vdash \forall y (\varphi_f(\bar{n}, y) \rightarrow y = \bar{m})$. We again sketch the argument informally, leaving the formalization to the reader.

Suppose $\varphi_f(\bar{n}, b)$. From this we get (a) $\varphi_g(b, \bar{n}, 0)$ and (b) $\forall w (w < b \rightarrow \neg \varphi_g(w, \bar{n}, 0))$. By **Lema 35.25**, $(b < \bar{m} \vee \bar{m} < b) \vee b = \bar{m}$. We'll show that both $b < \bar{m}$ and $\bar{m} < b$ leads to a contradiction.

If $\bar{m} < b$, then $\neg \varphi_g(\bar{m}, \bar{n}, 0)$ from (b). But $m = f(n)$, so $g(m, n) = 0$, and so $\mathbf{Q} \vdash \varphi_g(\bar{m}, \bar{n}, 0)$ since φ_g represents g . So we have a contradiction.

Now suppose $b < \bar{m}$. Then since $\mathbf{Q} \vdash \forall w (w < \bar{m} \rightarrow \neg \varphi_g(w, \bar{n}, 0))$ by **Eq. (35.6)**, we get $\neg \varphi_g(b, \bar{n}, 0)$. This again contradicts (a). $\square$

-NoValue-
content/incompleteness/representability-in-q/comp-representable.tex

35.8 Computable Functions are Representable in $\mathbf{Q}$

Teorema 35.27. *Every computable function is representable in $\mathbf{Q}$.*

*inc:req:crq:
sec*

Prova. For definiteness, and using the Church–Turing Thesis, let's say that a function is computable iff it is general recursive. The general recursive functions are those which can be defined from the zero function zero, the successor function succ, and the projection function P_i^n using composition, primitive recursion, and regular minimization. By **Lema 35.9**, any function h that can be defined from f and g can also be defined using composition and regular

35.9. REPRESENTING RELATIONS

minimization from f , g , and zero, succ, P_i^n , add, mult, $\chi_=_$. Consequently, a function is general recursive iff it can be defined from zero, succ, P_i^n , add, mult, $\chi_=_$ using composition and regular minimization.

We've furthermore shown that the basic functions in question are representable in $\mathbf{Q}$ (Proposições 35.10 a 35.13, 35.15 e 35.17), and that any function defined from representable functions by composition or regular minimization (Proposição 35.21, Proposição 35.26) is also representable. Thus every general recursive function is representable in $\mathbf{Q}$. $\square$

We have shown that the set of computable functions can be characterized as the set of functions representable in $\mathbf{Q}$. In fact, the proof is more general. From the definition of representability, it is not hard to see that any theory extending $\mathbf{Q}$ (or in which one can interpret $\mathbf{Q}$) can represent the computable functions. But, conversely, in any *derivação* system in which the notion of *derivação* is computable, every representable function is computable. So, for example, the set of computable functions can be characterized as the set of functions representable in Peano arithmetic, or even Zermelo–Fraenkel set theory. As Gödel noted, this is somewhat surprising. We will see that when it comes to provability, questions are very sensitive to which theory you consider; roughly, the stronger the axioms, the more you can prove. But across a wide range of axiomatic theories, the representable functions are exactly the computable ones; stronger theories do not represent more functions as long as they are axiomatizable.

explanation

-NoValue-

content/incompleteness/representability-in-q/representing-relations.tex

35.9 Representing Relations

inc:req:rel:sec Let us say what it means for a *relation* to be representable.

inc:req:rel: defn:representing-relations **Definição 35.28.** A relation $R(x_0, \dots, x_k)$ on the natural numbers is *representable in $\mathbf{Q}$* if there is a formula $\varphi_R(x_0, \dots, x_k)$ such that whenever $R(n_0, \dots, n_k)$ is true, $\mathbf{Q}$ proves $\varphi_R(\overline{n_0}, \dots, \overline{n_k})$, and whenever $R(n_0, \dots, n_k)$ is false, $\mathbf{Q}$ proves $\neg\varphi_R(\overline{n_0}, \dots, \overline{n_k})$.

inc:req:rel: thm:representing-rels **Teorema 35.29.** A relation is representable in $\mathbf{Q}$ if and only if it is computable.

Prova. For the forwards direction, suppose $R(x_0, \dots, x_k)$ is represented by the formula $\varphi_R(x_0, \dots, x_k)$. Here is an algorithm for computing R : on input $n_0, \dots, n_k$, simultaneously search for a proof of $\varphi_R(\overline{n_0}, \dots, \overline{n_k})$ and a proof of $\neg\varphi_R(\overline{n_0}, \dots, \overline{n_k})$. By our hypothesis, the search is bound to find one or the other; if it is the first, report “yes,” and otherwise, report “no.”

In the other direction, suppose $R(x_0, \dots, x_k)$ is computable. By definition, this means that the function $\chi_R(x_0, \dots, x_k)$ is computable. By Teorema 35.2,

χ_R is represented by a formula, say $\varphi_{\chi_R}(x_0, \dots, x_k, y)$. Let $\varphi_R(x_0, \dots, x_k)$ be the formula $\varphi_{\chi_R}(x_0, \dots, x_k, \bar{1})$. Then for any $n_0, \dots, n_k$, if $R(n_0, \dots, n_k)$ is true, then $\chi_R(n_0, \dots, n_k) = 1$, in which case $\mathbf{Q}$ proves $\varphi_{\chi_R}(\bar{n}_0, \dots, \bar{n}_k, \bar{1})$, and so $\mathbf{Q}$ proves $\varphi_R(\bar{n}_0, \dots, \bar{n}_k)$. On the other hand, if $R(n_0, \dots, n_k)$ is false, then $\chi_R(n_0, \dots, n_k) = 0$. This means that $\mathbf{Q}$ proves

$$\forall y (\varphi_{\chi_R}(\bar{n}_0, \dots, \bar{n}_k, y) \rightarrow y = \bar{0}).$$

Since $\mathbf{Q}$ proves $\bar{0} \neq \bar{1}$, $\mathbf{Q}$ proves $\neg \varphi_{\chi_R}(\bar{n}_0, \dots, \bar{n}_k, \bar{1})$, and so it proves $\neg \varphi_R(\bar{n}_0, \dots, \bar{n}_k)$. $\square$

Problema 35.6. Show that if R is representable in $\mathbf{Q}$, so is χ_R .

-NoValue-

[content/incompleteness/representability-in-q/undecidability.tex](#)

35.10 Undecidability

We call a theory $\mathbf{T}$ *undecidable* if there is no computational procedure which, after finitely many steps and unfailingly, provides a correct answer to the question “does $\mathbf{T}$ prove φ ?” for any sentence φ in the language of $\mathbf{T}$. So $\mathbf{Q}$ would be decidable iff there were a computational procedure which decides, given a sentence φ in the language of arithmetic, whether $\mathbf{Q} \vdash \varphi$ or not. We can make this more precise by asking: Is the relation $\text{Prov}_{\mathbf{Q}}(y)$, which holds of y iff y is the Gödel number of a sentence provable in $\mathbf{Q}$, recursive? The answer is: no.

inc:req:und:
sec

Teorema 35.30. $\mathbf{Q}$ is undecidable, O.s., the relation

$$\text{Prov}_{\mathbf{Q}}(y) \Leftrightarrow \text{Sent}(y) \wedge \exists x \text{Prf}_{\mathbf{Q}}(x, y)$$

is not recursive.

Prova. Suppose it were. Then we could solve the halting problem as follows: Given e and n , we know that $\varphi_e(n) \downarrow$ iff there is an s such that $T(e, n, s)$, where T is Kleene’s predicate from [Teorema 29.28](#). Since T is primitive recursive it is representable in $\mathbf{Q}$ by a formula ψ_T , that is, $\mathbf{Q} \vdash \psi_T(\bar{e}, \bar{n}, \bar{s})$ iff $T(e, n, s)$. If $\mathbf{Q} \vdash \psi_T(\bar{e}, \bar{n}, \bar{s})$ then also $\mathbf{Q} \vdash \exists y \psi_T(\bar{e}, \bar{n}, y)$. If no such s exists, then $\mathbf{Q} \vdash \neg \psi_T(\bar{e}, \bar{n}, \bar{s})$ for every s . But $\mathbf{Q}$ is ω -consistent, O.s., if $\mathbf{Q} \vdash \neg \varphi(\bar{n})$ for every $n \in \mathbb{N}$, then $\mathbf{Q} \not\vdash \exists y \varphi(y)$. We know this because the axioms of $\mathbf{Q}$ are true in the standard model $\mathfrak{N}$. So, $\mathbf{Q} \not\vdash \exists y \psi_T(\bar{e}, \bar{n}, y)$. In other words, $\mathbf{Q} \vdash \exists y \psi_T(\bar{e}, \bar{n}, y)$ iff there is an s such that $T(e, n, s)$, O.s., iff $\varphi_e(n) \downarrow$. From e and n we can compute $\# \exists y \psi_T(\bar{e}, \bar{n}, y)^\#$, let $g(e, n)$ be the primitive recursive function which does that. So

$$h(e, n) = \begin{cases} 1 & \text{if } \text{Prov}_{\mathbf{Q}}(g(e, n)) \\ 0 & \text{otherwise.} \end{cases}$$

This would show that h is recursive if $\text{Prov}_{\mathbf{Q}}$ is. But h is not recursive, by [Teorema 29.29](#), so $\text{Prov}_{\mathbf{Q}}$ cannot be either. $\square$

35.11. Σ_1 COMPLETENESS

Corolário 35.31. *First-order logic is undecidable.*

Prova. If first-order logic were decidable, provability in $\mathbf{Q}$ would be as well, since $\mathbf{Q} \vdash \varphi$ iff $\vdash \tau \rightarrow \varphi$, where τ is the conjunction of the axioms of $\mathbf{Q}$. $\square$

-NoValue- content/incompleteness/representability-in-q/sigma1-completeness.tex

35.11 Σ_1 completeness

inc:inp:slc: sec Despite the incompleteness of $\mathbf{Q}$ and its consistent, axiomatizable extensions, we have seen that $\mathbf{Q}$ does prove many basic facts about numerals. In fact, this can be extended quite considerably. To understand the scope of what can be proved in $\mathbf{Q}$, we introduce the notions of Δ_0 , Σ_1 , and Π_1 *fórmulas*. Roughly speaking, a Σ_1 *fórmula* is one of the form $\exists x \psi(x)$, where ψ is constructed using only propositional connectives and bounded quantifiers. We shall show that if φ is a Σ_1 *sentença* which is true in $\mathfrak{N}$, then $\mathbf{Q} \vdash \varphi$ (**Teorema 35.38**).

inc:inp:slc: defn:bd-quant **Definição 35.32.** A *bounded existential fórmula* is one of the form $\exists x (x < t \wedge \varphi(x))$ where t is any term, which we conventionally write as $(\exists x < t) \varphi(x)$. A *bounded universal fórmula* is one of the form $\forall x (x < t \rightarrow \varphi(x))$ where t is any term, which we conventionally write as $(\forall x < t) \varphi(x)$.

inc:inp:slc: defn:delta0-sigma1-pi1-frn **Definição 35.33.** A *fórmula* ψ is Δ_0 if it is built up from atomic *fórmulas* using only propositional connectives and bounded quantification. A *fórmula* φ is Σ_1 if $\varphi \equiv \exists x \psi(x)$ where ψ is Δ_0 . A *fórmula* φ is Π_1 if $\varphi \equiv \forall x \psi(x)$ where ψ is Δ_0 .

inc:inp:slc: lem:q-proves-clterm-id **Lema 35.34.** *Suppose t is a closed term such that $\text{Val}^{\mathfrak{N}}(t) = n$. Then $\mathbf{Q} \vdash t = \bar{n}$.*

Prova. We prove this by induction on the complexity of t . For the base case, $\text{Val}^{\mathfrak{N}}(0) = 0$, and $\mathbf{Q} \vdash 0 = \bar{0}$ since $\bar{0} \equiv 0$. For the inductive case, let t_1 and t_2 be terms such that $\text{Val}^{\mathfrak{N}}(t_1) = n_1$, $\text{Val}^{\mathfrak{N}}(t_2) = n_2$, $\mathbf{Q} \vdash t_1 = \bar{n}_1$, and $\mathbf{Q} \vdash t_2 = \bar{n}_2$.

Then $\text{Val}^{\mathfrak{N}}((t'_1)) = n_1 + 1$, and we have that $\mathbf{Q} \vdash t'_1 = \bar{n}_1'$ by the first-order rules for identity applied to the induction hypothesis and the *fórmula* $\bar{n}_1' = \overline{n_1'}$, so we have $\mathbf{Q} \vdash t'_1 = \overline{n_1 + 1}$ by the definition of numerals.

For sums we have

$$\text{Val}^{\mathfrak{N}}((t_1 + t_2)) = \text{Val}^{\mathfrak{N}}(t_1) + \text{Val}^{\mathfrak{N}}(t_2) = n_1 + n_2.$$

By the induction hypothesis and the rules for identity, $\mathbf{Q} \vdash t_1 + t_2 = \bar{n}_1 + t_2$, and then $\mathbf{Q} \vdash t_1 + t_2 = \bar{n}_1 + \bar{n}_2$ by a second application of the rules for identity. By **Lema 35.16**, $\mathbf{Q} \vdash \bar{n}_1 + \bar{n}_2 = \overline{n_1 + n_2}$, so $\mathbf{Q} \vdash t_1 + t_2 = \overline{n_1 + n_2}$.

Similar reasoning also works for $\times$, using **Lema 35.18**. Since this exhausts the closed terms of arithmetic, we have that $\mathbf{Q} \vdash t = \bar{n}$ for all closed terms t such that $\text{Val}^{\mathfrak{N}}(t) = n$. $\square$

Problema 35.7. Prove in detail the part of [Lema 35.34](#) involving $\times$.

Lema 35.35. Suppose t_1 and t_2 are closed terms. Then

*inc:inp:slc:
lem:atomic-completeness*

1. If $\text{Val}^{\mathfrak{N}}(t_1) = \text{Val}^{\mathfrak{N}}(t_2)$, then $\mathbf{Q} \vdash t_1 = t_2$.
2. If $\text{Val}^{\mathfrak{N}}(t_1) \neq \text{Val}^{\mathfrak{N}}(t_2)$, then $\mathbf{Q} \vdash t_1 \neq t_2$.
3. If $\text{Val}^{\mathfrak{N}}(t_1) < \text{Val}^{\mathfrak{N}}(t_2)$, then $\mathbf{Q} \vdash t_1 < t_2$.
4. If $\text{Val}^{\mathfrak{N}}(t_2) \leq \text{Val}^{\mathfrak{N}}(t_1)$, then $\mathbf{Q} \vdash \neg(t_1 < t_2)$.

Prova. Given terms t_1 and t_2 , we fix $n = \text{Val}^{\mathfrak{N}}(t_1)$ and $m = \text{Val}^{\mathfrak{N}}(t_2)$.

Suppose $\varphi \equiv t_1 = t_2$. By [Lema 35.34](#), $\mathbf{Q} \vdash t_1 = \bar{n}$ and $\mathbf{Q} \vdash t_2 = \bar{m}$. If $n = m$, then $\mathbf{Q} \vdash \bar{n} = \bar{m}$ and hence $\mathbf{Q} \vdash t_1 = t_2$ by the transitivity of identity. If $n \neq m$ then $\mathbf{Q} \vdash \bar{n} \neq \bar{m}$, and by the transitivity of identity again, $\mathbf{Q} \vdash t_1 \neq t_2$.

Now let $\varphi \equiv t_1 < t_2$. For both cases, we rely on axiom Q_8 , which states that $x < y \leftrightarrow \exists z z' + x = y$ for all x, y .

Suppose $\mathfrak{N} \models t_1 < t_2$. Then there exists some $k \in \mathbb{N}$ such that $n + k + 1 = m$. By [Lema 35.34](#), $\mathbf{Q} \vdash t_1 = \bar{n}$ and $\mathbf{Q} \vdash t_2 = \bar{m}$, and by the first part of this lemma, $\mathbf{Q} \vdash \bar{n} + \bar{k}' = \bar{m}$. By the transitivity of identity it follows that $\mathbf{Q} \vdash \bar{k}' + t_1 = t_2$, so $\mathbf{Q} \vdash \exists z z' + t_1 = t_2$. By the right-to-left direction of Q_8 , $\mathbf{Q} \vdash t_1 < t_2$.

Suppose instead that $\mathfrak{N} \not\models t_1 < t_2$, O.s., $m \leq n$. We work in $\mathbf{Q}$ and assume that $t_1 < t_2$. By the left-to-right direction of Q_8 , there is some z such that $z' + t_1 = t_2$. Since $\mathbf{Q} \vdash t_1 = \bar{n}$ and $\mathbf{Q} \vdash t_2 = \bar{m}$, $z' + \bar{n} = \bar{m}$. By an external induction on m using Q_5 , $z' + \bar{n} - \bar{m} = 0$. If $m = n$ then $z' \neq 0$, giving a contradiction via Q_3 . If $m < n$ then $(z' + \bar{n} - \bar{m} - 1)' = 0$ by Q_5 again, giving a contradiction via Q_3 . So $\mathbf{Q} \vdash \neg(t_1 < t_2)$. $\square$

Lema 35.36. Suppose φ is a *fórmula*, t a closed term, and $k = \text{Val}^{\mathfrak{N}}(t)$. Then

*inc:inp:slc:
lem:bounded-quant-equiv*

1. $\mathbf{Q} \vdash (\forall x < t) \varphi(x)$ iff $\mathbf{Q} \vdash \varphi(\bar{0}) \wedge \cdots \wedge \varphi(\overline{k-1})$.
2. $\mathbf{Q} \vdash (\exists x < t) \varphi(x)$ iff $\mathbf{Q} \vdash \varphi(\bar{0}) \vee \cdots \vee \varphi(\overline{k-1})$.

Prova. We prove the case for the bounded universal quantifier. If $\text{Val}^{\mathfrak{N}}(t) = 0$ then the left-hand side of the equivalence is provable in $\mathbf{Q}$, because there is no $x < \bar{0}$ by [Lema 35.23](#). Similarly, we can take an empty disjunction to be simply $\top$, which is also provable in $\mathbf{Q}$. We therefore suppose that $\text{Val}^{\mathfrak{N}}(t) = k + 1$ for some natural number k . By [Lema 35.34](#) we can assume that we are working with a *fórmula* of the form $(\forall x < \overline{k+1}) \varphi(x)$.

Suppose that $\mathbf{Q} \vdash (\forall x < \overline{k+1}) \varphi(x)$, and let $n \leq k$. Since $\mathbf{Q} \vdash \bar{n} < \overline{k+1}$ by [Lema 35.35](#), it follows by logic that $\mathbf{Q} \vdash \varphi(\bar{n})$. Applying this fact $k + 1$ times for each $n \leq k$, we get that $\mathbf{Q} \vdash \varphi(\bar{0}) \wedge \cdots \wedge \varphi(\bar{k})$ as desired.

For the other direction, suppose that $\mathbf{Q} \vdash \varphi(\bar{0}) \wedge \cdots \wedge \varphi(\bar{k})$. Working in $\mathbf{Q}$, suppose that $x < \overline{k+1}$. By [Lema 35.24](#) we have that $x = \bar{0} \vee \cdots \vee x = \bar{k}$, so

35.11. Σ_1 COMPLETENESS

by logic it follows that $\varphi(x)$, and hence the universal claim $(\forall x < \overline{k+1}) \varphi(x)$ follows.

The proof of the equivalence for bounded existentially quantified fórmulas is similar. $\square$

Problema 35.8. Give a detailed proof of the existential case in Lema 35.36.

*inc:inp:slc:
lem:delta0-completeness*

Lema 35.37. If φ is a Δ_0 sentença which is true in $\mathfrak{N}$, then $\mathbf{Q} \vdash \varphi$.

Prova. We prove this by induction on fórmula complexity. The base case is given by Lema 35.35, so we move to the induction step. For simplicity we split the case of negation into subcases depending on the structure of the fórmula to which the negation is applied.

1. Suppose $(\varphi \wedge \psi)$ is true in $\mathfrak{N}$, so φ and ψ are true in $\mathfrak{N}$. By the induction hypothesis, $\mathbf{Q} \vdash \varphi$ and $\mathbf{Q} \vdash \psi$, so $\mathbf{Q} \vdash (\varphi \wedge \psi)$ by logic.
2. Suppose $\neg(\varphi \wedge \psi)$ is true in $\mathfrak{N}$, so either $\neg\varphi$ or $\neg\psi$ is true in $\mathfrak{N}$. Without loss of generality, suppose the former. By the induction hypothesis $\mathbf{Q} \vdash \neg\varphi$, and hence $\mathbf{Q} \vdash \neg(\varphi \wedge \psi)$ by logic.
3. Suppose $(\varphi \vee \psi)$ is true in $\mathfrak{N}$, so either φ is true in $\mathfrak{N}$ or ψ is true in $\mathfrak{N}$. Without loss of generality, suppose the former holds. By the induction hypothesis $\mathbf{Q} \vdash \varphi$, and hence $\mathbf{Q} \vdash (\varphi \vee \psi)$ by logic.
4. Suppose $\neg(\varphi \vee \psi)$ is true in $\mathfrak{N}$, so $\neg\varphi$ and $\neg\psi$ are true in $\mathfrak{N}$. Then $\mathbf{Q} \vdash \neg\varphi$ and $\mathbf{Q} \vdash \neg\psi$ by the induction hypothesis. Consequently, $\mathbf{Q} \vdash \neg(\varphi \vee \psi)$ by logic.
5. Suppose that $(\forall x < t) \varphi(x)$ is true in $\mathfrak{N}$, where t is a closed term and $k = \text{Val}^{\mathfrak{N}}(t)$. By the induction hypothesis and logic, if $\varphi(\overline{n})$ is true in $\mathfrak{N}$ for all $n < \text{Val}^{\mathfrak{N}}(t)$ then $\mathbf{Q} \vdash \varphi(\overline{0}) \wedge \cdots \wedge \varphi(\overline{k-1})$. By Lema 35.36 it follows that $\mathbf{Q} \vdash (\forall x < t) \varphi(x)$.
6. The case for the bounded existential quantifier, where we have a sentença of the form $(\exists x < t) \varphi(x)$, is similar to that for the bounded universal quantifier.
7. Suppose that $\neg(\forall x < t) \varphi(x)$ is true in $\mathfrak{N}$, where t is a closed term. This sentença is equivalent to the sentença $(\exists x < t) \neg\varphi(x)$, with the equivalence derivable in $\mathbf{Q}$, so we may apply the reasoning for bounded existential quantifiers.
8. Similarly, suppose that $\neg(\exists x < t) \varphi(x)$ is true in $\mathfrak{N}$, where t is a closed term. This sentença is equivalent in $\mathbf{Q}$ to $(\forall x < t) \neg\varphi(x)$, and so we may apply the reasoning for bounded universal quantifiers.

9. Finally, suppose $\neg\varphi$ is true in $\mathfrak{N}$. The only cases remaining are when φ is atomic and when $\neg\varphi \equiv \neg\neg\psi$ for some Δ_0 *sentença* ψ . If φ is atomic then by **Lema 35.35**, $\mathbf{Q} \vdash \neg\varphi$. If $\neg\varphi \equiv \neg\neg\psi$, then by logic it is provably equivalent in $\mathbf{Q}$ to ψ , which is true in $\mathfrak{N}$ since $\neg\varphi$ is true in $\mathfrak{N}$. By the induction hypothesis we therefore have that $\mathbf{Q} \vdash \neg\varphi$. $\square$

Problema 35.9. Give a detailed proof of the existential case in **Lema 35.37**.

Teorema 35.38. *If φ is a Σ_1 *sentença* which is true in $\mathfrak{N}$, then $\mathbf{Q} \vdash \varphi$.*

*inc:inp:slc:
thm:sigma1-completeness*

Prova. If $\exists x\varphi(x)$ is a Σ_1 *sentença* which is true in $\mathfrak{N}$, then there exists a natural number n and a variable assignment s such that $s(x) = n$ and $\mathfrak{N}, s \models \varphi(x)$. By standard facts about the satisfaction relation it follows that $\mathfrak{N} \models \varphi(\bar{n})$. But $\varphi(\bar{n})$ is a Δ_0 *fórmula*, so by **Lema 35.37** we have that $\mathbf{Q} \vdash \varphi(\bar{n})$, and hence by logic we also have that $\mathbf{Q} \vdash \exists x\varphi(x)$. $\square$

This chapter depends on material in the chapter on computability theory, but can be left out if that hasn't been covered. It's currently a basic conversion of Jeremy Avigad's notes, has not been revised, and is missing exercises.

Capítulo 36

Theories and Computability

-NoValue-

`content/incompleteness/theories-computability/introduction.tex`

36.1 Introduction

*inc:tcp:int:
sec*

This section should be rewritten.

36.2. Q IS C.E.-COMPLETE

We have the following:

1. A definition of what it means for a function to be representable in **Q** (Definição 35.1)
2. a definition of what it means for a relation to be representable in **Q** (Definição 35.28)
3. a theorem asserting that the representable functions of **Q** are exactly the computable ones (Teorema 35.2)
4. a theorem asserting that the representable relations of **Q** are exactly the computable ones (Teorema 35.29)

A *theory* is a set of sentences that is deductively closed, that is, with the property that whenever T proves φ then φ is in T . It is probably best to think of a theory as being a collection of sentences, together with all the things that these sentences imply. From now on, we will use **Q** to refer to the *theory* consisting of the set of sentences derivable from the eight axioms in Seção 35.1. Remember that we can code formula of **Q** as numbers; if φ is such a formula, let $\# \varphi \#$ denote the number coding φ . Modulo this coding, we can now ask whether various sets of formulas are computable or not.

-NoValue-

content/incompleteness/theories-computability/q-is-ce.tex

36.2 Q is C.e.-Complete

inc:tcp:qce:
sec

Teorema 36.1. *Q is c.e. but not decidable. In fact, it is a complete c.e. set.*

Prova. It is not hard to see that **Q** is c.e., since it is the set of (codes for) sentences y such that there is a proof x of y in **Q**:

$$Q = \{y : \exists x \text{Prf}_Q(x, y)\}.$$

But we know that $\text{Prf}_Q(x, y)$ is computable (in fact, primitive recursive), and any set that can be written in the above form is c.e.

Saying that it is a complete c.e. set is equivalent to saying that $K \leq_m Q$, where $K = \{x : \varphi_x(x) \downarrow\}$. So let us show that K is reducible to **Q**. Since Kleene's predicate $T(e, x, s)$ is primitive recursive, it is representable in **Q**, say, by φ_T . Then for every x , we have

$$\begin{aligned} x \in K &\rightarrow \exists s T(x, x, s) \\ &\rightarrow \exists s (\mathbf{Q} \vdash \varphi_T(\bar{x}, \bar{x}, \bar{s})) \\ &\rightarrow \mathbf{Q} \vdash \exists s \varphi_T(\bar{x}, \bar{x}, s). \end{aligned}$$

Conversely, if $\mathbf{Q} \vdash \exists s \varphi_T(\bar{x}, \bar{x}, s)$, then, in fact, for some natural number n the formula $\varphi_T(\bar{x}, \bar{x}, \bar{n})$ must be true. Now, if $T(x, x, n)$ were false, $\mathbf{Q}$ would prove $\neg \varphi_T(\bar{x}, \bar{x}, \bar{n})$, since φ_T represents T . But then $\mathbf{Q}$ proves a false formula, which is a contradiction. So $T(x, x, n)$ must be true, which implies $\varphi_x(x) \downarrow$.

In short, we have that for every x , x is in K if and only if $\mathbf{Q}$ proves $\exists s T(\bar{x}, \bar{x}, s)$. So the function f which takes x to (a code for) the sentence $\exists s T(\bar{x}, \bar{x}, s)$ is a reduction of K to $\mathbf{Q}$. $\square$

-NoValue-

content/incompleteness/theories-computability/oconsis-ext-of-q-undec.tex

36.3 ω -Consistent Extensions of $\mathbf{Q}$ are Undecidable

explanation The proof that $\mathbf{Q}$ is c.e.-complete relied on the fact that any sentence provable in $\mathbf{Q}$ is “true” of the natural numbers. The next definition and theorem strengthen this theorem, by pinpointing just those aspects of “truth” that were needed in the proof above. Don’t dwell on this theorem too long, though, because we will soon strengthen it even further. We include it mainly for historical purposes: Gödel’s original paper used the notion of ω -consistency, but his result was strengthened by replacing ω -consistency with ordinary consistency soon after.

inc:tcp:qcn:
sec

Definição 36.2. A theory $\mathbf{T}$ is ω -consistent if the following holds: if $\exists x \varphi(x)$ is any sentence and $\mathbf{T}$ proves $\neg \varphi(\bar{0})$, $\neg \varphi(\bar{1})$, $\neg \varphi(\bar{2})$, $\dots$ then $\mathbf{T}$ does not prove $\exists x \varphi(x)$.

inc:tcp:qcn:
thm:oconsis-q

Teorema 36.3. Let $\mathbf{T}$ be any ω -consistent theory that includes $\mathbf{Q}$. Then $\mathbf{T}$ is not decidable.

Prova. If $\mathbf{T}$ includes $\mathbf{Q}$, then $\mathbf{T}$ represents the computable functions and relations. We need only modify the previous proof. As above, if $x \in K$, then $\mathbf{T}$ proves $\exists s \varphi_T(\bar{x}, \bar{x}, s)$. Conversely, suppose $\mathbf{T}$ proves $\exists s \varphi_T(\bar{x}, \bar{x}, s)$. Then x must be in K : otherwise, there is no halting computation of machine x on input x ; since φ_T represents Kleene’s T relation, $\mathbf{T}$ proves $\neg \varphi_T(\bar{x}, \bar{x}, \bar{0})$, $\neg \varphi_T(\bar{x}, \bar{x}, \bar{1})$, $\dots$, making $\mathbf{T}$ ω -inconsistent. $\square$

-NoValue-

content/incompleteness/theories-computability/extensions-of-q-not-decidable.tex

36.4 Consistent Extensions of $\mathbf{Q}$ are Undecidable

explanation Remember that a theory is *consistent* if it does not prove both φ and $\neg \varphi$ for any formula φ . Since anything follows from a contradiction, an inconsistent theory is trivial: every sentence is provable. Clearly, if a theory is ω -consistent, then it

inc:tcp:qcn:
sec

36.4. CONSISTENT EXTENSIONS OF $\mathbf{Q}$ ARE UNDECIDABLE

is consistent. But being consistent is a weaker requirement (O.s., there are theories that are consistent but not ω -consistent.). We can weaken the assumption in [Definição 36.2](#) to simple consistency to obtain a stronger theorem.

Lema 36.4. *There is no “universal computable relation.” That is, there is no binary computable relation $R(x, y)$, with the following property: whenever $S(y)$ is a unary computable relation, there is some k such that for every y , $S(y)$ is true if and only if $R(k, y)$ is true.*

Prova. Suppose $R(x, y)$ is a universal computable relation. Let $S(y)$ be the relation $\neg R(y, y)$. Since $S(y)$ is computable, for some k , $S(y)$ is equivalent to $R(k, y)$. But then we have that $S(k)$ is equivalent to both $R(k, k)$ and $\neg R(k, k)$, which is a contradiction. $\square$

Teorema 36.5. *Let $\mathbf{T}$ be any consistent theory that includes $\mathbf{Q}$. Then $\mathbf{T}$ is not decidable.*

Prova. Suppose $\mathbf{T}$ is a consistent, decidable extension of $\mathbf{Q}$. We will obtain a contradiction by using $\mathbf{T}$ to define a universal computable relation.

Let $R(x, y)$ hold if and only if

x codes a formula $\theta(u)$, and $\mathbf{T}$ proves $\theta(\bar{y})$.

Since we are assuming that $\mathbf{T}$ is decidable, R is computable. Let us show that R is universal. If $S(y)$ is any computable relation, then it is representable in $\mathbf{Q}$ (and hence $\mathbf{T}$) by a formula $\theta_S(u)$. Then for every n , we have

$$\begin{aligned} S(\bar{n}) &\rightarrow T \vdash \theta_S(\bar{n}) \\ &\rightarrow R(\ulcorner \theta_S(u) \urcorner^\#, n) \end{aligned}$$

and

$$\begin{aligned} \neg S(\bar{n}) &\rightarrow T \vdash \neg \theta_S(\bar{n}) \\ &\rightarrow T \nvdash \theta_S(\bar{n}) \quad (\text{since } \mathbf{T} \text{ is consistent}) \\ &\rightarrow \neg R(\ulcorner \theta_S(u) \urcorner^\#, n). \end{aligned}$$

That is, for every y , $S(y)$ is true if and only if $R(\ulcorner \theta_S(u) \urcorner^\#, y)$ is. So R is universal, and we have the contradiction we were looking for. $\square$

Let “true arithmetic” be the theory $\{\varphi : N \models \varphi\}$, that is, the set of sentences in the language of arithmetic that are true in the standard interpretation.

Corolário 36.6. *True arithmetic is not decidable.*

-NoValue-

[content/incompleteness/theories-computability/computably-axiomatizable.tex](#)

36.5 Axiomatizável Theories

A theory $\mathbf{T}$ is said to be *axiomatizável* if it has a computable set of axioms A . (Saying that A is a set of axioms for $\mathbf{T}$ means $T = \{\varphi : A \vdash \varphi\}$.) Any “reasonable” axiomatization of the natural numbers will have this property. In particular, any theory with a finite set of axioms is *axiomatizável*. inc:tep:cax:
sec

Lema 36.7. *Suppose $\mathbf{T}$ is axiomatizável. Then $\mathbf{T}$ is computavelmente enumerável.*

Prova. Suppose A is a computable set of axioms for $\mathbf{T}$. To determine if $\varphi \in T$, just search for a *derivação* of φ from the axioms.

Put slightly differently, φ is in $\mathbf{T}$ if and only if there is a finite list of axioms $\psi_1, \dots, \psi_k$ in A and a *derivação* of $(\psi_1 \wedge \dots \wedge \psi_k) \rightarrow \varphi$ in first-order logic. But we already know that any set with a definition of the form “there exists ... such that ...” is *c.e.*, provided the second “...” is computable. $\square$

-NoValue-

content/incompleteness/theories-computability/complete-decidable.tex

36.6 Axiomatizável Complete Theories are Decidable

A theory is said to be *complete* if for every sentence φ , either φ or $\neg\varphi$ is provable. inc:tep:cdc:
sec

Lema 36.8. *Suppose a theory $\mathbf{T}$ is complete and axiomatizável. Then $\mathbf{T}$ is decidable.*

Prova. Suppose $\mathbf{T}$ is complete and A is a computable set of axioms. If $\mathbf{T}$ is inconsistent, it is clearly computable. (Algorithm: “just say yes.”) So we can assume that $\mathbf{T}$ is also consistent.

To decide whether or not a sentence φ is in $\mathbf{T}$, simultaneously search for a *derivação* of φ from $\mathbf{T}$ and a *derivação* of $\neg\varphi$. Since $\mathbf{T}$ is complete, you are bound to find one or the other; and since $\mathbf{T}$ is consistent, if you find a *derivação* of $\neg\varphi$, there is no *derivação* of φ .

Put in different terms, we already know that $\mathbf{T}$ is *c.e.*; so by a theorem we proved before, it suffices to show that the complement of $\mathbf{T}$ is *c.e.* also. But a *fórmula* φ is in $\bar{\mathbf{T}}$ if and only if $\neg\varphi$ is in $\mathbf{T}$; so $\bar{\mathbf{T}} \leq_m \mathbf{T}$. $\square$

-NoValue-

content/incompleteness/theories-computability/first-incompleteness.tex

36.7. **Q HAS NO COMPLETE, CONSISTENT, AXIOMATIZABLE EXTENSIONS**

36.7 Q has no Complete, Consistent, Axiomatizável Extensions

inc:tep:inc:
sec
inc:tep:inc:
thm:first-incompleteness

Teorema 36.9. *There is no complete, consistent, axiomatizável extension of **Q**.*

Prova. We already know that there is no consistent, decidable extension of **Q**. But if **T** is complete and axiomatizado, then it is decidable. $\square$

This theorems is not that far from Gödel's original 1931 formulation of the First Incompleteness Theorem. Aside from the more modern terminology, the key differences are this: Gödel has " ω -consistent" instead of "consistent"; and he could not say "axiomatizável" in full generality, since the formal notion of computability was not in place yet. (The formal models of computability were developed over the following decade, including by Gödel, and in large part to be able to characterize the kinds of theories that are susceptible to the Gödel phenomenon.) explanation

The theorem says you can't have it all, namely, completeness, consistency, and axiomatizabilidade. If you give up any one of these, though, you can have the other two: **Q** is consistent and computably axiomatized, but not complete; the inconsistent theory is complete, and computably axiomatized (say, by $\{0 \neq 0\}$), but not consistent; and the set of true sentence of arithmetic is complete and consistent, but it is not computably axiomatized.

-NoValue-

content/incompleteness/theories-computability/inseparability.tex

36.8 Sentences Provable and Refutable in Q are Computably Inseparable

inc:tep:ins:
sec
Let $\bar{\mathbf{Q}}$ be the set of sentences whose *negations* are provable in **Q**, O.s., $\bar{\mathbf{Q}} = \{\varphi : \mathbf{Q} \vdash \neg\varphi\}$. Remember that disjoint sets A and B are said to be computably inseparable if there is no computable set C such that $A \subseteq C$ and $B \subseteq \bar{C}$.

Lema 36.10. ***Q** and $\bar{\mathbf{Q}}$ are computably inseparable.*

Prova. Suppose C is a computable set such that $\mathbf{Q} \subseteq C$ and $\bar{\mathbf{Q}} \subseteq \bar{C}$. Let $R(x, y)$ be the relation

x codes a formula $\theta(u)$ and $\theta(\bar{y})$ is in C .

We will show that $R(x, y)$ is a universal computable relation, yielding a contradiction.

Suppose $S(y)$ is computable, represented by $\theta_S(u)$ in **Q**. Then

$$\begin{aligned} S(\bar{n}) &\rightarrow \mathbf{Q} \vdash \theta_S(\bar{n}) \\ &\rightarrow \theta_S(\bar{n}) \in C \end{aligned}$$

and

$$\begin{aligned}\neg S(\bar{n}) &\rightarrow \mathbf{Q} \vdash \neg \theta_S(\bar{n}) \\ &\rightarrow \theta_S(\bar{n}) \in \bar{\mathbf{Q}} \\ &\rightarrow \theta_S(\bar{n}) \notin C\end{aligned}$$

So $S(y)$ is equivalent to $R(\#(\theta_S(\bar{u})), y)$. □

-NoValue-

content/incompleteness/theories-computability/consis-with-q.tex

36.9 Theories Consistent with $\mathbf{Q}$ are Undecidable

The following theorem says that not only is $\mathbf{Q}$ undecidable, but, in fact, any theory that does not disagree with $\mathbf{Q}$ is undecidable. inc:tep:con:sec

Teorema 36.11. *Let $\mathbf{T}$ be any theory in the language of arithmetic that is consistent with $\mathbf{Q}$ (O.s., $\mathbf{T} \cup \mathbf{Q}$ is consistent). Then $\mathbf{T}$ is undecidable.*

Prova. Remember that $\mathbf{Q}$ has a finite set of axioms, $Q_1, \dots, Q_8$. We can even replace these by a single axiom, $\alpha = Q_1 \wedge \dots \wedge Q_8$.

Suppose $\mathbf{T}$ is a decidable theory consistent with $\mathbf{Q}$. Let

$$C = \{\varphi : \mathbf{T} \vdash \alpha \rightarrow \varphi\}.$$

We show that C would be a computable separation of $\mathbf{Q}$ and $\bar{\mathbf{Q}}$, a contradiction. First, if φ is in $\mathbf{Q}$, then φ is provable from the axioms of $\mathbf{Q}$; by the deduction theorem, there is a derivação of $\alpha \rightarrow \varphi$ in first-order logic. So φ is in C .

On the other hand, if φ is in $\bar{\mathbf{Q}}$, then there is a proof of $\alpha \rightarrow \neg\varphi$ in first-order logic. If $\mathbf{T}$ also proves $\alpha \rightarrow \varphi$, then $\mathbf{T}$ proves $\neg\alpha$, in which case $\mathbf{T} \cup \mathbf{Q}$ is inconsistent. But we are assuming $\mathbf{T} \cup \mathbf{Q}$ is consistent, so $\mathbf{T}$ does not prove $\alpha \rightarrow \varphi$, and so φ is not in C .

We've shown that if φ is in $\mathbf{Q}$, then it is in C , and if φ is in $\bar{\mathbf{Q}}$, then it is in $\bar{C}$. So C is a computable separation, which is the contradiction we were looking for. □

This theorem is very powerful. For example, it implies:

Corolário 36.12. *First-order logic for the language of arithmetic (that is, the set $\{\varphi : \varphi \text{ is provable in first-order logic}\}$) is undecidable.*

Prova. First-order logic is the set of consequences of $\emptyset$, which is consistent with $\mathbf{Q}$. □

-NoValue-

content/incompleteness/theories-computability/interpretability.tex

36.10 Theories in which $\mathbf{Q}$ is Interpretable are Undecidable

inc:tep:itp:
sec We can strengthen these results even more. Informally, an interpretation of a language $\mathcal{L}_1$ in another language $\mathcal{L}_2$ involves defining the universe, relation symbols, and function symbols of $\mathcal{L}_1$ with *fórmulas* in $\mathcal{L}_2$. Though we won't take the time to do this, one can make this definition precise.

Teorema 36.13. *Suppose $\mathbf{T}$ is a theory in a language in which one can interpret the language of arithmetic, in such a way that $\mathbf{T}$ is consistent with the interpretation of $\mathbf{Q}$. Then $\mathbf{T}$ is undecidable. If $\mathbf{T}$ proves the interpretation of the axioms of $\mathbf{Q}$, then no consistent extension of $\mathbf{T}$ is decidable.*

The proof is just a small modification of the proof of the last theorem; one could use a counterexample to get a separation of $\mathbf{Q}$ and $\bar{\mathbf{Q}}$. One can take **ZFC**, Zermelo–Fraenkel set theory with the axiom of choice, to be an axiomatic foundation that is powerful enough to carry out a good deal of ordinary mathematics. In **ZFC** one can define the natural numbers, and via this interpretation, the axioms of $\mathbf{Q}$ are true. So we have

Corolário 36.14. *There is no decidable extension of **ZFC**.*

Corolário 36.15. *There is no complete, consistent, computably *axiomatizável* extension of **ZFC**.*

The language of **ZFC** has only a single binary relation, $\in$. (In fact, you don't even need equality.) So we have

Corolário 36.16. *First-order logic for any language with a binary relation symbol is undecidable.*

This result extends to any language with two unary function symbols, since one can use these to simulate a binary relation symbol. The results just cited are tight: it turns out that first-order logic for a language with only *unary* relation symbols and at most one *unary* function symbol is decidable.

One more bit of trivia. We know that the set of sentences in the language $0, ', +, \times, <$ true in the standard model is undecidable. In fact, one can define $<$ in terms of the other symbols, and then one can define $+$ in terms of $\times$ and $'$. So the set of true sentences in the language $0, ', \times$ is undecidable. On the other hand, Presburger has shown that the set of sentences in the

language $0, ', +$ true in the language of arithmetic is decidable. The procedure is computationally infeasible, however.

Capítulo 37

Incompleteness and Provability

-NoValue-

`content/incompleteness/incompleteness-provability/introduction.tex`

37.1 Introduction

Hilbert thought that a system of axioms for a mathematical structure, such as the natural numbers, is inadequate unless it allows one to derive all true statements about the structure. Combined with his later interest in formal systems of deduction, this suggests that he thought that we should guarantee that, say, the formal systems we are using to reason about the natural numbers is not only consistent, but also *complete*, O.s., every statement in its language is either *derivável* or its negation is. Gödel's first incompleteness theorem shows that no such system of axioms exists: there is no complete, consistent, *axiomatizável* formal system for arithmetic. In fact, no “sufficiently strong,” consistent, *axiomatizável* mathematical theory is complete. inc:inp:int:
sec

A more important goal of Hilbert's, the centerpiece of his program for the justification of modern (“classical”) mathematics, was to find finitary consistency proofs for formal systems representing classical reasoning. With regard to Hilbert's program, then, Gödel's second incompleteness theorem was a much bigger blow. The second incompleteness theorem can be stated in vague terms, like the first incompleteness theorem. Roughly speaking, it says that no sufficiently strong theory of arithmetic can prove its own consistency. We will have to take “sufficiently strong” to include a little bit more than **Q**.

The idea behind Gödel's original proof of the incompleteness theorem can be found in the Epimenides paradox. Epimenides, a Cretan, asserted that all Cretans are liars; a more direct form of the paradox is the assertion “this sentence is false.” Essentially, by replacing truth with *derivabilidade*, Gödel was able to formalize a *sentença* which, in a roundabout way, asserts that it itself is not *derivável*. If that *sentença* were *derivável*, the theory would then

37.1. INTRODUCTION

be inconsistent. Gödel showed that the negation of that *sentença* is also not *derivável* from the system of axioms he was considering. (For this second part, Gödel had to assume that the theory **T** is what's called " ω -consistent." ω -Consistency is related to consistency, but is a stronger property.¹ A few years after Gödel, Rosser showed that assuming simple consistency of **T** is enough.)

The first challenge is to understand how one can construct a *sentença* that refers to itself. For every *fórmula* φ in the language of **Q**, let $\ulcorner \varphi \urcorner$ denote the numeral corresponding to $\# \varphi \#$. Think about what this means: φ is a *fórmula* in the language of **Q**, $\# \varphi \#$ is a natural number, and $\ulcorner \varphi \urcorner$ is a *term* in the language of **Q**. So every *fórmula* φ in the language of **Q** has a *name*, $\ulcorner \varphi \urcorner$, which is a term in the language of **Q**; this provides us with a conceptual framework in which *fórmulas* in the language of **Q** can "say" things about other *fórmulas*. The following lemma is known as the fixed-point lemma.

Lema 37.1. *Let **T** be any theory extending **Q**, and let $\psi(x)$ be any *fórmula* with only the variable x free. Then there is a *sentença* φ such that $\mathbf{T} \vdash \varphi \leftrightarrow \psi(\ulcorner \varphi \urcorner)$.*

The lemma asserts that given any property $\psi(x)$, there is a *sentença* φ that asserts " $\psi(x)$ is true of me," and **T** "knows" this.

How can we construct such a *sentença*? Consider the following version of the Epimenides paradox, due to Quine:

"Yields falsehood when preceded by its quotation" yields falsehood
when preceded by its quotation.

This sentence is not directly self-referential. It simply makes an assertion about the syntactic objects between quotes, and, in doing so, it is on par with sentences like

1. "Robert" is a nice name.
2. "I ran." is a short sentence.
3. "Has three words" has three words.

But what happens when one takes the phrase "yields falsehood when preceded by its quotation," and precedes it with a quoted version of itself? Then one has the original sentence! In short, the sentence asserts that it is false.

-NoValue-

content/incompleteness/incompleteness-provability/fixed-point-lemma.tex

¹That is, any ω -consistent theory is consistent, but not vice versa.

37.2 The Fixed-Point Lemma

explanation The fixed-point lemma says that for any **fórmula** $\psi(x)$, there is a **sentença** φ such that $\mathbf{T} \vdash \varphi \leftrightarrow \psi(\ulcorner \varphi \urcorner)$, provided $\mathbf{T}$ extends $\mathbf{Q}$. In the case of the liar sentence, we'd want φ to be equivalent (provably in $\mathbf{T}$) to “ $\ulcorner \varphi \urcorner$ is false,” O.s., the statement that $\# \varphi$ is the Gödel number of a false **sentença**. To understand the idea of the proof, it will be useful to compare it with Quine's informal gloss of φ as, “‘yields a falsehood when preceded by its own quotation’ yields a falsehood when preceded by its own quotation.” The operation of taking an expression, and then forming a sentence by preceding this expression by its own quotation may be called *diagonalizing* the expression, and the result its diagonalization. So, the diagonalization of ‘yields a falsehood when preceded by its own quotation’ is “‘yields a falsehood when preceded by its own quotation’ yields a falsehood when preceded by its own quotation.” Now note that Quine's liar sentence is not the diagonalization of ‘yields a falsehood’ but of ‘yields a falsehood when preceded by its own quotation.’ So the property being diagonalized to yield the liar sentence itself involves diagonalization!

In the language of arithmetic, we form quotations of a **fórmula** with one free variable by computing its Gödel numbers and then substituting the standard numeral for that Gödel number into the free variable. The diagonalization of $\alpha(x)$ is $\alpha(\bar{n})$, where $n = \# \alpha(x)$. (From now on, let's abbreviate $\# \alpha(x)$ as $\ulcorner \alpha(x) \urcorner$.) So if $\psi(x)$ is “is a falsehood,” then “yields a falsehood if preceded by its own quotation,” would be “yields a falsehood when applied to the Gödel number of its diagonalization.” If we had a symbol *diag* for the function $\text{diag}(n)$ which computes the Gödel number of the diagonalization of the **fórmula** with Gödel number n , we could write $\alpha(x)$ as $\psi(\text{diag}(x))$. And Quine's version of the liar sentence would then be the diagonalization of it, O.s., $\alpha(\ulcorner \alpha(x) \urcorner)$ or $\psi(\text{diag}(\ulcorner \psi(\text{diag}(x)) \urcorner))$. Of course, $\psi(x)$ could now be any other property, and the same construction would work. For the incompleteness theorem, we'll take $\psi(x)$ to be “ x is not **derivável** in $\mathbf{T}$.” Then $\alpha(x)$ would be “yields a **sentença** not **derivável** in $\mathbf{T}$ when applied to the Gödel number of its diagonalization.”

To formalize this in $\mathbf{T}$, we have to find a way to formalize *diag*. The function $\text{diag}(n)$ is computable, in fact, it is primitive recursive: if n is the Gödel number of a formula $\alpha(x)$, $\text{diag}(n)$ returns the Gödel number of $\alpha(\ulcorner \alpha(x) \urcorner)$. (Recall, $\ulcorner \alpha(x) \urcorner$ is the standard numeral of the Gödel number of $\alpha(x)$, O.s., $\# \alpha(x)$.) If *diag* were a function symbol in $\mathbf{T}$ representing the function *diag*, we could take φ to be the formula $\psi(\text{diag}(\ulcorner \psi(\text{diag}(x)) \urcorner))$. Notice that

$$\begin{aligned} \text{diag}(\# \psi(\text{diag}(x))) &= \# \psi(\text{diag}(\ulcorner \psi(\text{diag}(x)) \urcorner)) \\ &= \# \varphi. \end{aligned}$$

Assuming $\mathbf{T}$ can **deriva**

$$\text{diag}(\ulcorner \psi(\text{diag}(x)) \urcorner) = \ulcorner \varphi \urcorner,$$

it can **deriva** $\psi(\text{diag}(\ulcorner \psi(\text{diag}(x)) \urcorner)) \leftrightarrow \psi(\ulcorner \varphi \urcorner)$. But the left hand side is, by definition, φ .

37.2. THE FIXED-POINT LEMMA

Of course, *diag* will in general not be a function symbol of **T**, and certainly is not one of **Q**. But, since *diag* is computable, it is *representable* in **Q** by some formula $\theta_{\text{diag}}(x, y)$. So instead of writing $\psi(\text{diag}(x))$ we can write $\exists y (\theta_{\text{diag}}(x, y) \wedge \psi(y))$. Otherwise, the proof sketched above goes through, and in fact, it goes through already in **Q**.

inc:inp:fix: lem:fixed-point **Lema 37.2.** *Let $\psi(x)$ be any formula with one free variable x . Then there is a sentence φ such that $\mathbf{Q} \vdash \varphi \leftrightarrow \psi(\ulcorner \varphi \urcorner)$.*

Prova. Given $\psi(x)$, let $\alpha(x)$ be the formula $\exists y (\theta_{\text{diag}}(x, y) \wedge \psi(y))$ and let φ be its diagonalization, O.s., the formula $\alpha(\ulcorner \alpha(x) \urcorner)$.

Since θ_{diag} represents *diag*, and $\text{diag}(\ulcorner \alpha(x) \urcorner) = \ulcorner \varphi \urcorner$, **Q** can *deriva*

$$\theta_{\text{diag}}(\ulcorner \alpha(x) \urcorner, \ulcorner \varphi \urcorner) \quad (37.1)$$

$$\forall y (\theta_{\text{diag}}(\ulcorner \alpha(x) \urcorner, y) \rightarrow y = \ulcorner \varphi \urcorner). \quad (37.2)$$

Now we show that $\mathbf{Q} \vdash \varphi \leftrightarrow \psi(\ulcorner \varphi \urcorner)$. We argue informally, using just logic and facts *derivável* in **Q**.

First, suppose φ , O.s., $\alpha(\ulcorner \alpha(x) \urcorner)$. Going back to the definition of $\alpha(x)$, we see that $\alpha(\ulcorner \alpha(x) \urcorner)$ just is

$$\exists y (\theta_{\text{diag}}(\ulcorner \alpha(x) \urcorner, y) \wedge \psi(y)).$$

Consider such a y . Since $\theta_{\text{diag}}(\ulcorner \alpha(x) \urcorner, y)$, by **Eq. (37.2)**, $y = \ulcorner \varphi \urcorner$. So, from $\psi(y)$ we have $\psi(\ulcorner \varphi \urcorner)$.

Now suppose $\psi(\ulcorner \varphi \urcorner)$. By **Eq. (37.1)**, we have

$$\theta_{\text{diag}}(\ulcorner \alpha(x) \urcorner, \ulcorner \varphi \urcorner) \wedge \psi(\ulcorner \varphi \urcorner).$$

It follows that

$$\exists y (\theta_{\text{diag}}(\ulcorner \alpha(x) \urcorner, y) \wedge \psi(y)).$$

But that's just $\alpha(\ulcorner \alpha(x) \urcorner)$, O.s., φ . □

You should compare this to the proof of the fixed-point lemma in computability theory. The difference is that here we want to define a *statement* in terms of itself, whereas there we wanted to define a *function* in terms of itself; this difference aside, it is really the same idea. *digression*

Problema 37.1. A fórmula $\varphi(x)$ is a *truth definition* if $\mathbf{Q} \vdash \psi \leftrightarrow \varphi(\ulcorner \psi \urcorner)$ for all *sentenças* ψ . Show that no *fórmula* is a truth definition by using the fixed-point lemma.

-NoValue-

content/incompleteness/incompleteness-provability/first-incompleteness-thm.tex

37.3 The First Incompleteness Theorem

We can now describe Gödel's original proof of the first incompleteness theorem. Let $\mathbf{T}$ be any computably axiomatized theory in a language extending the language of arithmetic, such that $\mathbf{T}$ includes the axioms of $\mathbf{Q}$. This means that, in particular, $\mathbf{T}$ represents computable functions and relations.

inc:inp:lin:
sec

We have argued that, given a reasonable coding of formulas and proofs as numbers, the relation $\text{Prf}_{\mathbf{T}}(x, y)$ is computable, where $\text{Prf}_{\mathbf{T}}(x, y)$ holds if and only if x is the Gödel number of a *derivação* of the *fórmula* with Gödel number y in $\mathbf{T}$. In fact, for the particular theory that Gödel had in mind, Gödel was able to show that this relation is primitive recursive, using the list of 45 functions and relations in his paper. The 45th relation, $xB y$, is just $\text{Prf}_{\mathbf{T}}(x, y)$ for his particular choice of $\mathbf{T}$. Remember that where Gödel uses the word “recursive” in his paper, we would now use the phrase “primitive recursive.”

Since $\text{Prf}_{\mathbf{T}}(x, y)$ is computable, it is representable in $\mathbf{T}$. We will use $\text{Prf}_{\mathbf{T}}(x, y)$ to refer to the formula that represents it. Let $\text{Prov}_{\mathbf{T}}(y)$ be the formula $\exists x \text{Prf}_{\mathbf{T}}(x, y)$. This describes the 46th relation, $\text{Bew}(y)$, on Gödel's list. As Gödel notes, this is the only relation that “cannot be asserted to be recursive.” What he probably meant is this: from the definition, it is not clear that it is computable; and later developments, in fact, show that it isn't.

Let $\mathbf{T}$ be an *axiomatizável* theory containing $\mathbf{Q}$. Then $\text{Prf}_{\mathbf{T}}(x, y)$ is decidable, hence representable in $\mathbf{Q}$ by a *fórmula* $\text{Prf}_{\mathbf{T}}(x, y)$. Let $\text{Prov}_{\mathbf{T}}(y)$ be the formula we described above. By the fixed-point lemma, there is a formula $\gamma_{\mathbf{T}}$ such that $\mathbf{Q}$ (and hence $\mathbf{T}$) *deriva*

$$\gamma_{\mathbf{T}} \leftrightarrow \neg \text{Prov}_{\mathbf{T}}(\ulcorner \gamma_{\mathbf{T}} \urcorner). \quad (37.3)$$

inc:inp:lin:
eqn:qpf

Note that $\gamma_{\mathbf{T}}$ says, in essence, “ $\gamma_{\mathbf{T}}$ is not *derivável* in $\mathbf{T}$.”

Lema 37.3. *If $\mathbf{T}$ is a consistent, *axiomatizável* theory extending $\mathbf{Q}$, then $\mathbf{T} \not\vdash \gamma_{\mathbf{T}}$.*

inc:inp:lin:
lem:cons-G-unprov

Prova. Suppose $\mathbf{T}$ *deriva* $\gamma_{\mathbf{T}}$. Then there is a *derivação*, and so, for some number m , the relation $\text{Prf}_{\mathbf{T}}(m, \# \gamma_{\mathbf{T}})$ holds. But then $\mathbf{Q}$ *deriva* the sentence $\text{Prf}_{\mathbf{T}}(\overline{m}, \ulcorner \gamma_{\mathbf{T}} \urcorner)$. So $\mathbf{Q}$ *deriva* $\exists x \text{Prf}_{\mathbf{T}}(x, \ulcorner \gamma_{\mathbf{T}} \urcorner)$, which is, by definition, $\text{Prov}_{\mathbf{T}}(\ulcorner \gamma_{\mathbf{T}} \urcorner)$. By Eq. (37.3), $\mathbf{Q}$ *deriva* $\neg \gamma_{\mathbf{T}}$, and since $\mathbf{T}$ extends $\mathbf{Q}$, so does $\mathbf{T}$. We have shown that if $\mathbf{T}$ *deriva* $\gamma_{\mathbf{T}}$, then it also *deriva* $\neg \gamma_{\mathbf{T}}$, and hence it would be inconsistent. $\square$

Definição 37.4. A theory $\mathbf{T}$ is *ω -consistent* if the following holds: if $\exists x \varphi(x)$ is any sentence and $\mathbf{T}$ *deriva* $\neg \varphi(\overline{0})$, $\neg \varphi(\overline{1})$, $\neg \varphi(\overline{2})$, ... then $\mathbf{T}$ does not prove $\exists x \varphi(x)$.

inc:inp:lin:
thm:oconsis-q

Note that every ω -consistent theory is also consistent. This follows simply from the fact that if $\mathbf{T}$ is inconsistent, then $\mathbf{T} \vdash \varphi$ for every φ . In particular, if $\mathbf{T}$ is inconsistent, it *deriva* both $\neg \varphi(\overline{n})$ for every n and also *deriva* $\exists x \varphi(x)$. So, if $\mathbf{T}$ is inconsistent, it is ω -inconsistent. By contraposition, if $\mathbf{T}$ is ω -consistent, it must be consistent.

37.4. ROSSER'S THEOREM

inc:inp:1in: lem:omega-cons-G-unref **Lema 37.5.** *If $\mathbf{T}$ is an ω -consistent, axiomatizável theory extending $\mathbf{Q}$, then $\mathbf{T} \not\vdash \neg\gamma_{\mathbf{T}}$.*

Prova. We show that if $\mathbf{T}$ deriva $\neg\gamma_{\mathbf{T}}$, then it is ω -inconsistent. Suppose $\mathbf{T}$ deriva $\neg\gamma_{\mathbf{T}}$. If $\mathbf{T}$ is inconsistent, it is ω -inconsistent, and we are done. Otherwise, $\mathbf{T}$ is consistent, so it does not deriva $\gamma_{\mathbf{T}}$ by Lema 37.3. Since there is no derivação of $\gamma_{\mathbf{T}}$ in $\mathbf{T}$, $\mathbf{Q}$ deriva

$$\neg\text{Prf}_{\mathbf{T}}(\bar{0}, \ulcorner\gamma_{\mathbf{T}}\urcorner), \neg\text{Prf}_{\mathbf{T}}(\bar{1}, \ulcorner\gamma_{\mathbf{T}}\urcorner), \neg\text{Prf}_{\mathbf{T}}(\bar{2}, \ulcorner\gamma_{\mathbf{T}}\urcorner), \dots$$

and so does $\mathbf{T}$. On the other hand, by Eq. (37.3), $\neg\gamma_{\mathbf{T}}$ is equivalent to $\exists x \text{Prf}_{\mathbf{T}}(x, \ulcorner\gamma_{\mathbf{T}}\urcorner)$. So $\mathbf{T}$ is ω -inconsistent. $\square$

Problema 37.2. Every ω -consistent theory is consistent. Show that the converse does not hold, O.s., that there are consistent but ω -inconsistent theories. Do this by showing that $\mathbf{Q} \cup \{\neg\gamma_{\mathbf{Q}}\}$ is consistent but ω -inconsistent.

inc:inp:1in: thm:first-incompleteness **Teorema 37.6.** *Let $\mathbf{T}$ be any ω -consistent, axiomatizável theory extending $\mathbf{Q}$. Then $\mathbf{T}$ is not complete.*

Prova. If $\mathbf{T}$ is ω -consistent, it is consistent, so $\mathbf{T} \not\vdash \gamma_{\mathbf{T}}$ by Lema 37.3. By Lema 37.5, $\mathbf{T} \not\vdash \neg\gamma_{\mathbf{T}}$. This means that $\mathbf{T}$ is incomplete, since it deriva neither $\gamma_{\mathbf{T}}$ nor $\neg\gamma_{\mathbf{T}}$. $\square$

-NoValue-

content/incompleteness/incompleteness-provability/rosser-thm.tex

37.4 Rosser's Theorem

inc:inp:ros: sec Can we modify Gödel's proof to get a stronger result, replacing “ ω -consistent” with simply “consistent”? The answer is “yes,” using a trick discovered by Rosser. Rosser's trick is to use a “modified” derivabilidade predicate $\text{RProv}_{\mathbf{T}}(y)$ instead of $\text{Prov}_{\mathbf{T}}(y)$.

inc:inp:ros: thm:rosser **Teorema 37.7.** *Let $\mathbf{T}$ be any consistent, axiomatizável theory extending $\mathbf{Q}$. Then $\mathbf{T}$ is not complete.*

Prova. Recall that $\text{Prov}_{\mathbf{T}}(y)$ is defined as $\exists x \text{Prf}_{\mathbf{T}}(x, y)$, where $\text{Prf}_{\mathbf{T}}(x, y)$ represents the decidable relation which holds iff x is the Gödel number of a derivação of the sentença with Gödel number y . The relation that holds between x and y if x is the Gödel number of a refutation of the sentence with Gödel number y is also decidable. Let $\text{not}(x)$ be the primitive recursive function which does the following: if x is the code of a formula φ , $\text{not}(x)$ is a code of $\neg\varphi$. Then $\text{Ref}_{\mathbf{T}}(x, y)$ holds iff $\text{Prf}_{\mathbf{T}}(x, \text{not}(y))$. Let $\text{Ref}_{\mathbf{T}}(x, y)$ represent it. Then,

if $\mathbf{T} \vdash \neg\varphi$ and δ is a corresponding *derivação*, $\mathbf{Q} \vdash \text{Ref}_{\mathbf{T}}(\ulcorner\delta\urcorner, \ulcorner\varphi\urcorner)$. We define $\text{RProv}_{\mathbf{T}}(y)$ as

$$\exists x (\text{Prf}_{\mathbf{T}}(x, y) \wedge \forall z (z < x \rightarrow \neg \text{Ref}_{\mathbf{T}}(z, y))).$$

Roughly, $\text{RProv}_{\mathbf{T}}(y)$ says “there is a proof of y in $\mathbf{T}$, and there is no shorter refutation of y .” Assuming $\mathbf{T}$ is consistent, $\text{RProv}_{\mathbf{T}}(y)$ is true of the same numbers as $\text{Prov}_{\mathbf{T}}(y)$; but from the point of view of *provability* in $\mathbf{T}$ (and we now know that there is a difference between truth and provability!) the two have different properties. If $\mathbf{T}$ is *inconsistent*, then the two do *not* hold of the same numbers! ($\text{RProv}_{\mathbf{T}}(y)$ is often read as “ y is Rosser provable.” Since, as just discussed, Rosser provability is not some special kind of provability—in inconsistent theories, there are *sentenças* that are provable but not Rosser provable—this may be confusing. To avoid the confusion, you could instead read it as “ y is shmovable.”)

By the fixed-point lemma, there is a formula $\rho_{\mathbf{T}}$ such that

$$\mathbf{Q} \vdash \rho_{\mathbf{T}} \leftrightarrow \neg \text{RProv}_{\mathbf{T}}(\ulcorner\rho_{\mathbf{T}}\urcorner). \quad (37.4) \quad \text{inc:inp:ros:RT}$$

In contrast to the proof of *Teorema 37.6*, here we claim that if $\mathbf{T}$ is consistent, $\mathbf{T}$ doesn’t *deriva* $\rho_{\mathbf{T}}$, and $\mathbf{T}$ also doesn’t *deriva* $\neg\rho_{\mathbf{T}}$. (In other words, we don’t need the assumption of ω -consistency.)

First, let’s show that $\mathbf{T} \not\vdash \rho_{\mathbf{T}}$. Suppose it did, so there is a *derivação* of $\rho_{\mathbf{T}}$ from T ; let n be its Gödel number. Then $\mathbf{Q} \vdash \text{Prf}_{\mathbf{T}}(\bar{n}, \ulcorner\rho_{\mathbf{T}}\urcorner)$, since $\text{Prf}_{\mathbf{T}}$ represents $\text{Prf}_{\mathbf{T}}$ in $\mathbf{Q}$. Also, for each $k < n$, k is not the Gödel number of a *derivação* of $\neg\rho_{\mathbf{T}}$, since $\mathbf{T}$ is consistent. So for each $k < n$, $\mathbf{Q} \vdash \neg \text{Ref}_{\mathbf{T}}(\bar{k}, \ulcorner\rho_{\mathbf{T}}\urcorner)$. By *Lema 35.24*, $\mathbf{Q} \vdash \forall z (z < \bar{n} \rightarrow \neg \text{Ref}_{\mathbf{T}}(z, \ulcorner\rho_{\mathbf{T}}\urcorner))$. Thus,

$$\mathbf{Q} \vdash \exists x (\text{Prf}_{\mathbf{T}}(x, \ulcorner\rho_{\mathbf{T}}\urcorner) \wedge \forall z (z < x \rightarrow \neg \text{Ref}_{\mathbf{T}}(z, \ulcorner\rho_{\mathbf{T}}\urcorner))),$$

but that’s just $\text{RProv}_{\mathbf{T}}(\ulcorner\rho_{\mathbf{T}}\urcorner)$. By *Eq. (37.4)*, $\mathbf{Q} \vdash \neg\rho_{\mathbf{T}}$. Since $\mathbf{T}$ extends $\mathbf{Q}$, also $\mathbf{T} \vdash \neg\rho_{\mathbf{T}}$. We’ve assumed that $\mathbf{T} \vdash \rho_{\mathbf{T}}$, so $\mathbf{T}$ would be inconsistent, contrary to the assumption of the theorem.

Now, let’s show that $\mathbf{T} \not\vdash \neg\rho_{\mathbf{T}}$. Again, suppose it did, and suppose n is the Gödel number of a *derivação* of $\neg\rho_{\mathbf{T}}$. Then $\text{Ref}_{\mathbf{T}}(n, \ulcorner\neg\rho_{\mathbf{T}}\urcorner)$ holds, and since $\text{Ref}_{\mathbf{T}}$ represents $\text{Ref}_{\mathbf{T}}$ in $\mathbf{Q}$, $\mathbf{Q} \vdash \text{Ref}_{\mathbf{T}}(\bar{n}, \ulcorner\neg\rho_{\mathbf{T}}\urcorner)$. We’ll again show that $\mathbf{T}$ would then be inconsistent because it would also *deriva* $\rho_{\mathbf{T}}$. Since

$$\mathbf{Q} \vdash \rho_{\mathbf{T}} \leftrightarrow \neg \text{RProv}_{\mathbf{T}}(\ulcorner\rho_{\mathbf{T}}\urcorner),$$

and since $\mathbf{T}$ extends $\mathbf{Q}$, it suffices to show that

$$\mathbf{Q} \vdash \neg \text{RProv}_{\mathbf{T}}(\ulcorner\rho_{\mathbf{T}}\urcorner).$$

The *sentença* $\neg \text{RProv}_{\mathbf{T}}(\ulcorner\rho_{\mathbf{T}}\urcorner)$, O.s.,

$$\neg \exists x (\text{Prf}_{\mathbf{T}}(x, \ulcorner\rho_{\mathbf{T}}\urcorner) \wedge \forall z (z < x \rightarrow \neg \text{Ref}_{\mathbf{T}}(z, \ulcorner\rho_{\mathbf{T}}\urcorner))),$$

37.5. COMPARISON WITH GÖDEL'S ORIGINAL PAPER

is logically equivalent to

$$\forall x (\text{Prf}_{\mathbf{T}}(x, \ulcorner \rho_{\mathbf{T}} \urcorner) \rightarrow \exists z (z < x \wedge \text{Ref}_{\mathbf{T}}(z, \ulcorner \rho_{\mathbf{T}} \urcorner))).$$

We argue informally using logic, making use of facts about what **Q** *deriva*. Suppose x is arbitrary and $\text{Prf}_{\mathbf{T}}(x, \ulcorner \rho_{\mathbf{T}} \urcorner)$. We already know that $\mathbf{T} \not\vdash \rho_{\mathbf{T}}$, and so for every k , $\mathbf{Q} \vdash \neg \text{Prf}_{\mathbf{T}}(\bar{k}, \ulcorner \rho_{\mathbf{T}} \urcorner)$. Thus, for every k it follows that $x \neq \bar{k}$. In particular, we have (a) that $x \neq \bar{n}$. We also have $\neg(x = \bar{0} \vee x = \bar{1} \vee \dots \vee x = \overline{n-1})$ and so by [Lema 35.24](#), (b) $\neg(x < \bar{n})$. By [Lema 35.25](#), $\bar{n} < x$. Since $\mathbf{Q} \vdash \text{Ref}_{\mathbf{T}}(\bar{n}, \ulcorner \rho_{\mathbf{T}} \urcorner)$, we have $\bar{n} < x \wedge \text{Ref}_{\mathbf{T}}(\bar{n}, \ulcorner \rho_{\mathbf{T}} \urcorner)$, and from that $\exists z (z < x \wedge \text{Ref}_{\mathbf{T}}(z, \ulcorner \rho_{\mathbf{T}} \urcorner))$. Since x was arbitrary we get, as required, that

$$\forall x (\text{Prf}_{\mathbf{T}}(x, \ulcorner \rho_{\mathbf{T}} \urcorner) \rightarrow \exists z (z < x \wedge \text{Ref}_{\mathbf{T}}(z, \ulcorner \rho_{\mathbf{T}} \urcorner))). \quad \square$$

Problema 37.3. Two sets A and B of natural numbers are said to be *computably inseparable* if there is no *decidível* set X such that $A \subseteq X$ and $B \subseteq \bar{X}$ ($\bar{X}$ is the complement, $\mathbb{N} \setminus X$, of X). Let $\mathbf{T}$ be a consistent *axiomatizável* extension of **Q**. Suppose A is the set of Gödel numbers of *sentenças* provable in $\mathbf{T}$ and B the set of Gödel numbers of sentences refutable in $\mathbf{T}$. Prove that A and B are computably inseparable.

-NoValue-

[content/incompleteness/incompleteness-provability/godels-paper.tex](#)

37.5 Comparison with Gödel's Original Paper

[inc:inp:gop:](#)
[sec](#)

It is worthwhile to spend some time with Gödel's 1931 paper. The introduction sketches the ideas we have just discussed. Even if you just skim through the paper, it is easy to see what is going on at each stage: first Gödel describes the formal system P (syntax, axioms, proof rules); then he defines the primitive recursive functions and relations; then he shows that xBy is primitive recursive, and argues that the primitive recursive functions and relations are represented in **P**. He then goes on to prove the incompleteness theorem, as above. In Section 3, he shows that one can take the unprovable assertion to be a sentence in the language of arithmetic. This is the origin of the β -lemma, which is what we also used to handle sequences in showing that the recursive functions are representable in **Q**. Gödel doesn't go so far to isolate a minimal set of axioms that suffice, but we now know that **Q** will do the trick. Finally, in Section 4, he sketches a proof of the second incompleteness theorem.

-NoValue-

[content/incompleteness/incompleteness-provability/provability-conditions.tex](#)

37.6 The Derivabilidade Conditions for PA

Peano arithmetic, or **PA**, is the theory extending **Q** with induction axioms for all **fórmulas**. In other words, one adds to **Q** axioms of the form inc:inp:prc:sec

$$(\varphi(0) \wedge \forall x (\varphi(x) \rightarrow \varphi(x'))) \rightarrow \forall x \varphi(x)$$

for every **fórmula** φ . Notice that this is really a *schema*, which is to say, infinitely many axioms (and it turns out that **PA** is *not* finitely axiomatizable). But since one can effectively determine whether or not a string of symbols is an instance of an induction axiom, the set of axioms for **PA** is computable. **PA** is a much more robust theory than **Q**. For example, one can easily prove that addition and multiplication are commutative, using induction in the usual way. In fact, most finitary number-theoretic and combinatorial arguments can be carried out in **PA**.

Since **PA** is computably axiomatized, the **derivabilidade** predicate $\text{Prf}_{\mathbf{PA}}(x, y)$ is computable and hence represented in **Q** (and so, in **PA**). As before, we will take $\text{Prf}_{\mathbf{PA}}(x, y)$ to denote the formula representing the relation. Let $\text{Prov}_{\mathbf{PA}}(y)$ be the formula $\exists x \text{Prf}_{\mathbf{PA}}(x, y)$, which, intuitively says, “ y is **derivável** from the axioms of **PA**.” The reason we need a little bit more than the axioms of **Q** is we need to know that the theory we are using is strong enough to **deriva** a few basic facts about this **derivabilidade** predicate. In fact, what we need are the following facts:

P1. If $\mathbf{PA} \vdash \varphi$, then $\mathbf{PA} \vdash \text{Prov}_{\mathbf{PA}}(\ulcorner \varphi \urcorner)$.

P2. For all **fórmulas** φ and ψ ,

$$\mathbf{PA} \vdash \text{Prov}_{\mathbf{PA}}(\ulcorner \varphi \rightarrow \psi \urcorner) \rightarrow (\text{Prov}_{\mathbf{PA}}(\ulcorner \varphi \urcorner) \rightarrow \text{Prov}_{\mathbf{PA}}(\ulcorner \psi \urcorner)).$$

P3. For every **fórmula** φ ,

$$\mathbf{PA} \vdash \text{Prov}_{\mathbf{PA}}(\ulcorner \varphi \urcorner) \rightarrow \text{Prov}_{\mathbf{PA}}(\ulcorner \text{Prov}_{\mathbf{PA}}(\ulcorner \varphi \urcorner) \urcorner).$$

The only way to verify that these three properties hold is to describe the **fórmula** $\text{Prov}_{\mathbf{PA}}(y)$ carefully and use the axioms of **PA** to describe the relevant formal **derivações**. Conditions (1) and (2) are easy; it is really condition (3) that requires work. (Think about what kind of work it entails ...) Carrying out the details would be tedious and uninteresting, so here we will ask you to take it on faith that **PA** has the three properties listed above. A reasonable choice of $\text{Prov}_{\mathbf{PA}}(y)$ will also satisfy

P4. If $\mathbf{PA} \vdash \text{Prov}_{\mathbf{PA}}(\ulcorner \varphi \urcorner)$, then $\mathbf{PA} \vdash \varphi$.

But we will not need this fact.

digression

Incidentally, Gödel was lazy in the same way we are being now. At the end of the 1931 paper, he sketches the proof of the second incompleteness theorem, and promises the details in a later paper. He never got around to it; since

37.7. THE SECOND INCOMPLETENESS THEOREM

everyone who understood the argument believed that it could be carried out (he did not need to fill in the details.)

-NoValue-

content/incompleteness/incompleteness-provability/second-incompleteness-thm.tex

37.7 The Second Incompleteness Theorem

inc:inp:2in:
sec How can we express the assertion that **PA** doesn't prove its own consistency? Saying **PA** is inconsistent amounts to saying that $\mathbf{PA} \vdash 0 = 1$. So we can take the consistency statement $\mathbf{Con}_{\mathbf{PA}}$ to be the **sentença** $\neg \mathbf{Prov}_{\mathbf{PA}}(\ulcorner 0 = 1 \urcorner)$, and then the following theorem does the job:

inc:inp:2in:
thm:second-incompleteness **Teorema 37.8.** *Assuming **PA** is consistent, then **PA** does not **deriva** $\mathbf{Con}_{\mathbf{PA}}$.*

It is important to note that the theorem depends on the particular representation of $\mathbf{Con}_{\mathbf{PA}}$ (O.s., the particular representation of $\mathbf{Prov}_{\mathbf{PA}}(y)$). All we will use is that the representation of $\mathbf{Prov}_{\mathbf{PA}}(y)$ satisfies the three **derivabilidade** conditions, so the theorem generalizes to any theory with a **derivabilidade** predicate having these properties.

It is informative to read Gödel's sketch of an argument, since the theorem follows like a good punch line. It goes like this. Let $\gamma_{\mathbf{PA}}$ be the Gödel sentence that we constructed in the proof of **Teorema 37.6**. We have shown "If **PA** is consistent, then **PA** does not **deriva** $\gamma_{\mathbf{PA}}$." If we formalize this *in* **PA**, we have a proof of

$$\mathbf{Con}_{\mathbf{PA}} \rightarrow \neg \mathbf{Prov}_{\mathbf{PA}}(\ulcorner \gamma_{\mathbf{PA}} \urcorner).$$

Now suppose **PA** **deriva** $\mathbf{Con}_{\mathbf{PA}}$. Then it **deriva** $\neg \mathbf{Prov}_{\mathbf{PA}}(\ulcorner \gamma_{\mathbf{PA}} \urcorner)$. But since $\gamma_{\mathbf{PA}}$ is a Gödel sentence, this is equivalent to $\gamma_{\mathbf{PA}}$. So **PA** **deriva** $\gamma_{\mathbf{PA}}$.

But: we know that if **PA** is consistent, it doesn't **deriva** $\gamma_{\mathbf{PA}}$! So if **PA** is consistent, it can't **deriva** $\mathbf{Con}_{\mathbf{PA}}$.

To make the argument more precise, we will let $\gamma_{\mathbf{PA}}$ be the Gödel sentence for **PA** and use the **derivabilidade** conditions (P1)–(P3) to show that **PA** **deriva** $\mathbf{Con}_{\mathbf{PA}} \rightarrow \gamma_{\mathbf{PA}}$. This will show that **PA** doesn't **deriva** $\mathbf{Con}_{\mathbf{PA}}$. Here is a sketch

CAPÍTULO 37. INCOMPLETENESS AND PROVABILITY

of the proof, in **PA**. (For simplicity, we drop the **PA** subscripts.)

$$\gamma \leftrightarrow \neg \text{Prov}(\ulcorner \gamma \urcorner) \quad (37.5) \quad \text{inc:inp:2in: G2-1}$$

γ is a Gödel sentence

$$\gamma \rightarrow \neg \text{Prov}(\ulcorner \gamma \urcorner) \quad (37.6) \quad \text{inc:inp:2in: G2-2}$$

from Eq. (37.5)

$$\gamma \rightarrow (\text{Prov}(\ulcorner \gamma \urcorner) \rightarrow \perp) \quad (37.7) \quad \text{inc:inp:2in: G2-3}$$

from Eq. (37.6) by logic

$$\text{Prov}(\ulcorner \gamma \rightarrow (\text{Prov}(\ulcorner \gamma \urcorner) \rightarrow \perp) \urcorner) \quad (37.8) \quad \text{inc:inp:2in: G2-4}$$

by from Eq. (37.7) by condition P1

$$\text{Prov}(\ulcorner \gamma \urcorner) \rightarrow \text{Prov}(\ulcorner \text{Prov}(\ulcorner \gamma \urcorner) \rightarrow \perp \urcorner) \quad (37.9) \quad \text{inc:inp:2in: G2-5}$$

from Eq. (37.8) by condition P2

$$\text{Prov}(\ulcorner \gamma \urcorner) \rightarrow (\text{Prov}(\ulcorner \text{Prov}(\ulcorner \gamma \urcorner) \urcorner) \rightarrow \text{Prov}(\ulcorner \perp \urcorner)) \quad (37.10) \quad \text{inc:inp:2in: G2-6}$$

from Eq. (37.9) by condition P2 and logic

$$\text{Prov}(\ulcorner \gamma \urcorner) \rightarrow \text{Prov}(\ulcorner \text{Prov}(\ulcorner \gamma \urcorner) \urcorner) \quad (37.11) \quad \text{inc:inp:2in: G2-7}$$

by P3

$$\text{Prov}(\ulcorner \gamma \urcorner) \rightarrow \text{Prov}(\ulcorner \perp \urcorner) \quad (37.12) \quad \text{inc:inp:2in: G2-8}$$

from Eq. (37.10) and Eq. (37.11) by logic

$$\text{Con} \rightarrow \neg \text{Prov}(\ulcorner \gamma \urcorner) \quad (37.13) \quad \text{inc:inp:2in: G2-9}$$

contraposition of Eq. (37.12) and $\text{Con} \equiv \neg \text{Prov}(\ulcorner \perp \urcorner)$

$$\text{Con} \rightarrow \gamma$$

from Eq. (37.5) and Eq. (37.13) by logic

The use of logic in the above just elementary facts from propositional logic, p.ex., Eq. (37.7) uses $\vdash \neg \varphi \leftrightarrow (\varphi \rightarrow \perp)$ and Eq. (37.12) uses $\varphi \rightarrow (\psi \rightarrow \chi), \varphi \rightarrow \psi \vdash \varphi \rightarrow \chi$. The use of condition P2 in Eq. (37.9) and Eq. (37.10) relies on instances of P2, $\text{Prov}(\ulcorner \varphi \rightarrow \psi \urcorner) \rightarrow (\text{Prov}(\ulcorner \varphi \urcorner) \rightarrow \text{Prov}(\ulcorner \psi \urcorner))$. In the first one, $\varphi \equiv \gamma$ and $\psi \equiv \text{Prov}(\ulcorner \gamma \urcorner) \rightarrow \perp$; in the second, $\varphi \equiv \text{Prov}(\ulcorner \gamma \urcorner)$ and $\psi \equiv \perp$.

The more abstract version of the second incompleteness theorem is as follows:

Teorema 37.9. *Let **T** be any consistent, axiomatizado theory extending **Q** and let $\text{Prov}_{\mathbf{T}}(y)$ be any formula satisfying derivabilidade conditions P1–P3 for **T**. Then **T** does not deriva $\text{Con}_{\mathbf{T}}$.* inc:inp:2in: thm:second-incompleteness-gen

Problema 37.4. Show that **PA** deriva $\gamma_{\mathbf{PA}} \rightarrow \text{Con}_{\mathbf{PA}}$.

digression

The moral of the story is that no “reasonable” consistent theory for mathematics can deriva its own consistency statement. Suppose **T** is a theory of mathematics that includes **Q** and Hilbert’s “finitary” reasoning (whatever that may be). Then, the whole of **T** cannot deriva the consistency statement of **T**, and so, a fortiori, the finitary fragment can’t deriva the consistency statement

37.8. LÖB'S THEOREM

of $\mathbf{T}$ either. In that sense, there cannot be a finitary consistency proof for “all of mathematics.”

There is some leeway in interpreting the term “finitary,” and Gödel, in the 1931 paper, grants the possibility that something we may consider “finitary” may lie outside the kinds of mathematics Hilbert wanted to formalize. But Gödel was being charitable; today, it is hard to see how we might find something that can reasonably be called finitary but is not formalizable in, say, **ZFC**, Zermelo–Fraenkel set theory with the axiom of choice.

-NoValue-

content/incompleteness/incompleteness-provability/lob-thm.tex

37.8 Löb's Theorem

inc:inp:lob: sec The Gödel sentence for a theory $\mathbf{T}$ is a fixed point of $\neg \text{Prov}_{\mathbf{T}}(y)$, O.s., a **sentença** γ such that

$$\mathbf{T} \vdash \neg \text{Prov}_{\mathbf{T}}(\ulcorner \gamma \urcorner) \leftrightarrow \gamma.$$

It is not **derivável**, because if $\mathbf{T} \vdash \gamma$, (a) by **derivabilidade** condition (1), $\mathbf{T} \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \gamma \urcorner)$, and (b) $\mathbf{T} \vdash \gamma$ together with $\mathbf{T} \vdash \neg \text{Prov}_{\mathbf{T}}(\ulcorner \gamma \urcorner) \leftrightarrow \gamma$ gives $\mathbf{T} \vdash \neg \text{Prov}_{\mathbf{T}}(\ulcorner \gamma \urcorner)$, and so $\mathbf{T}$ would be inconsistent. Now it is natural to ask about the status of a fixed point of $\text{Prov}_{\mathbf{T}}(y)$, O.s., a **sentença** δ such that

$$\mathbf{T} \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \delta \urcorner) \leftrightarrow \delta.$$

If it were **derivável**, $\mathbf{T} \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \delta \urcorner)$ by condition (1), but the same conclusion follows if we apply modus ponens to the equivalence above. Hence, we don't get that $\mathbf{T}$ is inconsistent, at least not by the same argument as in the case of the Gödel sentence. This of course does not show that $\mathbf{T}$ *does* **deriva** δ .

We can make headway on this question if we generalize it a bit. The left-to-right direction of the fixed point equivalence, $\text{Prov}_{\mathbf{T}}(\ulcorner \delta \urcorner) \rightarrow \delta$, is an instance of a general schema called a *reflection principle*: $\text{Prov}_{\mathbf{T}}(\ulcorner \varphi \urcorner) \rightarrow \varphi$. It is called that because it expresses, in a sense, that $\mathbf{T}$ can “reflect” about what it can **deriva**; basically it says, “If $\mathbf{T}$ can **deriva** φ , then φ is true,” for any φ . This is true for sound theories only, of course, and this suggests that theories will in general not **deriva** every instance of it. So which instances can a theory (strong enough, and satisfying the **derivabilidade** conditions) **deriva**? Certainly all those where φ itself is **derivável**. And that's it, as the next result shows.

inc:inp:lob: thm:lob **Teorema 37.10.** *Let $\mathbf{T}$ be an **axiomatizável** theory extending $\mathbf{Q}$, and suppose $\text{Prov}_{\mathbf{T}}(y)$ is a formula satisfying conditions P1–P3 from **Seção 37.7**. If $\mathbf{T}$ **deriva** $\text{Prov}_{\mathbf{T}}(\ulcorner \varphi \urcorner) \rightarrow \varphi$, then in fact $\mathbf{T}$ **deriva** φ .*

Put differently, if $\mathbf{T} \not\vdash \varphi$, then $\mathbf{T} \not\vdash \text{Prov}_{\mathbf{T}}(\ulcorner \varphi \urcorner) \rightarrow \varphi$. This result is known as Löb's theorem.

The heuristic for the proof of Löb's theorem is a clever proof that Santa Claus exists. (If you don't like that conclusion, you are free to substitute any other conclusion you would like.) Here it is: explanation

CAPÍTULO 37. INCOMPLETENESS AND PROVABILITY

1. Let X be the sentence, “If X is true, then Santa Claus exists.”
2. Suppose X is true.
3. Then what it says holds; O.s., we have: if X is true, then Santa Claus exists.
4. Since we are assuming X is true, we can conclude that Santa Claus exists, by modus ponens from (2) and (3).
5. We have succeeded in deriving (4), “Santa Claus exists,” from the assumption (2), “ X is true.” By conditional proof, we have shown: “If X is true, then Santa Claus exists.”
6. But this is just the sentence X . So we have shown that X is true.
7. But then, by the argument (2)–(4) above, Santa Claus exists.

A formalization of this idea, replacing “is true” with “is derivável,” and “Santa Claus exists” with φ , yields the proof of Löb’s theorem. The trick is to apply the fixed-point lemma to the fórmula $\text{Prov}_{\mathbf{T}}(y) \rightarrow \varphi$. The fixed point of that corresponds to the sentence X in the preceding sketch.

Prova de Teorema 37.10. Suppose φ is a sentença such that $\mathbf{T}$ deriva $\text{Prov}_{\mathbf{T}}(\ulcorner \varphi \urcorner) \rightarrow \varphi$. Let $\psi(y)$ be the fórmula $\text{Prov}_{\mathbf{T}}(y) \rightarrow \varphi$, and use the fixed-point lemma to find a sentença θ such that $\mathbf{T}$ deriva $\theta \leftrightarrow \psi(\ulcorner \theta \urcorner)$. Then each of the following is

37.8. LÖB'S THEOREM

derivável in $\mathbf{T}$:

$$\text{inc:inp:lob:} \quad \theta \leftrightarrow (\text{Prov}_{\mathbf{T}}(\ulcorner \theta \urcorner) \rightarrow \varphi) \quad (37.14)$$

L-1

θ is a fixed point of $\psi(y)$

$$\text{inc:inp:lob:} \quad \theta \rightarrow (\text{Prov}_{\mathbf{T}}(\ulcorner \theta \urcorner) \rightarrow \varphi) \quad (37.15)$$

L-2

from Eq. (37.14)

$$\text{inc:inp:lob:} \quad \text{Prov}_{\mathbf{T}}(\ulcorner \theta \rightarrow (\text{Prov}_{\mathbf{T}}(\ulcorner \theta \urcorner) \rightarrow \varphi) \urcorner) \quad (37.16)$$

L-3

from Eq. (37.15) by condition P1

$$\text{inc:inp:lob:} \quad \text{Prov}_{\mathbf{T}}(\ulcorner \theta \urcorner) \rightarrow \text{Prov}_{\mathbf{T}}(\ulcorner \text{Prov}_{\mathbf{T}}(\ulcorner \theta \urcorner) \rightarrow \varphi \urcorner) \quad (37.17)$$

L-4

from Eq. (37.16) using condition P2

$$\text{inc:inp:lob:} \quad \text{Prov}_{\mathbf{T}}(\ulcorner \theta \urcorner) \rightarrow (\text{Prov}_{\mathbf{T}}(\ulcorner \text{Prov}_{\mathbf{T}}(\ulcorner \theta \urcorner) \urcorner) \rightarrow \text{Prov}_{\mathbf{T}}(\ulcorner \varphi \urcorner)) \quad (37.18)$$

L-5

from Eq. (37.17) using P2 again

$$\text{inc:inp:lob:} \quad \text{Prov}_{\mathbf{T}}(\ulcorner \theta \urcorner) \rightarrow \text{Prov}_{\mathbf{T}}(\ulcorner \text{Prov}_{\mathbf{T}}(\ulcorner \theta \urcorner) \urcorner) \quad (37.19)$$

L-6

by derivabilidade condition P3

$$\text{inc:inp:lob:} \quad \text{Prov}_{\mathbf{T}}(\ulcorner \theta \urcorner) \rightarrow \text{Prov}_{\mathbf{T}}(\ulcorner \varphi \urcorner) \quad (37.20)$$

L-7

from Eq. (37.18) and Eq. (37.19)

$$\text{inc:inp:lob:} \quad \text{Prov}_{\mathbf{T}}(\ulcorner \varphi \urcorner) \rightarrow \varphi \quad (37.21)$$

L-8

by assumption of the theorem

$$\text{inc:inp:lob:} \quad \text{Prov}_{\mathbf{T}}(\ulcorner \theta \urcorner) \rightarrow \varphi \quad (37.22)$$

L-9

from Eq. (37.20) and Eq. (37.21)

$$\text{inc:inp:lob:} \quad (\text{Prov}_{\mathbf{T}}(\ulcorner \theta \urcorner) \rightarrow \varphi) \rightarrow \theta \quad (37.23)$$

L-10

from Eq. (37.14)

$$\text{inc:inp:lob:} \quad \theta \quad (37.24)$$

L-11

from Eq. (37.22) and Eq. (37.23)

$$\text{inc:inp:lob:} \quad \text{Prov}_{\mathbf{T}}(\ulcorner \theta \urcorner) \quad (37.25)$$

L-12

from Eq. (37.24) by condition P1

φ from Eq. (37.21) and Eq. (37.25) $\square$

With Löb's theorem in hand, there is a short proof of the second incompleteness theorem (for theories having a derivabilidade predicate satisfying conditions P1–P3): if $\mathbf{T} \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \perp \urcorner) \rightarrow \perp$, then $\mathbf{T} \vdash \perp$. If $\mathbf{T}$ is consistent, $\mathbf{T} \not\vdash \perp$. So, $\mathbf{T} \not\vdash \text{Prov}_{\mathbf{T}}(\ulcorner \perp \urcorner) \rightarrow \perp$, O.s., $\mathbf{T} \not\vdash \text{Con}_{\mathbf{T}}$. We can also apply it to show that δ , the fixed point of $\text{Prov}_{\mathbf{T}}(x)$, is derivável. For since

$$\mathbf{T} \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \delta \urcorner) \leftrightarrow \delta$$

in particular

$$\mathbf{T} \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \delta \urcorner) \rightarrow \delta$$

and so by Löb's theorem, $\mathbf{T} \vdash \delta$.

Problema 37.5. Let $\mathbf{T}$ be a computably axiomatized theory, and let $\text{Prov}_{\mathbf{T}}$ be a *derivabilidade* predicate for $\mathbf{T}$. Consider the following four statements:

1. If $T \vdash \varphi$, then $T \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \varphi \urcorner)$.
2. $T \vdash \varphi \rightarrow \text{Prov}_{\mathbf{T}}(\ulcorner \varphi \urcorner)$.
3. If $T \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \varphi \urcorner)$, then $T \vdash \varphi$.
4. $T \vdash \text{Prov}_{\mathbf{T}}(\ulcorner \varphi \urcorner) \rightarrow \varphi$

Under what conditions are each of these statements true?

-NoValue-

`content/incompleteness/incompleteness-provability/tarski-thm.tex`

37.9 The Undefinability of Truth

The notion of *definability* depends on having a formal semantics for the language of arithmetic. We have described a set of formulas and sentences in the language of arithmetic. The “intended interpretation” is to read such sentences as making assertions about the natural numbers, and such an assertion can be true or false. Let $\mathfrak{N}$ be the *estrutura* with domain $\mathbb{N}$ and the standard interpretation for the symbols in the language of arithmetic. Then $\mathfrak{N} \models \varphi$ means “ φ is true in the standard interpretation.”

`inc:inp:tar:
sec`

Definição 37.11. A relation $R(x_1, \dots, x_k)$ of natural numbers is *definable* in $\mathfrak{N}$ if and only if there is a formula $\varphi(x_1, \dots, x_k)$ in the language of arithmetic such that for every $n_1, \dots, n_k$, $R(n_1, \dots, n_k)$ if and only if $\mathfrak{N} \models \varphi(\bar{n}_1, \dots, \bar{n}_k)$.

Put differently, a relation is definable in $\mathfrak{N}$ if and only if it is representable in the theory $\mathbf{TA}$, where $\mathbf{TA} = \{\varphi : \mathfrak{N} \models \varphi\}$ is the set of true sentences of arithmetic. (If this is not immediately clear to you, you should go back and check the definitions and convince yourself that this is the case.)

Lema 37.12. *Every computable relation is definable in $\mathfrak{N}$.*

Prova. It is easy to check that the formula representing a relation in $\mathbf{Q}$ defines the same relation in $\mathfrak{N}$. □

Now one can ask, is the converse also true? That is, is every relation definable in $\mathfrak{N}$ computable? The answer is no. For example:

Lema 37.13. *The halting relation is definable in $\mathfrak{N}$.*

37.9. THE UNDEFINABILITY OF TRUTH

Prova. Recall that the Kleene normal form theorem states that every partial computable function f has an index e such that $f(x) = U(\mu s T(e, x, s))$ for all $x \in \mathbb{N}$, where U and T are primitive recursive and therefore total. Thus, $f(x)$ is defined (O.s., the computation halts) iff there is an s such that $T(e, x, s)$ holds.

Now let H be the halting relation, O.s.,

$$H = \{\langle e, x \rangle : \exists s T(e, x, s)\}.$$

Let θ_T define T in $\mathfrak{N}$. Then

$$H = \{\langle e, x \rangle : \mathfrak{N} \models \exists s \theta_T(\bar{e}, \bar{x}, s)\},$$

so $\exists s \theta_T(z, x, s)$ defines H in $\mathfrak{N}$. □

Problema 37.6. Show that $Q(n) \Leftrightarrow n \in \{\# \varphi^\# : \mathbf{Q} \vdash \varphi\}$ is definable in arithmetic.

What about **TA** itself? Is it definable in arithmetic? That is: is the set $\{\# \varphi^\# : \mathfrak{N} \models \varphi\}$ definable in arithmetic? Tarski's theorem answers this in the negative.

inc:inp:tar: thm:tarski **Teorema 37.14.** *The set of true sentenças of arithmetic is not definable in arithmetic.*

Prova. Suppose $\theta(x)$ defined it, O.s., $\mathfrak{N} \models \varphi$ iff $\mathfrak{N} \models \theta(\ulcorner \varphi \urcorner)$. By the fixed-point lemma, there is a formula φ such that $\mathbf{Q} \vdash \varphi \leftrightarrow \neg \theta(\ulcorner \varphi \urcorner)$, and hence $\mathfrak{N} \models \varphi \leftrightarrow \neg \theta(\ulcorner \varphi \urcorner)$. But then $\mathfrak{N} \models \varphi$ if and only if $\mathfrak{N} \models \neg \theta(\ulcorner \varphi \urcorner)$, which contradicts the fact that $\theta(y)$ is supposed to define the set of true statements of arithmetic. □

Tarski applied this analysis to a more general philosophical notion of truth. Given any language L , Tarski argued that an adequate notion of truth for L would have to satisfy, for each sentence X ,

‘ X ’ is true if and only if X .

Tarski's oft-quoted example, for English, is the sentence

‘Snow is white’ is true if and only if snow is white.

However, for any language strong enough to represent the diagonal function, and any linguistic predicate $T(x)$, we can construct a sentence X satisfying “ X if and only if not $T(\ulcorner X \urcorner)$.” Given that we do not want a truth predicate to declare some sentences to be both true and false, Tarski concluded that one cannot specify a truth predicate for all sentences in a language without, somehow, stepping outside the bounds of the language. In other words, a the truth predicate for a language cannot be defined in the language itself.

Parte VIII

Second-order Logic

This is the beginnings of a part on second-order logic.

Capítulo 38

Syntax and Semantics

Basic syntax and semantics for SOL covered so far. As a chapter it's too short. Substitution for second-order variables has to be covered to be able to talk about *derivação* systems for SOL, and there's some subtle issues there.

-NoValue-

`content/second-order-logic/syntax-and-semantics/introduction.tex`

38.1 Introduction

In first-order logic, we combine the non-logical symbols of a given language, O.s., its *símbolos de constante*, *símbolos de função*, and *símbolos de predicado*, with the logical symbols to express things about first-order *estruturas*. This is done using the notion of satisfaction, which relates a *estrutura* $\mathfrak{M}$, together with a variable assignment s , and a *fórmula* φ : $\mathfrak{M}, s \models \varphi$ holds iff what φ expresses when its *símbolos de constante*, *símbolos de função*, and *símbolos de predicado* are interpreted as $\mathfrak{M}$ says, and its free variables are interpreted as s says, is true. The interpretation of the *predicado de identidade* $=$ is built into the definition of $\mathfrak{M}, s \models \varphi$, as is the interpretation of $\forall$ and $\exists$. The former is always interpreted as the identity relation on the *domínio* $|\mathfrak{M}|$ of the structure,

38.2. TERMS AND FÓRMULAS

and the quantifiers are always interpreted as ranging over the entire **domínio**. But, crucially, quantification is only allowed over elements of the **domínio**, and so only object **variáveis** are allowed to follow a quantifier.

In second-order logic, both the language and the definition of satisfaction are extended to include free and bound function and predicate variables, and quantification over them. These variables are related to **símbolos de função** and **símbolos de predicado** the same way that object variables are related to **símbolos de constante**. They play the same role in the formation of terms and **fórmulas** of second-order logic, and quantification over them is handled in a similar way. In the *standard* semantics, the second-order quantifiers range over all possible objects of the right type (n -place functions from $|\mathfrak{M}|$ to $|\mathfrak{M}|$ for function variables, n -place relations for predicate variables). For instance, while $\forall v_0 (P_0^1(v_0) \vee \neg P_0^1(v_0))$ is a formula in both first- and second-order logic, in the latter we can also consider $\forall V_0^1 \forall v_0 (V_0^1(v_0) \vee \neg V_0^1(v_0))$ and $\exists V_0^1 \forall v_0 (V_0^1(v_0) \vee \neg V_0^1(v_0))$. Since these contain no free variables, they are **sentenças** of second-order logic. Here, V_0^1 is a second-order 1-place predicate variable. The allowable interpretations of V_0^1 are the same that we can assign to a 1-place **símbolo de predicado** like P_0^1 , O.s., subsets of $|\mathfrak{M}|$. Quantification over them then amounts to saying that $\forall v_0 (V_0^1(v_0) \vee \neg V_0^1(v_0))$ holds for all ways of assigning a subset of $|\mathfrak{M}|$ as the value of V_0^1 , or for at least one. Since every set either contains or fails to contain a given object, both are true in any **estrutura**.

-NoValue-

`content/second-order-logic/syntax-and-semantics/terms-formulas.tex`

38.2 Terms and Fórmulas

`sol:syn:frm:`
`sec`

Like in first-order logic, expressions of second-order logic are built up from a basic vocabulary containing **variáveis**, **símbolos de constante**, **símbolos de predicado** and sometimes **símbolos de função**. From them, together with logical connectives, quantifiers, and punctuation symbols such as parentheses and commas, **terms** and **fórmulas** are formed. The difference is that in addition to variables for objects, second-order logic also contains variables for relations and functions, and allows quantification over them. So the logical symbols of second-order logic are those of first-order logic, plus:

1. A **infinitos enumeráveis** set of second-order relation **variáveis** of every arity n : $V_0^n, V_1^n, V_2^n, \dots$
2. A **infinitos enumeráveis** set of second-order function **variáveis**: $u_0^n, u_1^n, u_2^n, \dots$

Just as we use x, y, z as meta-variables for first-order variables v_i , we'll use X, Y, Z , etc., as metavariables for V_i^n and u, v , etc., as meta-variables for u_i^n .

`explanation`

The non-logical symbols of a second-order language are specified the same way a first-order language is: by listing its **símbolos de constante**, **símbolos de função**, and **símbolos de predicado**.

In first-order logic, the **predicado de identidade** $=$ is usually included. In first-order logic, the non-logical symbols of a language $\mathcal{L}$ are crucial to allow us to express anything interesting. There are of course **sentenças** that use no non-logical symbols, but with only $=$ it is hard to say anything interesting. In second-order logic, since we have an unlimited supply of relation and function variables, we can say anything we can say in a first-order language even without a special supply of non-logical symbols.

Definição 38.1 (Second-order Terms). The set of *second-order terms* of $\mathcal{L}$, $\text{Trm}^2(\mathcal{L})$, is defined by adding to **Definição 15.4** the clause

1. If u is an n -place function variable and $t_1, \dots, t_n$ are terms, then $u(t_1, \dots, t_n)$ is a term.

explanation

So, a second-order term looks just like a first-order term, except that where a first-order term contains a **símbolo de função** f_i^n , a second-order term may contain a function variable u_i^n in its place.

Definição 38.2 (Second-order fórmula). The set of *second-order fórmulas* $\text{Frm}^2(\mathcal{L})$ of the language $\mathcal{L}$ is defined by adding to **Definição 15.5** the clauses

1. If X is an n -place predicate variable and $t_1, \dots, t_n$ are second-order terms of $\mathcal{L}$, then $X(t_1, \dots, t_n)$ is an atomic **fórmula**.
2. If φ is a **fórmula** and u is a function variable, then $\forall u \varphi$ is a **fórmula**.
3. If φ is a **fórmula** and X is a predicate variable, then $\forall X \varphi$ is a **fórmula**.
4. If φ is a **fórmula** and u is a function variable, then $\exists u \varphi$ is a **fórmula**.
5. If φ is a **fórmula** and X is a predicate variable, then $\exists X \varphi$ is a **fórmula**.

-NoValue-

`content/second-order-logic/syntax-and-semantics/satisfaction.tex`

38.3 Satisfaction

explanation

To define the satisfaction relation $\mathfrak{M}, s \models \varphi$ for second-order **fórmulas**, we have to extend the definitions to cover second-order **variáveis**. The notion of a **estrutura** is the same for second-order logic as it is for first-order logic. There is only a difference for variable assignments s : these now must not just provide values for the first-order **variáveis**, but also for the second-order **variáveis**.

sol:syn:sat:
sec

Definição 38.3 (Variable Assignment). A *variable assignment* s for a **estrutura** $\mathfrak{M}$ is a function which maps each

38.3. SATISFACTION

1. object **variável** v_i to an element of $|\mathfrak{M}|$, O.s., $s(v_i) \in |\mathfrak{M}|$
2. n -place relation variable V_i^n to an n -place relation on $|\mathfrak{M}|$, O.s., $s(V_i^n) \subseteq |\mathfrak{M}|^n$;
3. n -place function variable u_i^n to an n -place function from $|\mathfrak{M}|$ to $|\mathfrak{M}|$, O.s., $s(u_i^n): |\mathfrak{M}|^n \rightarrow |\mathfrak{M}|$;

A **estrutura** assigns a **valor** to each **símbolo de constante** and **símbolo de função**, and a second-order variable assignment assigns objects and functions to each object and function variable. Together, they let us assign a value to every term. explanation

Definição 38.4 (Valor of a Term). If t is a term of the language $\mathcal{L}$, $\mathfrak{M}$ is a **estrutura** for $\mathcal{L}$, and s is a **variável** assignment for $\mathfrak{M}$, the **valor** $\text{Val}_s^{\mathfrak{M}}(t)$ is defined as for first-order terms, plus the following clause:

$$t \equiv u(t_1, \dots, t_n):$$

$$\text{Val}_s^{\mathfrak{M}}(t) = s(u)(\text{Val}_s^{\mathfrak{M}}(t_1), \dots, \text{Val}_s^{\mathfrak{M}}(t_n)).$$

Definição 38.5 (x -Variant). If s is a **variável** assignment for a **estrutura** $\mathfrak{M}$, then any **variável** assignment s' for $\mathfrak{M}$ which differs from s at most in what it assigns to x is called an x -variant of s . If s' is an x -variant of s we write $s' \sim_x s$. (Similarly for second-order variables X or u .)

Definição 38.6. If s is a **variável** assignment for a **estrutura** $\mathfrak{M}$ and $m \in |\mathfrak{M}|$, then the assignment $s[m/x]$ is the **variável** assignment defined by

$$s[m/y] = \begin{cases} m & \text{if } y \equiv x \\ s(y) & \text{otherwise,} \end{cases}$$

If X is an n -place relation **variável** and $M \subseteq |\mathfrak{M}|^n$, then $s[M/X]$ is the **variável** assignment defined by

$$s[M/y] = \begin{cases} M & \text{if } y \equiv X \\ s(y) & \text{otherwise.} \end{cases}$$

If u is an n -place function **variável** and $f: |\mathfrak{M}|^n \rightarrow |\mathfrak{M}|$, then $s[f/u]$ is the **variável** assignment defined by

$$s[f/y] = \begin{cases} f & \text{if } y \equiv u \\ s(y) & \text{otherwise.} \end{cases}$$

In each case, y may be any first- or second-order **variável**.

Definição 38.7 (Satisfaction). For second-order **fórmulas** φ , the definition of satisfaction is like **Definição 16.11** with the addition of:

1. $\varphi \equiv X^n(t_1, \dots, t_n)$: $\mathfrak{M}, s \models \varphi$ iff $\langle \text{Val}_s^{\mathfrak{M}}(t_1), \dots, \text{Val}_s^{\mathfrak{M}}(t_n) \rangle \in s(X^n)$.
2. $\varphi \equiv \forall X \psi$: $\mathfrak{M}, s \models \varphi$ iff for every $M \subseteq |\mathfrak{M}|^n$, $\mathfrak{M}, s[M/X] \models \psi$.
3. $\varphi \equiv \exists X \psi$: $\mathfrak{M}, s \models \varphi$ iff for at least one $M \subseteq |\mathfrak{M}|^n$ so that $\mathfrak{M}, s[M/X] \models \psi$.
4. $\varphi \equiv \forall u \psi$: $\mathfrak{M}, s \models \varphi$ iff for every $f: |\mathfrak{M}|^n \rightarrow |\mathfrak{M}|$, $\mathfrak{M}, s[f/u] \models \psi$.
5. $\varphi \equiv \exists u \psi$: $\mathfrak{M}, s \models \varphi$ iff for at least one $f: |\mathfrak{M}|^n \rightarrow |\mathfrak{M}|$ so that $\mathfrak{M}, s[f/u] \models \psi$.

Exemplo 38.8. Consider the *fórmula* $\forall z (X(z) \leftrightarrow \neg Y(z))$. It contains no second-order quantifiers, but does contain the second-order *variáveis* X and Y (here understood to be one-place). The corresponding first-order *sentença* $\forall z (P(z) \leftrightarrow \neg R(z))$ says that whatever falls under the interpretation of P does not fall under the interpretation of R and vice versa. In a *estrutura*, the interpretation of a *símbolo de predicado* P is given by the interpretation $P^{\mathfrak{M}}$. But for second-order *variáveis* like X and Y , the interpretation is provided, not by the *estrutura* itself, but by a *variável* assignment. Since the second-order *fórmula* is not a *sentença* (it includes free *variáveis* X and Y), it is only satisfied relative to a *estrutura* $\mathfrak{M}$ together with a *variável* assignment s .

$\mathfrak{M}, s \models \forall z (X(z) \leftrightarrow \neg Y(z))$ whenever the *membros* of $s(X)$ are not *membros* of $s(Y)$, and vice versa, O.s., iff $s(Y) = |\mathfrak{M}| \setminus s(X)$. For instance, take $|\mathfrak{M}| = \{1, 2, 3\}$. Since no *símbolos de predicado*, *símbolos de função*, or *símbolos de constante* are involved, the domain of $\mathfrak{M}$ is all that is relevant. Now for $s_1(X) = \{1, 2\}$ and $s_1(Y) = \{3\}$, we have $\mathfrak{M}, s_1 \models \forall z (X(z) \leftrightarrow \neg Y(z))$.

By contrast, if we have $s_2(X) = \{1, 2\}$ and $s_2(Y) = \{2, 3\}$, $\mathfrak{M}, s_2 \not\models \forall z (X(z) \leftrightarrow \neg Y(z))$. That's because $\mathfrak{M}, s_2[2/z] \models X(z)$ (since $2 \in s_2[2/z](X)$) but $\mathfrak{M}, s_2[2/z] \not\models \neg Y(z)$ (since also $2 \in s_2[2/z](Y)$).

Exemplo 38.9. $\mathfrak{M}, s \models \exists Y (\exists y Y(y) \wedge \forall z (X(z) \leftrightarrow \neg Y(z)))$ if there is an $N \subseteq |\mathfrak{M}|$ such that $\mathfrak{M}, s[N/Y] \models (\exists y Y(y) \wedge \forall z (X(z) \leftrightarrow \neg Y(z)))$. And that is the case for any $N \neq \emptyset$ (so that $\mathfrak{M}, s[N/Y] \models \exists y Y(y)$) and, as in the previous example, $N = |\mathfrak{M}| \setminus s(X)$. In other words, $\mathfrak{M}, s \models \exists Y (\exists y Y(y) \wedge \forall z (X(z) \leftrightarrow \neg Y(z)))$ iff $|\mathfrak{M}| \setminus s(X)$ is non-empty, O.s., $s(X) \neq |\mathfrak{M}|$. So, the *fórmula* is satisfied, p.ex., if $|\mathfrak{M}| = \{1, 2, 3\}$ and $s(X) = \{1, 2\}$, but not if $s(X) = \{1, 2, 3\} = |\mathfrak{M}|$.

Since the *fórmula* is not satisfied whenever $s(X) = |\mathfrak{M}|$, the *sentença*

$$\forall X \exists Y (\exists y Y(y) \wedge \forall z (X(z) \leftrightarrow \neg Y(z)))$$

is never satisfied: For any *estrutura* $\mathfrak{M}$, the assignment $s(X) = |\mathfrak{M}|$ will make the *sentença* false. On the other hand, the sentence

$$\exists X \exists Y (\exists y Y(y) \wedge \forall z (X(z) \leftrightarrow \neg Y(z)))$$

is satisfied relative to any assignment s , since we can always find $M \subseteq |\mathfrak{M}|$ but $M \neq |\mathfrak{M}|$ (p.ex., $M = \emptyset$).

38.4. SEMANTIC NOTIONS

Exemplo 38.10. The second-order *sentença* $\forall X \forall y X(y)$ says that every 1-place relation, O.s., every property, holds of every object. That is clearly never true, since in every $\mathfrak{M}$, for a variable assignment s with $s(X) = \emptyset$, and $s(y) = a \in |\mathfrak{M}|$ we have $\mathfrak{M}, s \not\models X(y)$. This means that $\varphi \rightarrow \forall X \forall y X(y)$ is equivalent in second-order logic to $\neg\varphi$, that is: $\mathfrak{M} \models \varphi \rightarrow \forall X \forall y X(y)$ iff $\mathfrak{M} \models \neg\varphi$. In other words, in second-order logic we can define $\neg$ using $\forall$ and $\rightarrow$.

Problema 38.1. Show that in second-order logic $\forall$ and $\rightarrow$ can define the other connectives:

1. Prove that in second-order logic $\varphi \wedge \psi$ is equivalent to $\forall X (\varphi \rightarrow (\psi \rightarrow \forall x X(x)) \rightarrow \forall x X(x))$.
2. Find a second-order formula using only $\forall$ and $\rightarrow$ equivalent to $\varphi \vee \psi$.

-NoValue- _____
<content/second-order-logic/syntax-and-semantics/semantic-notions.tex>

38.4 Semantic Notions

sol:syn:sem: sec The central logical notions of *validity*, *entailment*, and *satisfiability* are defined explanation the same way for second-order logic as they are for first-order logic, except that the underlying satisfaction relation is now that for second-order *fórmulas*. A second-order *sentença*, of course, is a *fórmula* in which all variables, including predicate and function variables, are bound.

Definição 38.11 (Validity). A sentence φ is *valid*, $\models \varphi$, iff $\mathfrak{M} \models \varphi$ for every *estrutura* $\mathfrak{M}$.

Definição 38.12 (Entailment). A set of sentences Γ *entails* a sentence φ , $\Gamma \models \varphi$, iff for every *estrutura* $\mathfrak{M}$ with $\mathfrak{M} \models \Gamma$, $\mathfrak{M} \models \varphi$.

Definição 38.13 (Satisfiability). A set of sentences Γ is *satisfiable* if $\mathfrak{M} \models \Gamma$ for some *estrutura* $\mathfrak{M}$. If Γ is not satisfiable it is called *unsatisfiable*.

-NoValue- _____
<content/second-order-logic/syntax-and-semantics/expressive-power.tex>

38.5 Expressive Power

sol:syn:exp: sec Quantification over second-order variables is responsible for an immense increase in the expressive power of the language over that of first-order logic. explanation Second-order existential quantification lets us say that functions or relations with certain properties exist. In first-order logic, the only way to do that is to specify a non-logical symbol (O.s., a *símbolo de função* or *símbolo de predicado*) for this purpose. Second-order universal quantification lets us say that all subsets of, relations on, or functions from the *domínio* to the *domínio* have

a property. In first-order logic, we can only say that the subsets, relations, or functions assigned to one of the non-logical symbols of the language have a property. And when we say that subsets, relations, functions exist that have a property, or that all of them have it, we can use second-order quantification in specifying this property as well. This lets us define relations not definable in first-order logic, and express properties of the domain not expressible in first-order logic.

Definição 38.14. If $\mathfrak{M}$ is a **estrutura** for a language $\mathcal{L}$, a relation $R \subseteq |\mathfrak{M}|^2$ is *definable* in $\mathcal{L}$ if there is some **fórmula** $\varphi_R(x, y)$ with only the variables x and y free, such that $R(a, b)$ holds (O.s., $\langle a, b \rangle \in R$) iff $\mathfrak{M}, s \models \varphi_R(x, y)$ for $s(x) = a$ and $s(y) = b$.

Exemplo 38.15. In first-order logic we can define the identity relation $\text{Id}_{|\mathfrak{M}|}$ (O.s., $\{\langle a, a \rangle : a \in |\mathfrak{M}|\}$) by the formula $x = y$. In second-order logic, we can define this relation *without* $=$. For if a and b are the same **membro** of $|\mathfrak{M}|$, then they are **membros** of the same subsets of $|\mathfrak{M}|$ (since sets are determined by their **membros**). Conversely, if a and b are different, then they are not **membros** of the same subsets: p.ex., $a \in \{a\}$ but $b \notin \{a\}$ if $a \neq b$. So “being **membros** of the same subsets of $|\mathfrak{M}|$ ” is a relation that holds of a and b iff $a = b$. It is a relation that can be expressed in second-order logic, since we can quantify over all subsets of $|\mathfrak{M}|$. Hence, the following **fórmula** defines $\text{Id}_{|\mathfrak{M}|}$:

$$\forall X (X(x) \leftrightarrow X(y))$$

Problema 38.2. Show that $\forall X (X(x) \rightarrow X(y))$ (note: $\rightarrow$ not $\leftrightarrow$!) defines $\text{Id}_{|\mathfrak{M}|}$.

Exemplo 38.16. If R is a two-place **símbolo de predicado**, $R^{\mathfrak{M}}$ is a two-place relation on $|\mathfrak{M}|$. Perhaps somewhat confusingly, we’ll use R as the **símbolo de predicado** for R and for the relation $R^{\mathfrak{M}}$ itself. The *transitive closure* R^* of R is the relation that holds between a and b iff for some $c_1, \dots, c_k$, $R(a, c_1)$, $R(c_1, c_2), \dots, R(c_k, b)$ holds. This includes the case if $k = 0$, O.s., if $R(a, b)$ holds, so does $R^*(a, b)$. This means that $R \subseteq R^*$. In fact, R^* is the smallest relation that includes R and that is transitive. We can say in second-order logic that X is a transitive relation that includes R :

$$\begin{aligned} \psi_R(X) \equiv & \forall x \forall y (R(x, y) \rightarrow X(x, y)) \wedge \\ & \forall x \forall y \forall z ((X(x, y) \wedge X(y, z)) \rightarrow X(x, z)). \end{aligned}$$

The first conjunct says that $R \subseteq X$ and the second that X is transitive.

To say that X is the smallest such relation is to say that it is itself included in every relation that includes R and is transitive. So we can define the transitive closure of R by the **fórmula**

$$R^*(X) \equiv \psi_R(X) \wedge \forall Y (\psi_R(Y) \rightarrow \forall x \forall y (X(x, y) \rightarrow Y(x, y))).$$

We have $\mathfrak{M}, s \models R^*(X)$ iff $s(X) = R^*$. The transitive closure of R cannot be expressed in first-order logic.

-NoValue-

content/second-order-logic/syntax-and-semantics/inf-count.tex

38.6 Describing Infinite and Enumerável Domínios

sol:syn:siz:
sec

A set M is (Dedekind) infinite iff there is uma *injetiva* function $f: M \rightarrow M$ which is not *sobrejetiva*, O.s., with $\text{Im}(f) \neq M$. In first-order logic, we can consider a one-place *símbolo de função* f and say that the function $f^{\mathfrak{M}}$ assigned to it in a *estrutura* $\mathfrak{M}$ is *injetiva* and $\text{Im}(f) \neq |\mathfrak{M}|$:

$$\forall x \forall y (f(x) = f(y) \rightarrow x = y) \wedge \exists y \forall x y \neq f(x).$$

If $\mathfrak{M}$ satisfies this *sentença*, $f^{\mathfrak{M}}: |\mathfrak{M}| \rightarrow |\mathfrak{M}|$ is *injetiva*, and so $|\mathfrak{M}|$ must be infinite. If $|\mathfrak{M}|$ is infinite, and hence such a function exists, we can let $f^{\mathfrak{M}}$ be that function and $\mathfrak{M}$ will satisfy the *sentença*. However, this requires that our language contains the non-logical symbol f which we use for this purpose. In second-order logic, we can simply say that such a function *exists*. This no-longer requires f , and we obtain the *sentença* in pure second-order logic

$$\text{Inf} \equiv \exists u (\forall x \forall y (u(x) = u(y) \rightarrow x = y) \wedge \exists y \forall x y \neq u(x)).$$

$\mathfrak{M} \models \text{Inf}$ iff $|\mathfrak{M}|$ is infinite. We can then define $\text{Fin} \equiv \neg \text{Inf}$; $\mathfrak{M} \models \text{Fin}$ iff $|\mathfrak{M}|$ is finite. No single *sentença* of pure first-order logic can express that the *domínio* is infinite although an infinite set of them can. There is no set of *sentenças* of pure first-order logic that is satisfied in a *estrutura* iff its domain is finite.

Proposição 38.17. $\mathfrak{M} \models \text{Inf}$ iff $|\mathfrak{M}|$ is infinite.

Prova. $\mathfrak{M} \models \text{Inf}$ iff $\mathfrak{M}, s \models \forall x \forall y (u(x) = u(y) \rightarrow x = y) \wedge \exists y \forall x y \neq u(x)$ for some s . If it does, $s(u)$ is uma *injetiva* function, and some $y \in |\mathfrak{M}|$ is not in the range of $s(u)$. Conversely, if there is uma *injetiva* $f: |\mathfrak{M}| \rightarrow |\mathfrak{M}|$ with $\text{Im}(f) \neq |\mathfrak{M}|$, then $s(u) = f$ is such a variable assignment. $\square$

A set M is *enumerável* if there is an enumeration

$$m_0, m_1, m_2, \dots$$

of its *membros* (without repetitions but possibly finite). Such an enumeration exists iff there is um *membro* $z \in M$ and a function $f: M \rightarrow M$ such that $z, f(z), f(f(z)), \dots$, are all the *membros* of M . For if the enumeration exists, $z = m_0$ and $f(m_k) = m_{k+1}$ (or $f(m_k) = m_k$ if m_k is the last *membro* of the enumeration) are the requisite *membro* and function. On the other hand, if such a z and f exist, then $z, f(z), f(f(z)), \dots$, is an enumeration of M , and M is *enumerável*. We can express the existence of z and f in second-order logic to produce a *sentença* true in a *estrutura* iff the *estrutura* is *enumerável*:

$$\text{Count} \equiv \exists z \exists u \forall X ((X(z) \wedge \forall x (X(x) \rightarrow X(u(x)))) \rightarrow \forall x X(x))$$

Proposição 38.18. $\mathfrak{M} \models \text{Count}$ iff $|\mathfrak{M}|$ is *enumerável*.

Prova. Suppose $|\mathfrak{M}|$ is *enumerável*, and let $m_0, m_1, \dots$, be an enumeration. By removing repetitions we can guarantee that no m_k appears twice. Define $f(m_k) = m_{k+1}$ and let $s(z) = m_0$ and $s(u) = f$. We show that

$$\mathfrak{M}, s \models \forall X ((X(z) \wedge \forall x (X(x) \rightarrow X(u(x)))) \rightarrow \forall x X(x))$$

Suppose $M \subseteq |\mathfrak{M}|$ is arbitrary. Suppose further that $\mathfrak{M}, s[M/X] \models (X(z) \wedge \forall x (X(x) \rightarrow X(u(x))))$. Then $s[M/X](z) \in M$ and whenever $x \in M$, also $(s[M/X](u))(x) \in M$. In other words, since $s[M/X] \sim_X s$, $m_0 \in M$ and if $x \in M$ then $f(x) \in M$, so $m_0 \in M$, $m_1 = f(m_0) \in M$, $m_2 = f(f(m_0)) \in M$, etc. Thus, $M = |\mathfrak{M}|$, and so $\mathfrak{M}, s[M/X] \models \forall x X(x)$. Since $M \subseteq |\mathfrak{M}|$ was arbitrary, we are done: $\mathfrak{M} \models \text{Count}$.

Now assume that $\mathfrak{M} \models \text{Count}$, O.s.,

$$\mathfrak{M}, s \models \forall X ((X(z) \wedge \forall x (X(x) \rightarrow X(u(x)))) \rightarrow \forall x X(x))$$

for some s . Let $m = s(z)$ and $f = s(u)$ and consider $M = \{m, f(m), f(f(m)), \dots\}$. M so defined is clearly *enumerável*. Then

$$\mathfrak{M}, s[M/X] \models (X(z) \wedge \forall x (X(x) \rightarrow X(u(x)))) \rightarrow \forall x X(x)$$

by assumption. Also, $\mathfrak{M}, s[M/X] \models X(z)$ since $M \ni m = s[M/X](z)$, and also $\mathfrak{M}, s[M/X] \models \forall x (X(x) \rightarrow X(u(x)))$ since whenever $x \in M$ also $f(x) \in M$. So, since both antecedent and conditional are satisfied, the consequent must also be: $\mathfrak{M}, s[M/X] \models \forall x X(x)$. But that means that $M = |\mathfrak{M}|$, and so $|\mathfrak{M}|$ is *enumerável* since M is, by definition. $\square$

Problema 38.3. The *sentença* $\text{Inf} \wedge \text{Count}$ is true in all and only *infinito enumerável* domains. Adjust the definition of *Count* so that it becomes a different *sentença* that directly expresses that the domain is *infinito enumerável*, and prove that it does.

Capítulo 39

Metatheory of Second-order Logic

-NoValue-

39.1 Introduction

sol:met:int:
sec

First-order logic has a number of nice properties. We know it is not decidable, but at least it is axiomatizable. That is, there are proof systems for first-order logic which are sound and complete, O.s., they give rise to a *derivabilidade* relation $\vdash$ with the property that for any set of *sentenças* Γ and *sentença* φ , $\Gamma \models \varphi$ iff $\Gamma \vdash \varphi$. This means in particular that the validities of first-order logic are *computavelmente enumerável*. There is a computable function $f: \mathbb{N} \rightarrow \text{Sent}(\mathcal{L})$ such that the values of f are all and only the valid *sentenças* of $\mathcal{L}$. This is so because *derivações* can be enumerated, and those that *deriva* a single *sentença* are then mapped to that *sentença*. Second-order logic is more expressive than first-order logic, and so it is in general more complicated to capture its validities. In fact, we'll show that second-order logic is not only undecidable, but its validities are not even *computavelmente enumerável*. This means there can be no sound and complete proof system for second-order logic (although sound, but incomplete proof systems are available and in fact are important objects of research).

First-order logic also has two more properties: it is compact (if every finite subset of a set Γ of *sentenças* is satisfiable, Γ itself is satisfiable) and the Löwenheim–Skolem Theorem holds for it (if Γ has an infinite model it has a *infinito enumerável* model). Both of these results fail for second-order logic. Again, the reason is that second-order logic can express facts about the size of *domínios* that first-order logic cannot.

-NoValue-

39.2 Second-order Arithmetic

sol:met:spa:
sec

Recall that the theory **PA** of Peano arithmetic includes the eight axioms of **Q**,

$$\begin{aligned} &\forall x \, x' \neq 0 \\ &\forall x \, \forall y \, (x' = y' \rightarrow x = y) \\ &\forall x \, (x = 0 \vee \exists y \, x = y') \\ &\forall x \, (x + 0) = x \\ &\forall x \, \forall y \, (x + y') = (x + y)' \\ &\forall x \, (x \times 0) = 0 \\ &\forall x \, \forall y \, (x \times y') = ((x \times y) + x) \\ &\forall x \, \forall y \, (x < y \leftrightarrow \exists z \, (z' + x) = y) \end{aligned}$$

plus all sentences of the form

$$(\varphi(o) \wedge \forall x (\varphi(x) \rightarrow \varphi(x'))) \rightarrow \forall x \varphi(x).$$

The latter is a “schema,” O.s., a pattern that generates infinitely many **sentenças** of the language of arithmetic, one for each **fórmula** $\varphi(x)$. We call this schema the (first-order) *axiom schema of induction*. In *second-order* Peano arithmetic **PA²**, induction can be stated as a single sentence. **PA²** consists of the first eight axioms above plus the (second-order) *induction axiom*:

$$\forall X ((X(o) \wedge \forall x (X(x) \rightarrow X(x'))) \rightarrow \forall x X(x)).$$

It says that if a subset X of the **domínio** contains $o^{\mathfrak{M}}$ and with any $x \in |\mathfrak{M}|$ also contains $\iota^{\mathfrak{M}}(x)$ (O.s., it is “closed under successor”) it contains everything in the **domínio** (O.s., $X = |\mathfrak{M}|$).

The induction axiom guarantees that any **estrutura** satisfying it contains only those **membros** of $|\mathfrak{M}|$ the axioms require to be there, O.s., the values of $\bar{n}$ for $n \in \mathbb{N}$. A model of **PA²** contains no non-standard numbers.

Teorema 39.1. *If $\mathfrak{M} \models \mathbf{PA}^2$ then $|\mathfrak{M}| = \{\text{Val}^{\mathfrak{M}}(\bar{n}) : n \in \mathbb{N}\}$.*

*sol:met:spa:
thm:sol-pa-standard*

Prova. Let $N = \{\text{Val}^{\mathfrak{M}}(\bar{n}) : n \in \mathbb{N}\}$, and suppose $\mathfrak{M} \models \mathbf{PA}^2$. Of course, for any $n \in \mathbb{N}$, $\text{Val}^{\mathfrak{M}}(\bar{n}) \in |\mathfrak{M}|$, so $N \subseteq |\mathfrak{M}|$.

Now for inclusion in the other direction. Consider a variable assignment s with $s(X) = N$. By assumption,

$$\begin{aligned} \mathfrak{M} \models \forall X ((X(o) \wedge \forall x (X(x) \rightarrow X(x'))) \rightarrow \forall x X(x)), \text{ thus} \\ \mathfrak{M}, s \models (X(o) \wedge \forall x (X(x) \rightarrow X(x'))) \rightarrow \forall x X(x). \end{aligned}$$

Consider the antecedent of this conditional. $\text{Val}^{\mathfrak{M}}(o) \in N$, and so $\mathfrak{M}, s \models X(o)$. The second conjunct, $\forall x (X(x) \rightarrow X(x'))$ is also satisfied. For suppose $x \in N$. By definition of N , $x = \text{Val}^{\mathfrak{M}}(\bar{n})$ for some n . That gives $\iota^{\mathfrak{M}}(x) = \text{Val}^{\mathfrak{M}}(\overline{n+1}) \in N$. So, $\iota^{\mathfrak{M}}(x) \in N$.

We have that $\mathfrak{M}, s \models X(o) \wedge \forall x (X(x) \rightarrow X(x'))$. Consequently, $\mathfrak{M}, s \models \forall x X(x)$. But that means that for every $x \in |\mathfrak{M}|$ we have $x \in s(X) = N$. So, $|\mathfrak{M}| \subseteq N$. $\square$

Corolário 39.2. *Any two models of **PA²** are isomorphic.*

*sol:met:spa:
cor:sol-pa-categorical*

Prova. By **Teorema 39.1**, the domain of any model of **PA²** is exhausted by $\text{Val}^{\mathfrak{M}}(\bar{n})$. Any such model is also a model of **Q**. By **Proposição 26.3**, any such model is standard, O.s., isomorphic to $\mathfrak{N}$. $\square$

Above we defined **PA²** as the theory that contains the first eight arithmetical axioms plus the second-order induction axiom. In fact, thanks to the expressive power of second-order logic, only the *first two* of the arithmetical axioms plus induction are needed for second-order Peano arithmetic.

39.3. SECOND-ORDER LOGIC IS NOT AXIOMATIZABLE

sol:met:spa:
prop:sol-pa-definable

Proposição 39.3. Let $\mathbf{PA}^{2\ddagger}$ be the second-order theory containing the first two arithmetical axioms (the successor axioms) and the second-order induction axiom. Then $\leq$, $+$, and $\times$ are definable in $\mathbf{PA}^{2\ddagger}$.

Prova. To show that $\leq$ is definable, we have to find a formula $\varphi_{\leq}(x, y)$ such that $\mathfrak{N} \models \varphi_{\leq}(\bar{n}, \bar{m})$ iff $n \leq m$. Consider the formula

$$\psi(x, Y) \equiv Y(x) \wedge \forall y (Y(y) \rightarrow Y(y'))$$

Clearly, $\psi(\bar{n}, Y)$ is satisfied by a set $Y \subseteq \mathbb{N}$ iff $\{m : n \leq m\} \subseteq Y$, so we can take $\varphi_{\leq}(x, y) \equiv \forall Y (\psi(x, Y) \rightarrow Y(y))$.

To see that addition is definable observe that $k+l = m$ iff there is a function u such that $u(0) = k$, $u(n') = u(n)'$ for all n , and $m = u(l)$. We can use this equivalence to define addition in $\mathbf{PA}^{2\ddagger}$ by the following formula:

$$\varphi_+(x, y, z) \equiv \exists u (u(0) = x \wedge \forall w u(x') = u(x)' \wedge u(y) = z)$$

It should be clear that $\mathfrak{N} \models \varphi_+(\bar{k}, \bar{l}, \bar{m})$ iff $k + l = m$. □

Problema 39.1. Complete the proof of [Proposição 39.3](#).

-NoValue-

[content/second-order-logic/metatheory/undecidability-and-axiomatizability.tex](#)

39.3 Second-order Logic is not Axiomatizable

sol:met:nax:
sol:met:nax:sec
thm:sol-undecidable

Teorema 39.4. Second-order logic is undecidable.

Prova. A first-order [sentença](#) is valid in first-order logic iff it is valid in second-order logic, and first-order logic is undecidable. □

sol:met:nax:
cor:sol-not-axiomatizable

Teorema 39.5. There is no sound and complete [derivação](#) system for second-order logic.

Prova. Let φ be a [sentença](#) in the language of arithmetic. $\mathfrak{N} \models \varphi$ iff $\mathbf{PA}^2 \models \varphi$. Let P be the conjunction of the nine axioms of $\mathbf{PA}^2$. $\mathbf{PA}^2 \models \varphi$ iff $P \rightarrow \varphi$, O.s., $\mathfrak{M} \models P \rightarrow \varphi$. Now consider the [sentença](#) $\forall z \forall u \forall u' \forall u'' \forall L (P' \rightarrow \varphi')$ resulting by replacing 0 by z , $'$ by the one-place function variable u , $+$ and $\times$ by the two-place function-variables u' and u'' , respectively, and $<$ by the two-place relation variable L and universally quantifying. It is a valid sentence of pure second-order logic iff the original sentence was valid iff $\mathbf{PA}^2 \models \varphi$ iff $\mathfrak{N} \models \varphi$. Thus if there were a sound and complete proof system for second-order logic, we could use it to define a computable enumeration $f: \mathbb{N} \rightarrow \text{Sent}(\mathcal{L}_A)$ of the [sentenças](#) true in $\mathfrak{N}$. This function would be representable in $\mathbf{Q}$ by some first-order formula $\psi_f(x, y)$. Then the [fórmula](#) $\exists x \psi_f(x, y)$ would define the set of true first-order [sentenças](#) of $\mathfrak{N}$, contradicting Tarski's Theorem. □

-NoValue-

content/second-order-logic/metatheory/compactness.tex

39.4 Second-order Logic is not Compact

explanation Call a set of sentences Γ *finitely satisfiable* if every one of its finite subsets is satisfiable. First-order logic has the property that if a set of **sentenças** Γ is finitely satisfiable, it is satisfiable. This property is called *compactness*. It has an equivalent version involving entailment: if $\Gamma \models \varphi$, then already $\Gamma_0 \models \varphi$ for some finite subset $\Gamma_0 \subseteq \Gamma$. In this version it is an immediate corollary of the completeness theorem: for if $\Gamma \models \varphi$, by completeness $\Gamma \vdash \varphi$. But a **derivação** can only make use of finitely many **sentenças** of Γ .

sol:met:com:
sec

Compactness is not true for second-order logic. There are sets of second-order **sentenças** that are finitely satisfiable but not satisfiable, and that entail some φ without a finite subset entailing φ .

Teorema 39.6. *Second-order logic is not compact.*

sol:met:com:
thm:sol-undecidable

Prova. Recall that

$$\text{Inf} \equiv \exists u (\forall x \forall y (u(x) = u(y) \rightarrow x = y) \wedge \exists y \forall x y \neq u(x))$$

is satisfied in a **estrutura** iff its domain is infinite. Let $\varphi^{\geq n}$ be a sentence that asserts that the domain has at least n **membros**, p.ex.,

$$\varphi^{\geq n} \equiv \exists x_1 \dots \exists x_n (x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge \dots \wedge x_{n-1} \neq x_n).$$

Consider the set of **sentenças**

$$\Gamma = \{\neg \text{Inf}, \varphi^{\geq 1}, \varphi^{\geq 2}, \varphi^{\geq 3}, \dots\}.$$

It is finitely satisfiable, since for any finite subset $\Gamma_0 \subseteq \Gamma$ there is some k so that $\varphi^{\geq k} \in \Gamma$ but no $\varphi^{\geq n} \in \Gamma$ for $n > k$. If $|\mathfrak{M}|$ has k **membros**, $\mathfrak{M} \models \Gamma_0$. But, Γ is not satisfiable: if $\mathfrak{M} \models \neg \text{Inf}$, $|\mathfrak{M}|$ must be finite, say, of size k . Then $\mathfrak{M} \not\models \varphi^{\geq k+1}$. $\square$

Problema 39.2. Give an example of a set Γ and a **sentença** φ so that $\Gamma \models \varphi$ but for every finite subset $\Gamma_0 \subseteq \Gamma$, $\Gamma_0 \not\models \varphi$.

-NoValue-

content/second-order-logic/metatheory/loewenheim-skolem.tex

39.5 The Löwenheim–Skolem Theorem Fails for Second-order Logic

sol:met:lst: sec The (Downward) Löwenheim–Skolem Theorem states that every set of explanation **sentenças** with an infinite model has a **enumerável** model. It, too, is a consequence of the completeness theorem: the proof of completeness generates a model for any consistent set of **sentenças**, and that model is **enumerável**. There is also an Upward Löwenheim–Skolem Theorem, which guarantees that if a set of **sentenças** has a **infinito enumerável** model it also has a **não enumerável** model. Both theorems fail in second-order logic.

sol:met:lst: thm:sol-no-ls **Teorema 39.7.** *The Löwenheim–Skolem Theorem fails for second-order logic: There are **sentenças** with infinite models but no **enumerável** models.*

Prova. Recall that

$$\text{Count} \equiv \exists z \exists u \forall X ((X(z) \wedge \forall x (X(x) \rightarrow X(u(x)))) \rightarrow \forall x X(x))$$

is true in a **estrutura** $\mathfrak{M}$ iff $|\mathfrak{M}|$ is **enumerável**, so $\neg\text{Count}$ is true in $\mathfrak{M}$ iff $|\mathfrak{M}|$ is **não enumerável**. There are such **estruturas**—take any **não enumerável** set as the **domínio**, p.ex., $\wp(\mathbb{N})$ or $\mathbb{R}$. So $\neg\text{Count}$ has infinite models but no **enumerável** models. $\square$

Teorema 39.8. *There are **sentenças** with **infinito enumerável** but no **não enumerável** models.*

Prova. $\text{Count} \wedge \text{Inf}$ is true in $\mathbb{N}$ but not in any **estrutura** $\mathfrak{M}$ with $|\mathfrak{M}|$ **não enumerável**. $\square$

Capítulo 40

Second-order Logic and Set Theory

This section deals with coding powersets and the continuum in second-order logic. The results are stated but proofs have yet to be filled in. There are no problems yet—and the definitions and results themselves may have problems. Use with caution and report anything that’s false or unclear.

-NoValue-

[content/second-order-logic/sol-and-set-theory/introduction.tex](#)

40.1 Introduction

Since second-order logic can quantify over subsets of the domain as well as functions, it is to be expected that some amount, at least, of set theory can be carried out in second-order logic. By “carry out,” we mean that it is possible to express set theoretic properties and statements in second-order logic, and is possible without any special, non-logical vocabulary for sets (p.ex., the membership *símbolo de predicado* of set theory). For instance, we can define unions and intersections of sets and the subset relationship, but also compare the sizes of sets, and state results such as Cantor’s Theorem.

sol:set:int:
sec

-NoValue-

[content/second-order-logic/sol-and-set-theory/comparing-sets.tex](#)

40.2 Comparing Sets

Proposição 40.1. *The fórmula $\forall x (X(x) \rightarrow Y(x))$ defines the subset relation, O.s., $\mathfrak{M}, s \models \forall x (X(x) \rightarrow Y(x))$ iff $s(X) \subseteq s(Y)$.*

sol:set:cmp:
sec

Proposição 40.2. *The fórmula $\forall x (X(x) \leftrightarrow Y(x))$ defines the identity relation on sets, O.s., $\mathfrak{M}, s \models \forall x (X(x) \leftrightarrow Y(x))$ iff $s(X) = s(Y)$.*

Proposição 40.3. *The fórmula $\exists x X(x)$ defines the property of being non-empty, O.s., $\mathfrak{M}, s \models \exists x X(x)$ iff $s(X) \neq \emptyset$.*

A set X is no larger than a set Y , $X \preceq Y$, iff there is *uma injetiva* function $f: X \rightarrow Y$. Since we can express that a function is injective, and also that its values for arguments in X are in Y , we can also define the relation of being no larger than on subsets of the domain.

Proposição 40.4. *The formula*

$$\exists u (\forall x (X(x) \rightarrow Y(u(x))) \wedge \forall x \forall y (u(x) = u(y) \rightarrow x = y))$$

defines the relation of being no larger than.

40.3. CARDINALITIES OF SETS

Two sets are the same size, or “equinumerous,” $X \approx Y$, iff there is **uma bi-jetiva** function $f: X \rightarrow Y$.

Proposição 40.5. *The formula*

$$\begin{aligned} \exists u (\forall x (X(x) \rightarrow Y(u(x))) \wedge \\ \forall x \forall y (u(x) = u(y) \rightarrow x = y) \wedge \\ \forall y (Y(y) \rightarrow \exists x (X(x) \wedge y = u(x)))) \end{aligned}$$

defines the relation of being equinumerous with.

We will abbreviate these **fórmulas**, respectively, as $X \subseteq Y$, $X = Y$, $X \neq \emptyset$, $X \preceq Y$, and $X \approx Y$. (This may be slightly confusing, since we use the same notation when we speak informally about sets X and Y —but here the notation is an abbreviation for **fórmulas** in second-order logic involving one-place relation variables X and Y .)

Proposição 40.6. *The **sentença** $\forall X \forall Y ((X \preceq Y \wedge Y \preceq X) \rightarrow X \approx Y)$ is valid.*

Prova. The **sentença** is satisfied in a **estrutura** $\mathfrak{M}$ if, for any subsets $X \subseteq |\mathfrak{M}|$ and $Y \subseteq |\mathfrak{M}|$, if $X \preceq Y$ and $Y \preceq X$ then $X \approx Y$. But this holds for *any* sets X and Y —it is the Schröder-Bernstein Theorem. $\square$

-NoValue-

content/second-order-logic/sol-and-set-theory/cardinalities.tex

40.3 Cardinalities of Sets

sol:set:crd:
sec Just as we can express that the domain is finite or infinite, **enumerável** or explanation **não enumerável**, we can define the property of a subset of $|\mathfrak{M}|$ being finite or infinite, **enumerável** or **não enumerável**.

Proposição 40.7. *The formula $\text{Inf}(X) \equiv$*

$$\begin{aligned} \exists u (\forall x \forall y (u(x) = u(y) \rightarrow x = y) \wedge \\ \exists y (X(y) \wedge \forall x (X(x) \rightarrow y \neq u(x))) \end{aligned}$$

is satisfied with respect to a variable assignment s iff $s(X)$ is infinite.

Proposição 40.8. *The formula $\text{Count}(X) \equiv$*

$$\begin{aligned} \exists z \exists u (X(z) \wedge \forall x (X(x) \rightarrow X(u(x))) \wedge \\ \forall Y ((Y(z) \wedge \forall x (Y(x) \rightarrow Y(u(x)))) \rightarrow X = Y)) \end{aligned}$$

*is satisfied with respect to a variable assignment s iff $s(X)$ is **enumerável**.*

We know from Cantor's Theorem that there are *não enumerável* sets, and in fact, that there are infinitely many different levels of infinite sizes. Set theory develops an entire arithmetic of sizes of sets, and assigns infinite cardinal numbers to sets. The natural numbers serve as the cardinal numbers measuring the sizes of finite sets. The cardinality of *infinito enumerável* sets is the first infinite cardinality, called $\aleph_0$ ("aleph-nought" or "aleph-zero"). The next infinite size is $\aleph_1$. It is the smallest size a set can be without being countable (O.s., of size $\aleph_0$). We can define " X has size $\aleph_0$ " as $\text{Aleph}_0(X) \leftrightarrow \text{Inf}(X) \wedge \text{Count}(X)$. X has size $\aleph_1$ iff all its subsets are finite or have size $\aleph_0$, but is not itself of size $\aleph_0$. Hence we can express this by the formula $\text{Aleph}_1(X) \equiv \forall Y (Y \subseteq X \rightarrow (\neg \text{Inf}(Y) \vee \text{Aleph}_0(Y))) \wedge \neg \text{Aleph}_0(X)$. Being of size $\aleph_2$ is defined similarly, etc.

There is one size of special interest, the so-called cardinality of the continuum. It is the size of $\wp(\mathbb{N})$, or, equivalently, the size of $\mathbb{R}$. That a set is the size of the continuum can also be expressed in second-order logic, but requires a bit more work.

-NoValue-

`content/second-order-logic/sol-and-set-theory/power-of-continuum.tex`

40.4 The Power of the Continuum

explanation In second-order logic we can quantify over subsets of the *domínio*, but not over sets of subsets of the *domínio*. To do this directly, we would need *third-order* logic. For instance, if we wanted to state Cantor's Theorem that there is no *injetiva* function from the power set of a set to the set itself, we might try to formulate it as "for every set X , and every set P , if P is the power set of X , then not $P \preceq X$ ". And to say that P is the power set of X would require formalizing that the *membros* of P are all and only the subsets of X , so something like $\forall Y (P(Y) \leftrightarrow Y \subseteq X)$. The problem lies in $P(Y)$: that is not a *fórmula* of second-order logic, since only terms can be arguments to one-place relation variables like P . sol:set:pow:sec

We can, however, *simulate* quantification over sets of sets, if the domain is large enough. The idea is to make use of the fact that two-place relations R relates *membros* of the *domínio* to *membros* of the *domínio*. Given such an R , we can collect all the *membros* to which some x is R -related: $\{y \in |\mathfrak{M}| : R(x, y)\}$ is the set "coded by" x . Conversely, if $Z \subseteq \wp(|\mathfrak{M}|)$ is some collection of subsets of $|\mathfrak{M}|$, and there are at least as many *membros* of $|\mathfrak{M}|$ as there are sets in Z , then there is also a relation $R \subseteq |\mathfrak{M}|^2$ such that every $Y \in Z$ is coded by some x using R .

Definição 40.9. If $R \subseteq |\mathfrak{M}|^2$, then x *R-codes* $\{y \in |\mathfrak{M}| : R(x, y)\}$.

If *um membro* $x \in |\mathfrak{M}|$ *R-codes* a set $Z \subseteq |\mathfrak{M}|$, then a set $Y \subseteq |\mathfrak{M}|$ codes a set of sets, namely the sets coded by the *membros* of Y . So a set Y can *R-code*

40.4. THE POWER OF THE CONTINUUM

$\wp(X)$. It does so iff for every $Z \subseteq X$, some $x \in Y$ R -codes Z , and every $x \in Y$ R -codes a $Z \subseteq X$.

Proposição 40.10. *The fórmula*

$$\text{Codes}(x, R, Z) \equiv \forall y (Z(y) \leftrightarrow R(x, y))$$

expresses that $s(x)$ $s(R)$ -codes $s(Z)$. The fórmula

$$\begin{aligned} \text{Pow}(Y, R, X) \equiv \\ \forall Z (Z \subseteq X \rightarrow \exists x (Y(x) \wedge \text{Codes}(x, R, Z))) \wedge \\ \forall x (Y(x) \rightarrow \forall Z (\text{Codes}(x, R, Z) \rightarrow Z \subseteq X)) \end{aligned}$$

expresses that $s(Y)$ $s(R)$ -codes the power set of $s(X)$, O.s., the elements of $s(Y)$ $s(R)$ -code exactly the subsets of $s(X)$.

With this trick, we can express statements about the power set by quantifying over the codes of subsets rather than the subsets themselves. For instance, Cantor's Theorem can now be expressed by saying that there is no **injetiva** function from the domain of any relation that codes the power set of X to X itself. explanation

Proposição 40.11. *The sentence*

$$\begin{aligned} \forall X \forall Y \forall R (\text{Pow}(Y, R, X) \rightarrow \\ \neg \exists u (\forall x \forall y (u(x) = u(y) \rightarrow x = y) \wedge \\ \forall x (Y(x) \rightarrow X(u(x))))) \end{aligned}$$

is valid.

The power set of a **infinito enumerável** set is **não enumerável**, and so its cardinality is larger than that of any **infinito enumerável** set (which is $\aleph_0$). The size of $\wp(\mathbb{N})$ is called the “power of the continuum,” since it is the same size as the points on the real number line, $\mathbb{R}$. If the **domínio** is large enough to code the power set of a **infinito enumerável** set, we can express that a set is the size of the continuum by saying that it is equinumerous with any set Y that codes the power set of set X of size $\aleph_0$. (If the domain is not large enough, O.s., it contains no subset equinumerous with $\mathbb{R}$, then there can also be no relation that codes $\wp(X)$.) explanation

Proposição 40.12. *If $\mathbb{R} \preceq |\mathfrak{M}|$, then the fórmula*

$$\begin{aligned} \text{Cont}(Y) \equiv \exists X \exists R ((\text{Aleph}_0(X) \wedge \text{Pow}(Y, R, X)) \wedge \\ \forall x \forall y ((Y(x) \wedge Y(y) \wedge \forall z (R(x, z) \leftrightarrow R(y, z)) \rightarrow x = y)) \end{aligned}$$

expresses that $s(Y) \approx \mathbb{R}$.

Prova. $\text{Pow}(Y, R, X)$ expresses that $s(Y)$ $s(R)$ -codes the power set of $s(X)$, which $\text{Aleph}_0(X)$ says is countable. So $s(Y)$ is at least as large as the power of the continuum, although it may be larger (if multiple elements of $s(Y)$ code the same subset of X). This is ruled out by the last conjunct, which requires the association between elements of $s(Y)$ and subsets of $s(Z)$ via $s(R)$ to be *injetiva*. $\square$

Proposição 40.13. $|\mathfrak{M}| \approx \mathbb{R}$ iff

$$\begin{aligned} \mathfrak{M} \models \exists X \exists Y \exists R (\text{Aleph}_0(X) \wedge \text{Pow}(Y, R, X) \wedge \\ \exists u (\forall x \forall y (u(x) = u(y) \rightarrow x = y) \wedge \\ \forall y (Y(y) \rightarrow \exists x y = u(x))))). \end{aligned}$$

explanation The Continuum Hypothesis is the statement that the size of the continuum is the first *não enumerável* cardinality, i.e, that $\wp(\mathbb{N})$ has size $\aleph_1$.

Proposição 40.14. *The Continuum Hypothesis is true iff*

$$\text{CH} \equiv \forall X (\text{Aleph}_1(X) \leftrightarrow \text{Cont}(X))$$

is valid.

Note that it isn't true that $\neg\text{CH}$ is valid iff the Continuum Hypothesis is false. In a *enumerável* domain, there are no subsets of size $\aleph_1$ and also no subsets of the size of the continuum, so CH is always true in a *enumerável* domain. However, we can give a different sentence that is valid iff the Continuum Hypothesis is false:

Proposição 40.15. *The Continuum Hypothesis is false iff*

$$\text{NCH} \equiv \forall X (\text{Cont}(X) \rightarrow \exists Y (Y \subseteq X \wedge \neg\text{Count}(Y) \wedge \neg X \approx Y))$$

is valid.

Parte IX

The Lambda Calculus

This part deals with the lambda calculus. The introduction chapter is based on Jeremy Avigad's notes; part of it is now redundant and covered in later chapters. The chapters on syntax, Church–Rosser property, and lambda definability were produced by Zesen Qian during his Mitacs summer internship. They still have to be reviewed and revised.

Capítulo 41

Introduction

This chapter consists of Jeremy's original concise notes on the lambda calculus. The sections need to be combined, and the material on lambda definability merged with the material in the separate, more detailed chapter on lambda definability.

-NoValue-

[content/lambda-calculus/introduction/overview.tex](#)

41.1 Overview

lam:int:ovr:
sec

The lambda calculus was originally designed by Alonzo Church in the early 1930s as a basis for constructive logic, and *not* as a model of the computable functions. But it was soon shown to be equivalent to other definitions of computability, such as the Turing computable functions and the partial recursive functions. The fact that this initially came as a small surprise makes the characterization all the more interesting.

Lambda notation is a convenient way of referring to a function directly by a symbolic expression which defines it, instead of defining a name for it.

Instead of saying “let f be the function defined by $f(x) = x + 3$,” one can say, “let f be the function $\lambda x.(x + 3)$.” In other words, $\lambda x.(x + 3)$ is just a *name* for the function that adds three to its argument. In this expression, x is a dummy variable, or a placeholder: the same function can just as well be denoted by $\lambda y.(y + 3)$. The notation works even with other parameters around. For example, suppose $g(x, y)$ is a function of two variables, and k is a natural number. Then $\lambda x.g(x, k)$ is the function which maps any x to $g(x, k)$.

This way of defining a function from a symbolic expression is known as *lambda abstraction*. The flip side of lambda abstraction is *application*: assuming one has a function f (say, defined on the natural numbers), one can *apply* it to any value, like 2. In conventional notation, of course, we write $f(2)$ for the result.

What happens when you combine lambda abstraction with application? Then the resulting expression can be simplified, by “plugging” the applicand in for the abstracted variable. For example,

$$(\lambda x.(x + 3))(2)$$

can be simplified to $2 + 3$.

Up to this point, we have done nothing but introduce new notations for conventional notions. The lambda calculus, however, represents a more radical departure from the set-theoretic viewpoint. In this framework:

1. Everything denotes a function.
2. Functions can be defined using lambda abstraction.
3. Anything can be applied to anything else.

For example, if F is a term in the lambda calculus, $F(F)$ is always assumed to be meaningful. This liberal framework is known as the *untyped* lambda calculus, where “untyped” means “no restriction on what can be applied to what.”

digression

There is also a *typed* lambda calculus, which is an important variation on the untyped version. Although in many ways the typed lambda calculus is similar to the untyped one, it is much easier to reconcile with a classical set-theoretic framework, and has some very different properties.

Research on the lambda calculus has proved to be central in theoretical computer science, and in the design of programming languages. LISP, designed by John McCarthy in the 1950s, is an early example of a language that was influenced by these ideas.

-NoValue-

content/lambda-calculus/introduction/syntax.tex

41.2 The Syntax of the Lambda Calculus

One starts with a sequence of variables $x, y, z, \dots$ and some constant symbols $a, b, c, \dots$. The set of terms is defined inductively, as follows:

lam:int:syn:
sec

41.2. THE SYNTAX OF THE LAMBDA CALCULUS

1. Each variable is a term.
2. Each constant is a term.
3. If M and N are terms, so is (MN) .
4. If M is a term and x is a variable, then $(\lambda x. M)$ is a term.

Terms of the form (MN) are called *applications* and those of the form $(\lambda x. M)$ *abstractions*.

The system without any constants at all is called the *pure* lambda calculus. We'll mainly be working in the pure λ -calculus, so all lowercase letters will stand for variables. We use uppercase letters (M , N , etc.) to stand for terms of the λ -calculus.

We will follow a few notational conventions:

- Convenção 1.*
1. When parentheses are left out, application takes place from left to right. For example, if M , N , P , and Q are terms, then $MNPQ$ abbreviates $((MN)P)Q$.
 2. Again, when parentheses are left out, lambda abstraction is to be given the widest scope possible. From example, $\lambda x. MNP$ is read $(\lambda x. ((MN)P))$.
 3. A lambda can be used to abstract multiple variables. For example, $\lambda xyz. M$ is short for $\lambda x. \lambda y. \lambda z. M$.

For example,

$$\lambda xy. xxyx \lambda z. xz$$

abbreviates

$$\lambda x. \lambda y. (((xxy)x)(\lambda z. (xz))).$$

You should memorize these conventions. They will drive you crazy at first, but you will get used to them, and after a while they will drive you less crazy than having to deal with a morass of parentheses.

Two terms that differ only in the names of the bound variables are called α -equivalent; for example, $\lambda x. x$ and $\lambda y. y$. It will be convenient to think of these as being the “same” term; in other words, when we say that M and N are the same, we also mean “up to renamings of the bound variables.” Variables that are in the scope of a λ are called “bound”, while others are called “free.” There are no free variables in the previous example; but in

$$(\lambda z. yz)x$$

y and x are free, and z is bound.

-NoValue-

content/lambda-calculus/introduction/reduction.tex

41.3 Reduction of Lambda Terms

What can one do with lambda terms? Simplify them. If M and N are any lambda terms and x is any variable, we can use $M[N/x]$ to denote the result of substituting N for x in M , after renaming any bound variables of M that would interfere with the free variables of N after the substitution. For example,

$$(\lambda w. xxw)[yyz/x] = \lambda w. (yyz)(yyz)w.$$

digression

Alternative notations for substitution are $[N/x]M$, $[x/N]M$, and also $M[x/N]$. Beware!

Intuitively, $(\lambda x. M)N$ and $M[N/x]$ have the same meaning; the act of replacing the first term by the second is called β -contraction. $(\lambda x. M)N$ is called a *redex* and $M[N/x]$ its *contractum*. Generally, if it is possible to change a term P to P' by β -contraction of some subterm, we say that P β -reduces to P' in one step, and write $P \rightarrow P'$. If from P we can obtain P' with some number of one-step reductions (possibly none), then P β -reduces to P' ; in symbols, $P \twoheadrightarrow P'$. A term that cannot be β -reduced any further is called β -irreducible, or β -normal. We will say “reduces” instead of “ β -reduces,” etc., when the context is clear.

Let us consider some examples.

1. We have

$$\begin{aligned} (\lambda x. xxy)\lambda z. z &\rightarrow (\lambda z. z)(\lambda z. z)y \\ &\rightarrow (\lambda z. z)y \\ &\rightarrow y. \end{aligned}$$

2. “Simplifying” a term can make it more complex:

$$\begin{aligned} (\lambda x. xxy)(\lambda x. xxy) &\rightarrow (\lambda x. xxy)(\lambda x. xxy)y \\ &\rightarrow (\lambda x. xxy)(\lambda x. xxy)yy \\ &\rightarrow \dots \end{aligned}$$

3. It can also leave a term unchanged:

$$(\lambda x. xx)(\lambda x. xx) \rightarrow (\lambda x. xx)(\lambda x. xx).$$

4. Also, some terms can be reduced in more than one way; for example,

$$(\lambda x. (\lambda y. yx)z)v \rightarrow (\lambda y. yv)z$$

by contracting the outermost application; and

$$(\lambda x. (\lambda y. yx)z)v \rightarrow (\lambda x. zx)v$$

by contracting the innermost one. Note, in this case, however, that both terms further reduce to the same term, zv .

41.4. THE CHURCH-ROSSER PROPERTY

The final outcome in the last example is not a coincidence, but rather illustrates a deep and important property of the lambda calculus, known as the “Church–Rosser property.”

-NoValue-

[content/lambda-calculus/introduction/church-rosser.tex](#)

41.4 The Church–Rosser Property

lam:int:cr:

sec

lam:int:cr:

thm:church-rosser

Teorema 41.1. *Let M , N_1 , and N_2 be terms, such that $M \rightarrow N_1$ and $M \rightarrow N_2$. Then there is a term P such that $N_1 \rightarrow P$ and $N_2 \rightarrow P$.*

Corolário 41.2. *Suppose M can be reduced to normal form. Then this normal form is unique.*

Prova. If $M \rightarrow N_1$ and $M \rightarrow N_2$, by the previous theorem there is a term P such that N_1 and N_2 both reduce to P . If N_1 and N_2 are both in normal form, this can only happen if $N_1 \equiv P \equiv N_2$. $\square$

Finally, we will say that two terms M and N are β -equivalent, or just *equivalent*, if they reduce to a common term; in other words, if there is some P such that $M \rightarrow P$ and $N \rightarrow P$. This is written $M \stackrel{\beta}{=} N$. Using [Teorema 41.1](#), you can check that $\stackrel{\beta}{=}$ is an equivalence relation, with the additional property that for every M and N , if $M \rightarrow N$ or $N \rightarrow M$, then $M \stackrel{\beta}{=} N$. (In fact, one can show that $\stackrel{\beta}{=}$ is the *smallest* equivalence relation having this property.)

-NoValue-

[content/lambda-calculus/introduction/currying.tex](#)

41.5 Currying

lam:rep:cur:

sec

A λ -abstract $\lambda x.M$ represents a function of one argument, which is quite a limitation when we want to define function accepting multiple arguments. One way to do this would be by extending the λ -calculus to allow the formation of pairs, triples, etc., in which case, say, a three-place function $\lambda x.M$ would expect its argument to be a triple. However, it is more convenient to do this by *Currying*.

Let’s consider an example. We’ll pretend for a moment that we have a $+$ operation in the λ -calculus. The addition function is 2-place, O.s., it takes two arguments. But a λ -abstract only gives us functions of one argument: the syntax does not allow expressions like $\lambda(x,y).(x+y)$. However, we can consider the one-place function $f_x(y)$ given by $\lambda y.(x+y)$, which adds x to its single argument y . Actually, this is not a single function, but a family of different functions “add x ,” one for each number x . Now we can define another one-place function g as $\lambda x.f_x$. Applied to argument x , $g(x)$ returns the function f_x —so its values are other functions. Now if we apply g to x , and then the result to y

we get: $(g(x))y = f_x(y) = x + y$. In this way, the one-place function g can do the same job as the two-place addition function. “Currying” simply refers to this trick for turning two-place functions into one place functions (whose values are one-place functions).

Here is an example properly in the syntax of the λ -calculus. How do we represent the function $f(x, y) = x$? If we want to define a function that accepts two arguments and returns the first, we can write $\lambda x. \lambda y. x$, which literally is a function that accepts an argument x and returns the function $\lambda y. x$. The function $\lambda y. x$ accepts another argument y , but drops it, and always returns x . Let’s see what happens when we apply $\lambda x. \lambda y. x$ to two arguments:

$$\begin{aligned} (\lambda x. \lambda y. x)MN &\xrightarrow{\beta} (\lambda y. M)N \\ &\xrightarrow{\beta} M \end{aligned}$$

In general, to write a function with parameters $x_1, \dots, x_n$ defined by some term N , we can write $\lambda x_1. \lambda x_2. \dots \lambda x_n. N$. If we apply n arguments to it we get:

$$\begin{aligned} (\lambda x_1. \lambda x_2. \dots \lambda x_n. N)M_1 \dots M_n &\xrightarrow{\beta} \\ &\xrightarrow{\beta} ((\lambda x_2. \dots \lambda x_n. N)[M_1/x_1])M_2 \dots M_n \\ &\equiv (\lambda x_2. \dots \lambda x_n. N[M_1/x_1])M_2 \dots M_n \\ &\vdots \\ &\xrightarrow{\beta} P[M_1/x_1] \dots [M_n/x_n] \end{aligned}$$

The last line literally means substituting M_i for x_i in the body of the function definition, which is exactly what we want when applying multiple arguments to a function.

-NoValue-

<content/lambda-calculus/introduction/lambda-definability.tex>

41.6 λ -definível Arithmetical Functions

How can the lambda calculus serve as a model of computation? At first, it is not even clear how to make sense of this statement. To talk about computability on the natural numbers, we need to find a suitable representation for such numbers. Here is one that works surprisingly well.

lam:int:rep:
sec

Definição 41.3. For each natural number n , define the *Church numeral* $\bar{n}$ to be the lambda term $\lambda x. \lambda y. (x(x(x(\dots x(y))))))$, where there are n x ’s in all.

The terms $\bar{n}$ are “iterators”: on input f , $\bar{n}$ returns the function mapping y to $f^n(y)$. Note that each numeral is normal. We can now say what it means for a lambda term to “compute” a function on the natural numbers.

41.7. λ -DEFINÍVEL FUNCTIONS ARE COMPUTABLE

Definição 41.4. Let $f(x_0, \dots, x_{k-1})$ be an n -ary partial function from $\mathbb{N}$ to $\mathbb{N}$. We say a λ -term F *λ -definem* f iff for every sequence of natural numbers $n_0, \dots, n_{k-1}$,

$$F \overline{n_0} \overline{n_1} \dots \overline{n_{k-1}} \twoheadrightarrow \overline{f(n_0, n_1, \dots, n_{k-1})}$$

if $f(n_0, \dots, n_{k-1})$ is defined, and $F, \overline{n_0} \overline{n_1} \dots \overline{n_{k-1}}$ has no normal form otherwise.

lam:int:rep: thm:lambda-def **Teorema 41.5.** *A function f is a partial computable function if and only if it is λ -definido by a lambda term.*

This theorem is somewhat striking. As a model of computation, the lambda calculus is a rather simple calculus; the only operations are lambda abstraction and application! From these meager resources, however, it is possible to implement any computational procedure.

explanation

-NoValue-

content/lambda-calculus/introduction/lambda-computable.tex

41.7 λ -definível Functions are Computable

lam:int:cmp: sec lam:int:cmp: thm:lambda-computable

Teorema 41.6. *If a partial function f is λ -definido by a lambda term, it is computable.*

Prova. Suppose a function f is λ -definido by a lambda term X . Let us describe an informal procedure to compute f . On input $m_0, \dots, m_{n-1}$, write down the term $X \overline{m_0} \dots \overline{m_{n-1}}$. Build a tree, first writing down all the one-step reductions of the original term; below that, write all the one-step reductions of those (O.s., the two-step reductions of the original term); and keep going. If you ever reach a numeral, return that as the answer; otherwise, the function is undefined.

An appeal to Church's thesis tells us that this function is computable. A better way to prove the theorem would be to give a recursive description of this search procedure. For example, one could define a sequence primitive recursive functions and relations, "IsASubterm," "Substitute," "ReducesToInOneStep," "ReductionSequence," "Numeral," etc. The partial recursive procedure for computing $f(m_0, \dots, m_{n-1})$ is then to search for a sequence of one-step reductions starting with $X \overline{m_0} \dots \overline{m_{n-1}}$ and ending with a numeral, and return the number corresponding to that numeral. The details are long and tedious but otherwise routine. $\square$

-NoValue-

content/lambda-calculus/introduction/computable-lambda.tex

41.8 Computable Functions are λ -definível

lam:int:lrp: sec lam:int:lrp: thm:computable-lambda

Teorema 41.7. *Every computable partial function is λ -definível.*

Prova. We need to show that every partial computable function f is λ -definido by a lambda term F . By Kleene's normal form theorem, it suffices to show that every primitive recursive function is λ -definido by a lambda term, and then that the functions λ -definível are closed under suitable compositions and unbounded search. To show that every primitive recursive function is λ -definido by a lambda term, it suffices to show that the initial functions are λ -definível, and that the partial functions that are λ -definível are closed under composition, primitive recursion, and unbounded search. $\square$

We will use a more conventional notation to make the rest of the proof more readable. For example, we will write $M(x, y, z)$ instead of $Mxyz$. While this is suggestive, you should remember that terms in the untyped lambda calculus do not have associated arities; so, for the same term M , it makes just as much sense to write $M(x, y)$ and $M(x, y, z, w)$. But using this notation indicates that we are treating M as a function of three variables, and helps make the intentions behind the definitions clearer. In a similar way, we will say “define M by $M(x, y, z) = \dots$ ” instead of “define M by $M = \lambda x. \lambda y. \lambda z. \dots$ ”

-NoValue-

[content/lambda-calculus/introduction/basic-pr-lambda.tex](#)

41.9 The Basic Primitive Recursive Functions are λ -definível

Lema 41.8. *The functions zero, succ, and P_i^n are λ -definível.*

lam:int:bas:
sec

Prova. zero is just $\lambda x. \lambda y. y$.

The successor function succ, is defined by $\text{Succ}(u) = \lambda x. \lambda y. x(uxy)$. You should think about why this works; for each numeral $\bar{n}$, thought of as an iterator, and each function f , $\text{Succ}(\bar{n}, f)$ is a function that, on input y , applies f n times starting with y , and then applies it once more.

There is nothing to say about projections: $\text{Proj}_i^n(x_0, \dots, x_{n-1}) = x_i$. In other words, by our conventions, Proj_i^n is the lambda term $\lambda x_0. \dots \lambda x_{n-1}. x_i$. $\square$

-NoValue-

[content/lambda-calculus/introduction/composition.tex](#)

41.10 The λ -definível Functions are Closed under Composition

Lema 41.9. *The λ -definível functions are closed under composition.*

lam:int:com:
sec

41.11. λ -DEFINÍVEL FUNCTIONS ARE CLOSED UNDER PRIMITIVE RECURSION

Prova. Suppose f is defined by composition from $h, g_0, \dots, g_{k-1}$. Assuming $h, g_0, \dots, g_{k-1}$ are λ -definido by $H, G_0, \dots, G_{k-1}$, respectively, we need to find a term F that λ -definem f . But we can simply define F by

$$F(x_0, \dots, x_{l-1}) = H(G_0(x_0, \dots, x_{l-1}), \dots, G_{k-1}(x_0, \dots, x_{l-1})).$$

In other words, the language of the lambda calculus is well suited to represent composition. $\square$

-NoValue-

 content/lambda-calculus/introduction/primitive-recursion.tex

41.11 λ -definível Functions are Closed under Primitive Recursion

When it comes to primitive recursion, we finally need to do some work. We will have to proceed in stages. As before, on the assumption that we already have terms G and H that λ -define functions g and h , respectively, we want a term F that λ -definem the function f defined by

$$\begin{aligned} f(0, \vec{z}) &= g(\vec{z}) \\ f(x+1, \vec{z}) &= h(x, f(x, \vec{z}), \vec{z}). \end{aligned}$$

So, in general, given lambda terms G' and H' , it suffices to find a term F such that

$$\begin{aligned} F(\bar{0}, \vec{z}) &\equiv G'(\vec{z}) \\ F(\overline{n+1}, \vec{z}) &\equiv H'(\bar{n}, F(\bar{n}, \vec{z}), \vec{z}) \end{aligned}$$

for every natural number n ; the fact that G' and H' λ -define g and h means that whenever we plug in numerals $\bar{m}$ for $\vec{z}$, $F(\overline{n+1}, \bar{m})$ will normalize to the right answer.

But for this, it suffices to find a term F satisfying

$$\begin{aligned} F(\bar{0}) &\equiv G \\ F(\overline{n+1}) &\equiv H(\bar{n}, F(\bar{n})) \end{aligned}$$

for every natural number n , where

$$\begin{aligned} G &= \lambda \vec{z}. G'(\vec{z}) \text{ and} \\ H(u, v) &= \lambda \vec{z}. H'(u, v(u, \vec{z}), \vec{z}). \end{aligned}$$

In other words, with lambda trickery, we can avoid having to worry about the extra parameters $\vec{z}$ —they just get absorbed in the lambda notation.

Before we define the term F , we need a mechanism for handling ordered pairs. This is provided by the next lemma.

Lema 41.10. *There is a lambda term D such that for each pair of lambda terms M and N , $D(M, N)(\bar{0}) \twoheadrightarrow M$ and $D(M, N)(\bar{1}) \twoheadrightarrow N$.*

Prova. First, define the lambda term K by

$$K(y) = \lambda x. y.$$

In other words, K is the term $\lambda y. \lambda x. y$. Looking at it differently, for every M , $K(M)$ is a constant function that returns M on any input.

Now define $D(x, y, z)$ by $D(x, y, z) = z(K(y))x$. Then we have

$$\begin{aligned} D(M, N, \bar{0}) &\twoheadrightarrow \bar{0}(K(N))M \twoheadrightarrow M \text{ and} \\ D(M, N, \bar{1}) &\twoheadrightarrow \bar{1}(K(N))M \twoheadrightarrow K(N)M \twoheadrightarrow N, \end{aligned}$$

as required. $\square$

The idea is that $D(M, N)$ represents the pair $\langle M, N \rangle$, and if P is assumed to represent such a pair, $P(\bar{0})$ and $P(\bar{1})$ represent the left and right projections, $(P)_0$ and $(P)_1$. We will use the latter notations.

Lema 41.11. *The λ -definível functions are closed under primitive recursion.*

Prova. We need to show that given any terms, G and H , we can find a term F such that

$$\begin{aligned} F(\bar{0}) &\equiv G \\ F(\overline{n+1}) &\equiv H(\bar{n}, F(\bar{n})) \end{aligned}$$

for every natural number n . The idea is roughly to compute sequences of *pairs*

$$\langle \bar{0}, F(\bar{0}) \rangle, \langle \bar{1}, F(\bar{1}) \rangle, \dots,$$

using numerals as iterators. Notice that the first pair is just $\langle \bar{0}, G \rangle$. Given a pair $\langle \bar{n}, F(\bar{n}) \rangle$, the next pair, $\langle \overline{n+1}, F(\overline{n+1}) \rangle$ is supposed to be equivalent to $\langle \overline{n+1}, H(\bar{n}, F(\bar{n})) \rangle$. We will design a lambda term T that makes this one-step transition.

The details are as follows. Define $T(u)$ by

$$T(u) = \langle S((u)_0), H((u)_0, (u)_1) \rangle.$$

Now it is easy to verify that for any number n ,

$$T(\langle \bar{n}, M \rangle) \twoheadrightarrow \langle \overline{n+1}, H(\bar{n}, M) \rangle.$$

As suggested above, given G and H , define $F(u)$ by

$$F(u) = (u(T, \langle \bar{0}, G \rangle))_1.$$

In other words, on input $\bar{n}$, F iterates T n times on $\langle \bar{0}, G \rangle$, and then returns the second component. To start with, we have

41.12. FIXED-POINT COMBINATORS

$$1. \bar{0}(T, \langle \bar{0}, G \rangle) \equiv \langle \bar{0}, G \rangle$$

$$2. F(\bar{0}) \equiv G$$

By induction on n , we can show that for each natural number one has the following:

$$1. \overline{n+1}(T, \langle \bar{0}, G \rangle) \equiv \langle \overline{n+1}, F(\overline{n+1}) \rangle$$

$$2. F(\overline{n+1}) \equiv H(\bar{n}, F(\bar{n}))$$

For the second clause, we have

$$\begin{aligned} F(\overline{n+1}) &\twoheadrightarrow (\overline{n+1}(T, \langle \bar{0}, G \rangle))_1 \\ &\equiv (T(\bar{n}(T, \langle \bar{0}, G \rangle)))_1 \\ &\equiv (T(\langle \bar{n}, F(\bar{n}) \rangle))_1 \\ &\equiv (\langle \overline{n+1}, H(\bar{n}, F(\bar{n})) \rangle)_1 \\ &\equiv H(\bar{n}, F(\bar{n})). \end{aligned}$$

Here we have used the induction hypothesis on the second-to-last line. For the first clause, we have

$$\begin{aligned} \overline{n+1}(T, \langle \bar{0}, G \rangle) &\equiv T(\bar{n}(T, \langle \bar{0}, G \rangle)) \\ &\equiv T(\langle \bar{n}, F(\bar{n}) \rangle) \\ &\equiv \langle \overline{n+1}, H(\bar{n}, F(\bar{n})) \rangle \\ &\equiv \langle \overline{n+1}, F(\overline{n+1}) \rangle. \end{aligned}$$

Here we have used the second clause in the last line. So we have shown $F(\bar{0}) \equiv G$ and, for every n , $F(\overline{n+1}) \equiv H(\bar{n}, F(\bar{n}))$, which is exactly what we needed. $\square$

-NoValue- content/lambda-calculus/introduction/fixed-point-combinator.tex

41.12 Fixed-Point Combinators

lam:int:fix: Suppose you have a lambda term g , and you want another term k with the sec property that k is β -equivalent to gk . Define terms

$$\text{diag}(x) = xx$$

and

$$l(x) = g(\text{diag}(x))$$

using our notational conventions; in other words, l is the term $\lambda x. g(xx)$. Let k be the term ll . Then we have

$$\begin{aligned} k &= (\lambda x. g(xx))(\lambda x. g(xx)) \\ &\twoheadrightarrow g((\lambda x. g(xx))(\lambda x. g(xx))) \\ &= gk. \end{aligned}$$

If one takes

$$Y = \lambda g. ((\lambda x. g(xx))(\lambda x. g(xx)))$$

then Yg and $g(Yg)$ reduce to a common term; so $Yg \equiv_{\beta} g(Yg)$. This is known as “Curry’s combinator.” If instead one takes

$$Y = (\lambda xg. g(xg))(\lambda xg. g(xg))$$

then in fact Yg reduces to $g(Yg)$, which is a stronger statement. This latter version of Y is known as “Turing’s combinator.”

-NoValue-

[content/lambda-calculus/introduction/minimization.tex](#)

41.13 The λ -definível Functions are Closed under Minimization

Lema 41.12. Suppose $f(x, y)$ is λ -definível. Let g be defined by

$$g(x) \simeq \mu y f(x, y).$$

Then g is λ -definível.

Prova. The idea is roughly as follows. Given x , we will use the fixed-point lambda term Y to define a function $h_x(n)$ which searches for a y starting at n ; then $g(x)$ is just $h_x(0)$. The function h_x can be expressed as the solution of a fixed-point equation:

$$h_x(n) \simeq \begin{cases} n & \text{if } f(x, n) = 0 \\ h_x(n+1) & \text{otherwise.} \end{cases}$$

Here are the details. Since f is primitive recursive, it is λ -definido by some term F . Remember that we also have a lambda term D , such that $D(M, N, \bar{0}) \rightarrow M$ and $D(M, N, \bar{1}) \rightarrow N$. Fixing x for the moment, to λ -define h_x we want to find a term H (depending on x) satisfying

$$H(\bar{n}) \equiv D(\bar{n}, H(S(\bar{n})), F(x, \bar{n})).$$

We can do this using the fixed-point term Y . First, let U be the term

$$\lambda h. \lambda z. D(z, (h(Sz)), F(x, z)),$$

and then let H be the term YU . Notice that the only free variable in H is x . Let us show that H satisfies the equation above.

By the definition of Y , we have

$$H = YU \equiv U(YU) = U(H).$$

In particular, for each natural number n , we have

$$\begin{aligned} H(\bar{n}) &\equiv U(H, \bar{n}) \\ &\rightarrow D(\bar{n}, H(S(\bar{n})), F(x, \bar{n})), \end{aligned}$$

as required. Notice that if you substitute a numeral $\bar{m}$ for x in the last line, the expression reduces to $\bar{n}$ if $F(\bar{m}, \bar{n})$ reduces to $\bar{0}$, and it reduces to $H(S(\bar{n}))$ if $F(\bar{m}, \bar{n})$ reduces to any other numeral.

To finish off the proof, let G be $\lambda x. H(\bar{0})$. Then G **λ -definem** g ; in other words, for every m , $G(\bar{m})$ reduces to $\overline{g(m)}$, if $g(m)$ is defined, and has no normal form otherwise. $\square$

Capítulo 42

Syntax

-NoValue-

[content/lambda-calculus/syntax/terms.tex](#)

42.1 Terms

lam:syn:trm: The terms of the lambda calculus are built up inductively from an infinite **sec** supply of variables $v_0, v_1, \dots$, the symbol “ λ ”, and parentheses. We will use $x, y, z, \dots$ to designate variables, and $M, N, P, \dots$ to designate terms.

lam:syn:trm: **Definição 42.1 (Terms).** The set of *terms* of the lambda calculus is defined **defn:term** inductively by:

- lam:syn:trm:** 1. If x is a variable, then x is a term.
- defn:term-var** 2. If x is a variable and M is a term, then $(\lambda x. M)$ is a term.
- lam:syn:trm:** 3. If both M and N are terms, then (MN) is a term.
- defn:term-abs**
- lam:syn:trm:**
- defn:term-app**

If a term $(\lambda x. M)$ is formed according to **(2)** we say it is the result of an *abstraction*, and the x in λx is called a *parâmetro*. A term (MN) formed according to **(3)** is the result of an *application*.

The terms defined above are fully parenthesized. This can get rather cumbersome, as the term $(\lambda x. ((\lambda x. x)(\lambda x. (xx))))$ demonstrates. We will introduce

conventions for avoiding parentheses. However, the official definition makes it easy to determine how a term is constructed according to [Definição 42.1](#). For example, the last step of forming the term $(\lambda x. ((\lambda x. x)(\lambda x. (xx))))$ must be abstraction where the [parâmetro](#) is x . It results by abstraction from the term $((\lambda x. x)(\lambda x. (xx)))$, which is an application of two terms. Each of these two terms is the result of an abstraction, and so on.

Problema 42.1. Describe the formation of $(\lambda g. (\lambda x. (g(xx)))(\lambda x. (g(xx))))$.

-NoValue-

[content/lambda-calculus/syntax/unique-readability.tex](#)

42.2 Unique Readability

We may wonder if for each term there is a unique way of forming it, and there is. For each lambda term there is only one way to construct and interpret it. In the following discussion, a *formation* is the procedure of constructing a term using the formation rules (one or several times) of [Definição 42.1](#).

lam:syn:unq:
sec

Lema 42.2. *A term starts with either a variable or a parenthesis.*

lam:syn:unq:
lem:term-start

Prova. Something counts as a term only if it is constructed according to [Definição 42.1](#). If it is the result of (1), it must be a variable. If it is the result of (2) or (3), it starts with a parenthesis. $\square$

Lema 42.3. *The result of an application starts with either two parentheses or a parenthesis and a variable.*

lam:syn:unq:
lem:app-start

Prova. If M is the result of an application, it is of the form (PQ) , so it begins with a parenthesis. Since P is a term, by [Lema 42.2](#), it begins either with a parenthesis or a variable. $\square$

Lema 42.4. *No proper initial part of a term is itself a term.*

lam:syn:unq:
lem:initial

Problema 42.2. Prove [Lema 42.4](#) by induction on the length of terms.

Proposição 42.5 (Unique Readability). *There is a unique formation for each term. In other words, if a term M is formed by a formation, then it is the only formation that can form this term.*

lam:syn:unq:
prop:unq

Prova. We prove this by induction on the formation of terms.

1. M is of the form x , where x is some variable. Since the results of abstractions and applications always start with parentheses, they cannot have been used to construct M ; Thus, the formation of M must be a single step of [Definição 42.1\(1\)](#).

42.3. ABBREVIATED SYNTAX

2. M is of the form $(\lambda x. N)$, where x is some variable and N is a term. It could not have been constructed according to [Definição 42.1\(1\)](#), because it is not a single variable. It is not the result of an application, by [Lema 42.3](#). Thus M can only be the result of an abstraction on N . By inductive hypothesis we know that formation of N is itself unique.
3. M is of the form (PQ) , where P and Q are terms. Since it starts with a parentheses, it cannot also be constructed by [Definição 42.1\(1\)](#). By [Lema 42.2](#), P cannot begin with λ , so (PQ) cannot be the result of an abstraction. Now suppose there were another way of constructing M by application, p.ex., it is also of the form $(P'Q')$. Then P is a proper initial segment of P' (or vice versa), and this is impossible by [Lema 42.4](#). So P and Q are uniquely determined, and by inductive hypothesis we know that formations of P and Q is unique. $\square$

A more readable paraphrase of the above proposition is as follows:

Proposição 42.6. *A term M can only be one of the following forms:*

1. x , where x is a variable uniquely determined by M .
2. $(\lambda x. N)$, where x is a variable and N is another term, both of which is uniquely determined by M .
3. (PQ) , where P and Q are two terms uniquely determined by M .

-NoValue-

[content/lambda-calculus/syntax/abbreviated-syntax.tex](#)

42.3 Abbreviated Syntax

lam:syn:abb:
sec

Terms as defined in [Definição 42.1](#) are sometimes cumbersome to write, so it is useful to introduce a more concise syntax. We must of course be careful to make sure that the terms in the concise notation also are uniquely readable. One widely used version called *abbreviated terms* is as follows.

1. When parentheses are left out, application takes place from left to right. For example, if M , N , P , and Q are terms, then $MNPQ$ abbreviates $((MN)P)Q$.
2. Again, when parentheses are left out, lambda abstraction is given the widest scope possible. For example, $\lambda x. MNP$ is read as $(\lambda x. MNP)$.
3. A lambda can be used to abstract multiple variables. For example, $\lambda xyz. M$ is short for $\lambda x. \lambda y. \lambda z. M$.

For example,

$$\lambda xy. xxyx\lambda z. xz$$

abbreviates

$$(\lambda x. (\lambda y. (((x)y)x)(\lambda z. (xz)))).$$

Problema 42.3. Expand the abbreviated term $\lambda g. (\lambda x. g(xx)) \lambda x. g(xx)$.

-NoValue-

[content/lambda-calculus/syntax/free-variables.tex](#)

42.4 Free Variables

Lambda calculus is about functions, and lambda abstraction is how functions arise. Intuitively, $\lambda x. M$ is the function with values given by M when the argument to the function is assigned to x . But not every occurrence of x in M is relevant: if M contains another abstract $\lambda x. N$ then the occurrences of x in N are relevant to $\lambda x. N$ but not to $\lambda x. M$. So, a lambda abstract λx inside $\lambda x. M$ *binds* those occurrences of x in M that are not already bound by another lambda abstract—the *free* occurrences of x in M .

lam:syn:fv:
sec

Definição 42.7 (Scope). If $\lambda x. M$ occurs inside a term N , then the corresponding occurrence of N is the *scope* of the λx .

Definição 42.8 (Free and bound occurrence). An occurrence of variable x in a term M is *free* if it is not in the scope of a λx , and *bound* otherwise. An occurrence of a variable x in $\lambda x. M$ is bound by the initial λx iff the occurrence of x in M is free.

Exemplo 42.9. In $\lambda x. xy$, both x and y are in the scope of λx , so x is bound by λx . Since y is not in the scope of any λy , it is free. In $\lambda x. xx$, both occurrences of x are bound by λx , since both are free in xx . In $((\lambda x. xx)x)$, the last occurrence of x is free, since it is not in the scope of a λx . In $\lambda x. (\lambda x. x)x$, the scope of the first λx is $(\lambda x. x)x$ and the scope of the second λx is the second-to-last occurrence of x . In $(\lambda x. x)x$, the last occurrence of x is free, and the second-to-last is bound. Thus, the second-to-last occurrence of x in $\lambda x. (\lambda x. x)x$ is bound by the second λx , and the last occurrence by the first λx .

For a term P , we can check all variable occurrences in it and get a set of free variables. This set is denoted by $FV(P)$ with a natural definition as follows:

Definição 42.10 (Free variables of a term). The set of *free variables* of a term is defined inductively by:

lam:syn:fv:
def:fv

1. $FV(x) = \{x\}$
2. $FV(\lambda x. N) = FV(N) \setminus \{x\}$
3. $FV(PQ) = FV(P) \cup FV(Q)$

lam:syn:fv:
def:fv1
lam:syn:fv:
def:fv2
lam:syn:fv:
def:fv3

Problema 42.4. 1. Identify the scopes of λg and the two λx in this term:
 $\lambda g. (\lambda x. g(xx)) \lambda x. g(xx)$.

2. In $\lambda g. (\lambda x. g(xx)) \lambda x. g(xx)$, are all occurrences of variables bound? By which abstractions are they bound respectively?

42.5. SUBSTITUTION

3. Give $\text{FV}(\lambda x. (\lambda y. (\lambda z. xy)z)y)$

A free variable is like a reference to the outside world (the *environment*), explanation and a term containing free variables can be seen as a partially specified term, since its behaviour depends on how we set up the environment. For example, in the term $\lambda x. fx$, which accepts an argument x and returns f of that argument, the variable f is free. This value of the term is dependent on the environment it is in, in particular the value of f in that environment.

If we apply abstraction to this term, we get $\lambda f. \lambda x. fx$. This term is no longer dependent on the environment variable f , because it now designates a function that accepts two arguments and returns the result of applying the first to the second. Changing f in the environment won't have any effect on the behavior of this term, as the term will only use whatever is passed as an argument, and not the value of f in the environment.

Definição 42.11 (Closed term, combinator). A term with no free variables is called a *closed term*, or a *combinator*.

lam:syn:fv:
lem:fv **Lema 42.12.**

1. If $y \neq x$, then $y \in \text{FV}(\lambda x. N)$ iff $y \in \text{FV}(N)$.
2. $y \in \text{FV}(PQ)$ iff $y \in \text{FV}(P)$ or $y \in \text{FV}(Q)$.

Prova. Exercise. □

Problema 42.5. Prove [Lema 42.12](#).

-NoValue- content/lambda-calculus/syntax/substitution.tex

42.5 Substitution

lam:syn:sub:
sec Free variables are references to environment variables, thus it makes sense to explanation actually use a specific value in the place of a free variable. For example, we may want to replace f in $\lambda x. fx$ with a specific term, like the identity function $\lambda y. y$. This results in $\lambda x. (\lambda y. y)x$. The process of replacing free variables with lambda terms is called substitution.

lam:syn:sub:
defn:substitution **Definição 42.13 (Substitution).** The *substitution* of a term N for a variable x in a term M , $M[N/x]$, is defined inductively by:

1. $x[N/x] = N$.
 2. $y[N/x] = y$ if $x \neq y$.
 3. $PQ[N/x] = (P[N/x])(Q[N/x])$.
- lam:syn:sub:
defn:substitution-1
lam:syn:sub:
defn:substitution-2
lam:syn:sub:
defn:substitution-3

4. $(\lambda y. P)[N/x] = \lambda y. P[N/x]$, if $x \neq y$ and $y \notin \text{FV}(N)$, otherwise undefined.

lam:syn:sub:
defn:substitution-4

explanation

In **Definição 42.13(4)**, we require $x \neq y$ because we don't want to replace *bound* occurrences of the variable x in M by N . For example, if we compute the substitution $\lambda x. x[y/x]$, the result should not be $\lambda x. y$ but simply $\lambda x. x$.

When substituting N for x in $\lambda y. P$, we also require that $y \notin \text{FV}(N)$. For example, we cannot substitute y for x in $\lambda y. x$, O.s., $\lambda y. x[y/x]$, because it would result in $\lambda y. y$, a term that stands for the function that accepts an argument and returns it directly. But the term $\lambda y. x$ stands for a function that always returns the term x (or whatever x refers to). So the result we actually want is a function that accepts an argument, drop it, and returns the environment variable y . To do this properly, we would first have to “rename” the bound variable y .

Problema 42.6. What is the result of the following substitutions?

1. $\lambda y. x(\lambda w. vwx)[(uv)/x]$
2. $\lambda y. x(\lambda x. x)[(\lambda y. xy)/x]$
3. $y(\lambda v. xv)[(\lambda y. vy)/x]$

Teorema 42.14. *If $x \notin \text{FV}(M)$, then $\text{FV}(M[N/x]) = \text{FV}(M)$, if the left-hand side is defined.*

lam:syn:sub:
thm:notinfv

Prova. By induction on the formation of M .

1. M is a variable: exercise.
2. M is of the form (PQ) : exercise.
3. M is of the form $\lambda y. P$, and since $\lambda y. P[N/x]$ is defined, it has to be $\lambda y. P[N/x]$. Then $P[N/x]$ has to be defined; also, $x \neq y$ and $x \notin \text{FV}(Q)$. Then:

$$\begin{aligned}
 \text{FV}(\lambda y. P[N/x]) &= \\
 &= \text{FV}(\lambda y. P[N/x]) && \text{by (4)} \\
 &= \text{FV}(P[N/x]) \setminus \{y\} && \text{by Definição 42.10(2)} \\
 &= \text{FV}(P) \setminus \{y\} && \text{by inductive hypothesis} \\
 &= \text{FV}(\lambda y. P) && \text{by Definição 42.10(2)}
 \end{aligned}$$

Problema 42.7. Complete the proof of **Teorema 42.14**.

Teorema 42.15. *If $x \in \text{FV}(M)$, then $\text{FV}(M[N/x]) = (\text{FV}(M) \setminus \{x\}) \cup \text{FV}(N)$, provided the left hand is defined.*

lam:syn:sub:
thm:info

42.5. SUBSTITUTION

Prova. By induction on the formation of M .

1. M is a variable: exercise.
2. M is of the form PQ : Since $(PQ)[N/y]$ is defined, it has to be $(P[N/x])(Q[N/x])$ with both substitution defined. Also, since $x \in \text{FV}(PQ)$, either $x \in \text{FV}(P)$ or $x \in \text{FV}(Q)$ or both. The rest is left as an exercise.
3. M is of the form $\lambda y. P$. Since $\lambda y. P[N/x]$ is defined, it has to be $\lambda y. P[N/x]$, with $P[N/x]$ defined, $x \neq y$ and $y \notin \text{FV}(N)$; also, since $y \in \text{FV}(\lambda x. P)$, we have $y \in \text{FV}(P)$ too. Now:

$$\begin{aligned}
 \text{FV}((\lambda y. P)[N/x]) &= \\
 &= \text{FV}(\lambda y. P[N/x]) \\
 &= \text{FV}(P[N/x]) \setminus \{y\} \\
 &= ((\text{FV}(P) \setminus \{y\}) \cup (\text{FV}(N) \setminus \{x\})) \quad \text{by inductive hypothesis} \quad \square \\
 &= (\text{FV}(P) \setminus \{x, y\}) \cup \text{FV}(N) \quad x \notin \text{FV}(N) \\
 &= (\text{FV}(\lambda y. P) \setminus \{x\}) \cup \text{FV}(N)
 \end{aligned}$$

Problema 42.8. Complete the proof of **Teorema 42.15**.

lam:syn:sub: thm:ctr **Teorema 42.16.** $x \notin \text{FV}(M[N/x])$, if the right-hand side is defined and $x \notin \text{FV}(N)$.

Prova. Exercise. $\square$

Problema 42.9. Prove **Teorema 42.16**.

lam:syn:sub: thm:inv **Teorema 42.17.** If $M[y/x]$ is defined and $y \notin \text{FV}(M)$, then $M[y/x][x/y] = M$.

Prova. By induction on the formation of M .

1. M is a variable z : Exercise.
2. M is of the form (PQ) . Then:

$$\begin{aligned}
 (PQ)[y/x][x/y] &= ((P[y/x])(Q[y/x]))[x/y] \\
 &= (P[y/x][x/y])(Q[y/x][x/y]) \\
 &= (PQ) \text{ by inductive hypothesis}
 \end{aligned}$$

3. M is of the form $\lambda z. N$. Because $\lambda z. N[y/x]$ is defined, we know that $z \neq y$. So:

$$\begin{aligned}
 (\lambda z. N)[y/x][x/y] &= (\lambda z. N[y/x])[x/y] \\
 &= \lambda z. N[y/x][x/y] \\
 &= \lambda z. N \text{ by inductive hypothesis} \quad \square
 \end{aligned}$$

Problema 42.10. Complete the proof of [Teorema 42.17](#).

-NoValue-

[content/lambda-calculus/syntax/alpha.tex](#)

42.6 α -Conversion

What is the relation between $\lambda x. x$ and $\lambda y. y$? They both represent the identity function. They are, of course, syntactically different terms. They differ only in the name of the bound variable, and one is the result of “renaming” the bound variable in the other. This is called α -conversion.

lam:syn:alp:
sec

Definição 42.18 (Change of bound variable, α). If a term M contains an occurrence of $\lambda x. N$, $y \notin \text{FV}(N)$, and $N[y/x]$ is defined, then replacing this occurrence by

$$\lambda y. N[y/x]$$

resulting in M' is called a *change of bound variable*, written as $M \xrightarrow{\alpha} M'$.

Definição 42.19 (Compatibility of relation). A relation R on terms is said to be *compatible* if it satisfies following conditions:

1. If RNN' then $R\lambda x. N\lambda x. N'$
2. If RPP' then $R(PQ)(P'Q)$
3. If RQQ' then $R(PQ)(PQ')$

Thus let's rephrase the definition:

Definição 42.20 (Change of bound variable, α). *Change of bound variable* ($\xrightarrow{\alpha}$) is the smallest compatible relation on terms satisfying following condition:

$$\lambda x. N \xrightarrow{\alpha} \lambda y. N[y/x] \quad \begin{array}{l} \text{if } x \neq y, y \notin \text{FV}(N) \\ \text{and } N[y/x] \text{ is defined} \end{array}$$

“Smallest” here means the relation contains only pairs that are required by compatibility and the additional condition, and nothing else. Thus this relation can also be defined as follows:

Definição 42.21 (Change of bound variable, α). *Change of bound variable* ($\xrightarrow{\alpha}$) is inductively defined as follows:

lam:syn:alp:
defn:aconvone

1. If $N \xrightarrow{\alpha} N'$ then $\lambda x. N \xrightarrow{\alpha} \lambda x. N'$
2. If $P \xrightarrow{\alpha} P'$ then $(PQ) \xrightarrow{\alpha} (P'Q)$
3. If $Q \xrightarrow{\alpha} Q'$ then $(PQ) \xrightarrow{\alpha} (PQ')$

lam:syn:alp:
defn:aconvone1

lam:syn:alp:
defn:aconvone2

lam:syn:alp:
defn:aconvone3

42.6. α -CONVERSION

lam:syn:alp:
defn:aconvone4

4. If $x \neq y$, $y \notin \text{FV}(N)$ and $N[y/x]$ is defined, then $\lambda x. N \xrightarrow{\alpha} \lambda y. N[y/x]$.

The definitions are equivalent, but we leave the proof as an exercise. From now on we will use the inductive definition.

Definição 42.22 (α -conversion, $\xrightarrow{\alpha}$). α -conversion ($\xrightarrow{\alpha}$) is the smallest reflexive and transitive relation on terms containing $\xrightarrow{\alpha}$.

As above, “smallest” means the relation only contains pairs required by transitivity, and $\xrightarrow{\alpha}$, which leads to the following equivalent definition:

lam:syn:alp:
defn:aconv

Definição 42.23 (α -conversion, $\xrightarrow{\alpha}$). α -conversion ($\xrightarrow{\alpha}$) is inductively defined as follows:

lam:syn:alp:
defn:aconv1

1. If $P \xrightarrow{\alpha} Q$ and $Q \xrightarrow{\alpha} R$, then $P \xrightarrow{\alpha} R$.

lam:syn:alp:
defn:aconv2

2. If $P \xrightarrow{\alpha} Q$, then $P \xrightarrow{\alpha} Q$.

lam:syn:alp:
defn:aconv3

3. $P \xrightarrow{\alpha} P$.

Exemplo 42.24. $\lambda x. fx$ α -converts to $\lambda y. fy$, and conversely. Informally speaking, they are both functions that accept an argument and return f of that argument, referring to the environment variable f .

$\lambda x. fx$ does not α -convert to $\lambda x. gx$. Informally speaking, they refer to the environment variables f and g respectively, and this makes them different functions: they behave differently in environments where f and g are different.

Problema 42.11. Are the following pairs of terms α -convertible?

1. $\lambda x. \lambda y. x$ and $\lambda y. \lambda x. y$
2. $\lambda x. \lambda y. x$ and $\lambda c. \lambda b. a$
3. $\lambda x. \lambda y. x$ and $\lambda c. \lambda b. a$

lam:syn:alp:
lem:fv-one

Lema 42.25. If $P \xrightarrow{\alpha} Q$ then $\text{FV}(P) = \text{FV}(Q)$.

Prova. By induction on the *derivação* of $P \xrightarrow{\alpha} Q$.

1. If the last rule is (4), then P is of the form $\lambda x. N$ and Q of the form $\lambda y. N[y/x]$, with $x \neq y$, $y \notin \text{FV}(N)$ and $N[y/x]$ defined. We distinguish cases according to whether $x \in \text{FV}(N)$:

- a) If $x \in \text{FV}(N)$, then:

$$\begin{aligned} \text{FV}(\lambda y. N[y/x]) &= \text{FV}(N[y/x]) \setminus \{y\} \\ &= ((\text{FV}(N) \setminus \{x\}) \cup \{y\}) \setminus \{y\} \quad \text{by Teorema 42.15} \\ &= \text{FV}(N) \setminus \{x\} \\ &= \text{FV}(\lambda x. N) \end{aligned}$$

b) If $x \notin FV(N)$, then:

$$\begin{aligned} FV(\lambda y. N[y/x]) &= FVN[y/x] \setminus \{y\} \\ &= FV(N) \setminus \{x\} && \text{by Teorema 42.14} \\ &= FV(\lambda x. N). \end{aligned}$$

2. The other three cases are left as exercises. $\square$

Problema 42.12. Complete the proof of [Lema 42.25](#).

Lema 42.26. If $P \xrightarrow{\alpha} Q$ then $Q \xrightarrow{\alpha} P$.

*lam:syn:alp:
lem:inv*

Prova. Induction on the [derivação](#) of $P \xrightarrow{\alpha} Q$.

1. If the last rule is [\(4\)](#), then P is of the form $\lambda x. N$ and Q of the form $\lambda y. N[y/x]$, where $x \neq y$, $y \notin FV(N)$ and $N[y/x]$ defined. First, we have $y \notin FV(N[y/x])$ by [Teorema 42.16](#). By [Teorema 42.17](#) we have that $N[y/x][x/y]$ is not only defined, but also equal to N . Then by [\(4\)](#), we have $\lambda y. N[y/x] \xrightarrow{\alpha} \lambda x. N[y/x][x/y] = \lambda x. N$. $\square$

Problema 42.13. Complete the proof of [Lema 42.26](#)

Teorema 42.27. α -Conversion is an equivalence relation on terms, *O.s.*, it is reflexive, symmetric, and transitive.

- Prova.*
1. For each term M , M can be changed to M by zero changes of bound variables.
 2. If P is α -converts to Q by a series of changes of bound variables, then from Q we can just inverse these changes (by [Lema 42.26](#)) in opposite order to obtain P .
 3. If P α -converts to Q by a series of changes of bound variables, and Q to R by another series, then we can change P to R by first applying the first series and then the second series. $\square$

From now on we say that M and N are α -equivalent, $M \stackrel{\alpha}{\equiv} N$, iff M α -converts to N (which, as we've just shown, is the case iff N α -converts to M).

Teorema 42.28. If $M \stackrel{\alpha}{\equiv} N$, then $FV(M) = FV(N)$.

*lam:syn:alp:
thm:fv*

Prova. Immediate from [Lema 42.25](#). $\square$

Lema 42.29. If $R \stackrel{\alpha}{\equiv} R'$ and $M[R/y]$ is defined, then $M[R'/y]$ is defined and α -equivalent to $M[R/y]$.

*lam:syn:alp:
lem:sub:R*

Prova. Exercise. $\square$

42.6. α -CONVERSION

Problema 42.14. Prove [Lema 42.29](#).

Recall that in [Seção 42.5](#), substitution is undefined in some cases; however, using α -conversion on terms, we can make substitution always defined by renaming bound variables. The result preserves α -equivalence, as shown in this theorem:

lam:syn:alp: **Teorema 42.30.** *thm:sub* For any M , R , and y , there exists M' such that $M \stackrel{\alpha}{=} M'$ and $M'[R/y]$ is defined. Moreover, if there is another pair $M'' \stackrel{\alpha}{=} M$ and R'' where $M''[R''/y]$ is defined and $R'' \stackrel{\alpha}{=} R$, then $M'[R/y] \stackrel{\alpha}{=} M''[R''/y]$.

Prova. By induction on the formation of M :

1. M is a variable z : Exercise.
2. Suppose M is of the form $\lambda x. N$. Select a variable z other than x and y and such that $z \notin \text{FV}(N)$ and $z \notin \text{FV}(R)$. By inductive hypothesis, we there is N' such that $N' \stackrel{\alpha}{=} N$ and $N'[z/x]$ is defined. Then $\lambda x. N \stackrel{\alpha}{=} \lambda x. N'$ too, by [Definição 42.21\(1\)](#). Now $\lambda x. N' \stackrel{\alpha}{=} \lambda z. N'[z/x]$ by [Definição 42.21\(4\)](#). We can do this because $z \neq x$, $z \notin \text{FV}(N')$ and $N'[z/x]$ is defined. Finally, $\lambda z. N'[z/x][R/y]$ is defined, because $z \neq y$ and $z \notin \text{FV}(R)$.

Moreover, if there is another N'' and R'' satisfying the same conditions,

$$\begin{aligned}
 (\lambda z. N''[z/x])[R''/y] &= \\
 &= \lambda z. N''[z/x][R''/y] \\
 &= \lambda z. N''[z/x][R/y] && \text{by [Lema 42.29](#)} \\
 &= \lambda z. N'[z/x][R/y] && \text{by inductive hypothesis} \\
 &= (\lambda z. N'[z/x])[R/y]
 \end{aligned}$$

3. M is of the form (PQ) : Exercise. □

Problema 42.15. Complete the proof of [Teorema 42.30](#).

lam:syn:alp: **Corolário 42.31.** *cor:sub* For any M , R , and y , there exists a pair of M' and R' such that $M' \stackrel{\alpha}{=} M$, $R \stackrel{\alpha}{=} R'$ and $M'[R'/y]$ is defined. Moreover, if there is another pair $M'' \stackrel{\alpha}{=} M$ and R'' with $M'[R'/y]$ defined, then $M'[R'/y] \stackrel{\alpha}{=} M''[R''/y]$.

Prova. Immediate from [Teorema 42.30](#). □

-NoValue-

42.7 The De Bruijn Index

α -Equivalence is very natural, as terms that are α -equivalent “mean the same.” In fact, it is possible to give a syntax for lambda terms which does not distinguish terms that can be α -converted to each other. The best known replaces variables by their *De Bruijn index*.

lam:syn:deb:
sec

When we write $\lambda x. M$, we explicitly state that x is the parameter of the function, so that we can use x in M to refer to this parameter. In the de Bruijn index, however, parameters have no name and reference to them in the function body is denoted by a number denoting the levels of abstraction between them. For example, consider the example of $\lambda x. \lambda y. yx$: the outer abstraction is on binds the variable x ; the inner abstraction binds the variable is y ; the sub-term yx lies in the scope of the inner abstraction: there is no abstraction between y and its abstract λy , but one abstract between x and its abstract λx . Thus we write 0 1 for yx , and $\lambda. \lambda. 01$ for the entire term.

Definição 42.32. De Bruijn terms are inductively defines as follows:

1. n , where n is any natural number.
2. PQ , where P and Q are both De Bruijn terms.
3. $\lambda. N$, where N is a De Bruijn term.

A formalized translation from ordinary lambda terms to De Bruijn indexed terms is as follows:

Definição 42.33.

$$\begin{aligned} F_\Gamma(x) &= \Gamma(x) \\ F_\Gamma(PQ) &= F_\Gamma(P)F_\Gamma(Q) \\ F_\Gamma(\lambda x. N) &= \lambda. F_{x,\Gamma}(N) \end{aligned}$$

where Γ is a list of variables indexed from zero, and $\Gamma(x)$ denotes the position of the variable x in Γ . For example, if Γ is x, y, z , then $\Gamma(x)$ is 0 and $\Gamma(z)$ is 2.

x, Γ denotes the list resulted from pushing x to the head of Γ ; for instance, continuing the Γ in last example, w, Γ is w, x, y, z .

Recovering a standard lambda term from a de Bruijn term is done as follows:

Definição 42.34.

$$\begin{aligned} G_\Gamma(n) &= \Gamma[n] \\ G_\Gamma(PQ) &= G_\Gamma(P)G_\Gamma(Q) \\ G_\Gamma(\lambda. N) &= \lambda x. G_{x,\Gamma}(N) \end{aligned}$$

where Γ is again a list of variables indexed from zero, and $\Gamma[n]$ denotes the variable in position n . For example, if Γ is x, y, z , then $\Gamma[1]$ is y .

The variable x in last equation is chosen to be any variable that not in Γ .

42.8. TERMS AS α -EQUIVALENCE CLASSES

Here we give some results without proving them:

Proposição 42.35. *If $M \xrightarrow{\alpha} M'$, and Γ is any list containing $FV(M)$, then $F_\Gamma(M) \equiv F_\Gamma(M')$.*

-NoValue-

content/lambda-calculus/syntax/term-revisited.tex

42.8 Terms as α -Equivalence Classes

From now on, we will consider terms up to α -equivalence. That means when we write a term, we mean its α -equivalence class it is in. For example, we write $\lambda a. \lambda b. ac$ for the set of all terms α -equivalent to it, such as $\lambda a. \lambda b. ac$, $\lambda b. \lambda a. bc$, etc.

Also, while in previous sections letters such as N, Q are used to denote a term, from now on we use them to denote a class, and it is these classes instead of terms that will be our subjects of study in what follows. Letters such as x, y continues to denote a variable.

We also adopt the notation $\underline{M}$ to denote an arbitrary **membro** of the class M , and $\underline{M}_0, \underline{M}_1$, etc. if we need more than one.

We reuse the notations from terms to simplify our wording. We have following definition on classes:

Definição 42.36. 1. $\lambda x. N$ is defined as the class containing $\lambda x. \underline{N}$.

2. PQ is defined to be the class containing $\underline{P}\underline{Q}$.

It is not hard to see that they are well defined, because α -conversion is compatible.

Definição 42.37. The *free variables* of an α -equivalence class M , or $FV(M)$, is defined to be $FV(\underline{M})$.

This is well defined since $FV(\underline{M}_0) = FV(\underline{M}_1)$, as shown in **Teorema 42.28**.

We also reuse the notation for substitution into classes:

Definição 42.38. The *substitution* of R for y in M , or $M[R/y]$, is defined to be $\underline{M}_{\underline{R}/y}$, for any $\underline{M}$ and $\underline{R}$ making the substitution defined.

This is also well defined as shown in **Corolário 42.31**.

Note how this definition significantly simplifies our reasoning. For example:

$$\lambda x. x[y/x] = \quad (42.1)$$

$$= \lambda z. z[y/x] \quad (42.2)$$

$$= \lambda z. z[y/x] \quad (42.3)$$

$$= \lambda z. z \quad (42.4)$$

Eq. (42.1) is undefined if we still regard it as substitution on terms; but as mentioned earlier, we now consider it a substitution on classes, which is why Eq. (42.2) can happen: we can replace $\lambda x. x$ with $\lambda z. z$ because they belong to the same class.

For the same reason, from now on we will assume that the representatives we choose always satisfy the conditions needed for substitution. For example, when we see $\lambda x. N[R/y]$, we will assume the representative $\lambda x. N$ is chosen so that $x \neq y$ and $x \notin FV(R)$.

Since it is a bit strange to call $\lambda x. x$ a “class”, let’s call them Λ -terms (or simply “terms” in the rest of the part) from now on, to distinguish them from λ -terms that we are familiar with.

We cannot say goodbye to terms yet: the whole definition of Λ -terms is based on λ -terms, and we haven’t provided a method to define functions on Λ -terms, which means all such functions have to be first defined on λ -terms, and then “projected” to Λ -terms, as we did for substitutions. However we assume the reader can intuitively understand how we can define functions on Λ -terms.

-NoValue-

<content/lambda-calculus/syntax/beta.tex>

42.9 β -reduction

When we see $(\lambda m. (\lambda y. y)m)$, it is natural to conjecture that it has some connection with $\lambda m. m$, namely the second term should be the result of “simplifying” the first. The notion of β -reduction captures this intuition formally.

lam:int:bet:
sec

Definição 42.39 (β -contraction, $\xrightarrow{\beta}$). The β -contraction ($\xrightarrow{\beta}$) is the smallest compatible relation on terms satisfying the following condition:

lam:int:bet:
defn:betacontr

$$(\lambda x. N)Q \xrightarrow{\beta} N[Q/x]$$

We say P is β -contracted to Q if $P \xrightarrow{\beta} Q$. A term of the form $(\lambda x. N)Q$ is called a *redex*.

Problema 42.16. Spell out the equivalent inductive definitions of β -contraction as we did for change of bound variable in Definição 42.21.

lam:int:bet:
prob:def

Definição 42.40 (β -reduction, $\xrightarrow{\beta\Rightarrow}$). β -reduction ($\xrightarrow{\beta\Rightarrow}$) is the smallest reflexive, transitive relation on terms containing $\xrightarrow{\beta}$. We say P is β -reduced to Q if $P \xrightarrow{\beta\Rightarrow} Q$.

lam:int:bet:
defn:betared

42.9. β -REDUCTION

We will write $\rightarrow$ instead of $\xrightarrow{\beta}$, and $\twoheadrightarrow$ instead of $\xrightarrow{\beta}$ when context is clear.

Informally speaking, $M \xrightarrow{\beta} N$ if and only if M can be changed to N by zero or several steps of β -contraction.

Definição 42.41 (β -normal). A term that cannot be β -contracted any further is said to be β -normal.

If $M \xrightarrow{\beta} N$ and N is β -normal, then we say N is a *normal form* of M . One may ask if the normal form of a term is unique, and the answer is yes, as we will see later.

Let us consider some examples.

1. We have

$$\begin{aligned} (\lambda x. xxy)\lambda z. z &\rightarrow (\lambda z. z)(\lambda z. z)y \\ &\rightarrow (\lambda z. z)y \\ &\rightarrow y \end{aligned}$$

2. “Simplifying” a term can actually make it more complex:

$$\begin{aligned} (\lambda x. xxy)(\lambda x. xxy) &\rightarrow (\lambda x. xxy)(\lambda x. xxy)y \\ &\rightarrow (\lambda x. xxy)(\lambda x. xxy)yy \\ &\rightarrow \dots \end{aligned}$$

3. It can also leave a term unchanged:

$$(\lambda x. xx)(\lambda x. xx) \rightarrow (\lambda x. xx)(\lambda x. xx)$$

4. Also, some terms can be reduced in more than one way; for example,

$$(\lambda x. (\lambda y. yx)z)v \rightarrow (\lambda y. yv)z$$

by contracting the outermost application; and

$$(\lambda x. (\lambda y. yx)z)v \rightarrow (\lambda x. zx)v$$

by contracting the innermost one. Note, in this case, however, that both terms further reduce to the same term, zv .

The final outcome in the last example is not a coincidence, but rather illustrates a deep and important property of the lambda calculus, known as the Church–Rosser property.

In general, there is more than one way to β -reduce a term, thus many reduction strategies have been invented, among which the most common is the *natural strategy*. The natural strategy always contracts the *left-most* redex, where the position of a redex is defined as its starting point in the term. The natural strategy has the useful property that a term can be reduced to a normal form by some strategy iff it can be reduced to normal form using the natural strategy. In what follows we will use the natural strategy unless otherwise specified.

Definição 42.42 (β -equivalence, $=$). β -Equivalence ($=$) is the relation inductively defined as follows:

1. $M = M$.
2. If $M = N$, then $N = M$.
3. If $M = N$, $N = O$, then $M = O$.
4. If $M = N$, then $PM = PN$.
5. If $M = N$, then $MQ = NQ$.
6. If $M = N$, then $\lambda x. M = \lambda x. N$.
7. $(\lambda x. N)Q = N[Q/x]$.

The first three rules make the relation an equivalence relation; the next three make it compatible; the last ensures that it contains β -contraction.

Informally speaking, two terms are β -equivalent if and only if one of them can be changed to the other in zero or more steps of β -contraction, or “inverse” of β -contraction. The inverse of β -contraction is defined so that M inverse- β -contracts to N iff N β -contracts to M .

Besides the above rules, we will extend the relation with more rules, and denote the extended equivalence relation as $\stackrel{X}{=}$, where X is the extending rule.

-NoValue-

[content/lambda-calculus/syntax/eta.tex](#)

42.10 η -conversion

There is another relation on λ terms. In [Seção 42.4](#) we used the example $\lambda x. (fx)$, which accepts an argument and applies f to it. In other words, it is the same function as f : $\lambda x. (fx)N$ and fN both reduce to fN . We use η -reduction (and η -extension) to capture this idea.

lam:syn:eta:
sec

Definição 42.43 (η -contraction, $\stackrel{\eta}{\rightarrow}$). η -contraction ($\stackrel{\eta}{\rightarrow}$) is the smallest compatible relation on terms satisfying the following condition:

lam:syn:eta:
defn:beredone

$$\lambda x. Mx \stackrel{\eta}{\rightarrow} M \text{ provided } x \notin FV(M)$$

Definição 42.44 ($\beta\eta$ -reduction, $\stackrel{\beta\eta}{\rightarrow}$). $\beta\eta$ -reduction ($\stackrel{\beta\eta}{\rightarrow}$) is the smallest reflexive, transitive relation on terms containing $\stackrel{\beta}{\rightarrow}$ and $\stackrel{\eta}{\rightarrow}$, O.s., the rules of reflexivity and transitive plus the following two rules:

lam:syn:eta:
defn:bered

1. If $M \stackrel{\beta}{\rightarrow} N$ then $M \stackrel{\beta\eta}{\rightarrow} N$.
2. If $M \stackrel{\eta}{\rightarrow} N$ then $M \stackrel{\beta\eta}{\rightarrow} N$.

lam:syn:eta:
defn:bered3

lam:syn:eta:
defn:bered4

Definição 42.45. We extend the equivalence relation $=$ with the η -conversion rule:

$$\lambda x. fx = f$$

and denote the extended relation as $\stackrel{\eta}{=}$.

η -equivalence is important because it is related to extensionality of lambda terms:

Definição 42.46 (Extensionality). We extend the equivalence relation $=$ with the (ext) rule:

$$\text{If } Mx = Nx \text{ then } M = N, \text{ provided } x \notin FV(MN).$$

and denote the extended relation as $\stackrel{ext}{=}$.

Roughly speaking, the rule states that two terms, viewed as functions, should be considered equal if they behave the same for the same argument.

We now prove that the η rule provides exactly the extensionality, and nothing else.

Teorema 42.47. $M \stackrel{ext}{=} N$ if and only if $M \stackrel{\eta}{=} N$.

Prova. First we prove that $\stackrel{\eta}{=}$ is closed under the extensionality rule. That is, ext rule doesn't add anything to $\stackrel{\eta}{=}$. We then have $\stackrel{\eta}{=}$ contains $\stackrel{ext}{=}$, and if $M \stackrel{ext}{=} N$, then $M \stackrel{\eta}{=} N$.

To prove $\stackrel{\eta}{=}$ is closed under ext , note that for any $M = N$ derived by the ext rule, we have $Mx \stackrel{\eta}{=} Nx$ as premise. Then we have $\lambda x. Mx \stackrel{\eta}{=} \lambda x. Nx$ by a rule of $=$, applying η on both side gives us $M \stackrel{\eta}{=} N$.

Similarly we prove that the η rule is contained in $\stackrel{ext}{=}$. For any $\lambda x. Mx$ and M with $x \notin FV(M)$, we have that $(\lambda x. Mx)x \stackrel{ext}{=} Mx$, giving us $\lambda x. Mx \stackrel{ext}{=} M$ by the ext rule. $\square$

Capítulo 43

The Church–Rosser Property

-NoValue-

content/lambda-calculus/church-rosser/definitions-and-properties.tex

43.1 Definition and Properties

In this chapter we introduce the concept of Church–Rosser property and some common properties of this property. lam:cr:dap:
sec

Definição 43.1 (Church–Rosser property, CR). A relation $\xrightarrow{X}$ on terms is said to satisfy the *Church–Rosser property* iff, whenever $M \xrightarrow{X} P$ and $M \xrightarrow{X} Q$, then there exists some N such that $P \xrightarrow{X} N$ and $Q \xrightarrow{X} N$.

We can view the lambda calculus as a model of computation in which terms in normal form are “values” and a reducibility relation on terms are the “calculation rules.” The Church–Rosser property states is that when there is more than one way to proceed with a calculation, there is still only a single value of the expression.

To take an example from elementary algebra, there’s more than one way to calculate $4 \times (1 + 2) + 3$. It can either be reduced to $4 \times 3 + 3$ (if we first reduce $1 + 2$ to 3) or to $4 \times 1 + 4 \times 2 + 3$ (if we first reduce $4 \times (1 + 2)$ using distributivity). Both of these, however, can be further reduced to $12 + 3$.

If we take $\xrightarrow{X}$ to be β -reduction, we easily see that a consequence of the Church–Rosser property is that if a term has a normal form, then it is unique. For suppose M can be reduced to P and Q , both of which are normal forms. By the Church–Rosser property, there exists some N such that both P and Q reduce to it. Since by assumption P and Q are normal forms, the reduction of P and Q to N can only be the trivial reduction, O.s., P , Q , and N are identical. This justifies our speaking of *the* normal form of a term.

In viewing the lambda calculus as a model of computation, then, the normal form of a term can be thought of as the “final result” of the computation starting with that term. The above corollary means there’s only one, if any, final result of a computation, just like there is only one result of computing $4 \times (1 + 2) + 3$, namely 15.

Teorema 43.2. *If a relation $\xrightarrow{X}$ satisfies the Church–Rosser property, and $\xrightarrow{X}$ is the smallest transitive relation containing $\xrightarrow{X}$, then $\xrightarrow{X}$ satisfies the Church–Rosser property too.* lam:cr:dap:
thm:str

Prova. Suppose

$$\begin{aligned} M &\xrightarrow{X} P_1 \xrightarrow{X} \dots \xrightarrow{X} P_m \text{ and} \\ M &\xrightarrow{X} Q_1 \xrightarrow{X} \dots \xrightarrow{X} Q_n. \end{aligned}$$

We will prove the theorem by constructing a grid N of terms of height is $m + 1$ and width $n + 1$. We use $N_{i,j}$ to denote the term in the i -th row and j -th column.

43.2. PARALLEL β -REDUCTION

We construct N in such a way that $N_{i,j} \xrightarrow{X} N_{i+1,j}$ and $N_{i,j} \xrightarrow{X} N_{i,j+1}$. It is defined as follows:

$$\begin{aligned} N_{0,0} &= M \\ N_{i,0} &= P_i && \text{if } 1 \leq i \leq m \\ N_{0,j} &= Q_j && \text{if } 1 \leq j \leq n \end{aligned}$$

and otherwise:

$$N_{i,j} = R$$

where R is a term such that $N_{i-1,j} \xrightarrow{X} R$ and $N_{i,j-1} \xrightarrow{X} R$. By the Church–Rosser property of $\xrightarrow{X}$, such a term always exists.

Now we have $N_{m,0} \xrightarrow{X} \dots \xrightarrow{X} N_{m,n}$ and $N_{0,n} \xrightarrow{X} \dots \xrightarrow{X} N_{m,n}$. Note $N_{m,0}$ is P and $N_{0,n}$ is Q . By definition of $\xrightarrow{X}$ the theorem follows. $\square$

-NoValue-

[content/lambda-calculus/church-rosser/parallel-beta-reduction.tex](#)

43.2 Parallel β -reduction

lam:cr:pb: sec We introduce the notion of *parallel β -reduction*, and prove the it has the Church–Rosser property.

lam:cr:pb: defn:bredpar **Definição 43.3 (parallel β -reduction, $\Rightarrow^\beta$).** Parallel reduction ($\Rightarrow^\beta$) of terms is inductively defined as follows:

- lam:cr:pb: defn:bredpar1 1. $x \Rightarrow^\beta x$.
- lam:cr:pb: defn:bredpar2 2. If $N \Rightarrow^\beta N'$ then $\lambda x. N \Rightarrow^\beta \lambda x. N'$.
- lam:cr:pb: defn:bredpar3 3. If $P \Rightarrow^\beta P'$ and $Q \Rightarrow^\beta Q'$ then $PQ \Rightarrow^\beta P'Q'$.
- lam:cr:pb: defn:bredpar4 4. If $N \Rightarrow^\beta N'$ and $Q \Rightarrow^\beta Q'$ then $(\lambda x. N)Q \Rightarrow^\beta N'[Q'/x]$.

Parallel β -reduction allows us to reduce any number of redices in a term in one step. It is different from β -reduction in the sense that we can only contract redices that occur in the original term, but not redices arising from parallel β -reduction. For example, the term $(\lambda f. fx)(\lambda y. y)$ can only be parallel β -reduced to itself or to $(\lambda y. y)x$, but not further to x , although it β -reduces to x , because this redex arises only after one step of parallel β -reduction. A second parallel β -reduction step yields x , though.

lam:cr:pb: thm:refl **Teorema 43.4.** $M \Rightarrow^\beta M$.

Prova. Exercise. □

Problema 43.1. Prove [Teorema 43.4](#).

Definição 43.5 (β -complete development). The β -complete development $M^{*\beta}$ of M is defined inductively as follows: lam:cr:pb:
defn:bcd

$$x^{*\beta} = x \quad (43.1) \quad \text{lam:cr:pb:}$$

$$(\lambda x. N)^{*\beta} = \lambda x. N^{*\beta} \quad (43.2) \quad \text{defn:bcd1}$$

$$(PQ)^{*\beta} = P^{*\beta}Q^{*\beta} \quad \text{if } P \text{ is not a } \lambda\text{-abstract} \quad (43.3) \quad \text{lam:cr:pb:}$$

$$((\lambda x. N)Q)^{*\beta} = N^{*\beta}[Q^{*\beta}/x] \quad (43.4) \quad \text{defn:bcd2}$$

The β -complete development of a term, as its name suggests, is a “complete parallel reduction.” While for parallel β -reduction we still can choose to not contract a redex, for complete development we have no choice but to contract all of them. Thus the complete development of $(\lambda f. fx)(\lambda y. y)$ is $(\lambda y. y)x$, not itself. defn:bcd3
lam:cr:pb:
defn:bcd4

This definition has the problem that we haven’t introduced how to define functions on (λ) -terms recursively. Will fix in future.

Lema 43.6. If $M \xRightarrow{\beta} M'$ and $R \xRightarrow{\beta} R'$, then $M[R/y] \xRightarrow{\beta} M'[R'/y]$. lam:cr:pb:
lem:comp

Prova. By induction on the [derivação](#) of $M \xRightarrow{\beta} M'$.

1. The last step is [\(1\)](#): Exercise.
2. The last step is [\(2\)](#): Then M is $\lambda x. N$ and M' is $\lambda x. N'$, where $N \xRightarrow{\beta} N'$.
We want to prove that $(\lambda x. N)[R/y] \xRightarrow{\beta} (\lambda x. N')[R'/y]$, O.s., $\lambda x. N[R/y] \xRightarrow{\beta} \lambda x. N'[R'/y]$. This follows immediately by [\(2\)](#) and the induction hypothesis.
3. The last step is [\(3\)](#): Exercise.
4. The last step is [\(4\)](#): M is $(\lambda x. N)Q$ and M' is $N'[Q'/x]$. We want to prove that $((\lambda x. N)Q)[R/y] \xRightarrow{\beta} N'[Q'/x][R'/y]$, O.s., $(\lambda x. N[R/y])Q[R/y] \xRightarrow{\beta} N'[R'/y][Q'[R'/y]/x]$. This follows by [\(4\)](#) and the induction hypothesis. □

Problema 43.2. Complete the proof of [Lema 43.6](#).

Lema 43.7. If $M \xRightarrow{\beta} M'$ then $M' \xRightarrow{\beta} M^{*\beta}$. lam:cr:pb:
lem:cont

Prova. By induction on the [derivação](#) of $M \xRightarrow{\beta} M'$.

43.3. β -REDUCTION

1. The last rule is (1): Exercise.
2. The last rule is (2): M is $\lambda x. N$ and M' is $\lambda x. N'$ with $N \xRightarrow{\beta} N'$. We want to show that $\lambda x. N' \xRightarrow{\beta} (\lambda x. N)^{* \beta}$, O.s., $\lambda x. N' \xRightarrow{\beta} \lambda x. N^{* \beta}$ by Eq. (43.2). It follows by (2) and the induction hypothesis.
3. The last rule is (3): M is PQ and M' is $P'Q'$ for some P, Q, P' and Q' , with $P \xRightarrow{\beta} P'$ and $Q \xRightarrow{\beta} Q'$. By induction hypothesis, we have $P' \xRightarrow{\beta} P^{* \beta}$ and $Q' \xRightarrow{\beta} Q^{* \beta}$.
 - a) If P is $\lambda x. N$ for some x and N , then P' must be $\lambda x. N'$ for some N' with $N \xRightarrow{\beta} N'$. By induction hypothesis we have $N' \xRightarrow{\beta} N^{* \beta}$ and $Q' \xRightarrow{\beta} Q^{* \beta}$. Then $(\lambda x. N')Q' \xRightarrow{\beta} N^{* \beta}[Q^{* \beta}/x]$ by (4).
 - b) If P is not a λ -abstract, then $P'Q' \xRightarrow{\beta} P^{* \beta}Q^{* \beta}$ by (3), and the right-hand side is $PQ^{* \beta}$ by Eq. (43.3).
4. The last rule is (4): M is $(\lambda x. N)Q$ and M' is $N'[Q'/x]$ for some x, N, Q, N' , and Q' , with $N \xRightarrow{\beta} N'$ and $Q \xRightarrow{\beta} Q'$. By induction hypothesis we know $N' \xRightarrow{\beta} N^{* \beta}$ and $Q' \xRightarrow{\beta} Q^{* \beta}$. By Lema 43.6 we have $N'[Q'/x] \xRightarrow{\beta} N^{* \beta}[Q^{* \beta}/x]$, the right-hand side of which is exactly $((\lambda x. N)Q)^{* \beta}$. $\square$

Problema 43.3. Complete the proof of Lema 43.7.

lam:cr:pb: **Teorema 43.8.** $\xRightarrow{\beta}$ has the Church–Rosser property.
thm:cr

Prova. Immediate from Lema 43.7. $\square$

-NoValue-

content/lambda-calculus/church-rosser/beta-reduction.tex

43.3 β -reduction

lam:cr:b: **Lema 43.9.** If $M \xrightarrow{\beta} M'$, then $M \xRightarrow{\beta} M'$.
sec
lam:cr:b:
lem:one-par

Prova. If $M \xrightarrow{\beta} M'$, then M is $(\lambda x. N)Q$, M' is $N[Q/x]$, for some x, N , and Q . Since $N \xRightarrow{\beta} N$ and $Q \xRightarrow{\beta} Q$ by Teorema 43.4, we immediately have $(\lambda x. N)Q \xRightarrow{\beta} N[Q/x]$ by Definição 43.3(4). $\square$

lam:cr:b: **Lema 43.10.** If $M \xRightarrow{\beta} M'$, then $M \xrightarrow{\beta} M'$.
lem:par-red

Prova. By induction on the derivação of $M \xRightarrow{\beta} M'$.

1. The last rule is (1): Then M and M' are just x , and $x \xrightarrow{\beta} x$.
2. The last rule is (2): M is $\lambda x. N$ and M' is $\lambda x. N'$ for some x, N, N' , where $N \xRightarrow{\beta} N'$. By induction hypothesis we have $N \xrightarrow{\beta} N'$. Then $\lambda x. N \xrightarrow{\beta} \lambda x. N'$ (by the same series of $\xrightarrow{\beta}$ contractions as $N \xrightarrow{\beta} N'$).
3. The last rule is (3): M is PQ and M' is $P'Q'$ for some P, Q, P', Q' , where $P \xRightarrow{\beta} P'$ and $Q \xRightarrow{\beta} Q'$. By induction hypothesis we have $P \xrightarrow{\beta} P'$ and $Q \xrightarrow{\beta} Q'$. So $PQ \xrightarrow{\beta} P'Q'$ by the reduction sequence $P \xrightarrow{\beta} P'$ followed by the reduction $Q \xrightarrow{\beta} Q'$.
4. The last rule is (4): M is $(\lambda x. N)Q$ and M' is $N'[Q'/x]$ for some x, N, M', Q, Q' , where $N \xRightarrow{\beta} N'$ and $Q \xRightarrow{\beta} Q'$. By induction hypothesis we get $Q \xrightarrow{\beta} Q'$ and $N \xrightarrow{\beta} N'$. So $(\lambda x. N)Q \xrightarrow{\beta} N'[Q'/x]$ by $N \xrightarrow{\beta} N'$ followed by $Q \xrightarrow{\beta} Q'$ and finally contraction of $(\lambda x. N')Q'$ to $N'[Q'/x]$. $\square$

Lema 43.11. $\xrightarrow{\beta}$ is the smallest transitive relation containing $\xRightarrow{\beta}$.

lam:cr:b:
lem:str

Prova. Let $\xrightarrow{X}$ be the smallest transitive relation containing $\xRightarrow{\beta}$.

$\xrightarrow{\beta} \subseteq \xrightarrow{X}$: Suppose $M \xrightarrow{\beta} M'$, O.s., $M \equiv M_1 \xrightarrow{\beta} \dots \xrightarrow{\beta} M_k \equiv M'$. By **Lema 43.9**, $M \equiv M_1 \xRightarrow{\beta} \dots \xRightarrow{\beta} M_k \equiv M'$. Since $\xrightarrow{X}$ contains $\xRightarrow{\beta}$ and is transitive, $M \xrightarrow{X} M'$.

$\xrightarrow{X} \subseteq \xrightarrow{\beta}$: Suppose $M \xrightarrow{X} M'$, O.s., $M \equiv M_1 \xRightarrow{\beta} \dots \xRightarrow{\beta} M_k \equiv M'$. By **Lema 43.10**, $M \equiv M_1 \xrightarrow{\beta} \dots \xrightarrow{\beta} M_k \equiv M'$. Since $\xrightarrow{\beta}$ is transitive, $M \xrightarrow{\beta} M'$. $\square$

Teorema 43.12. $\xrightarrow{\beta}$ satisfies the Church-Rosser property.

lam:cr:b:
thm:cr

Prova. Immediate from **Teorema 43.2**, **Teorema 43.8**, and **Lema 43.11**. $\square$

-NoValue-

content/lambda-calculus/church-rosser/parallel-beta-eta-reduction.tex

43.4 Parallel $\beta\eta$ -reduction

In this section we prove the Church-Rosser property for parallel $\beta\eta$ -reduction, the parallel reduction notion corresponding to $\beta\eta$ -reduction.

lam:cr:pbe:
sec

Definição 43.13 (Parallel $\beta\eta$ -reduction, $\xRightarrow{\beta\eta}$). Parallel $\beta\eta$ -reduction ($\xRightarrow{\beta\eta}$) on terms is inductively defined as follows:

lam:cr:pbe:
defn:beredpar

1. $x \xRightarrow{\beta\eta} x$.

lam:cr:pbe:
defn:beredpar1

43.4. PARALLEL $\beta\eta$ -REDUCTION

2. If $N \xrightarrow{\beta} N'$ then $\lambda x. N \xRightarrow{\beta\eta} \lambda x. N'$.
3. If $P \xRightarrow{\beta\eta} P'$ and $Q \xRightarrow{\beta\eta} Q'$ then $PQ \xRightarrow{\beta\eta} P'Q'$.
4. If $N \xRightarrow{\beta\eta} N'$ and $Q \xRightarrow{\beta\eta} Q'$ then $(\lambda x. N)Q \xRightarrow{\beta\eta} N'[Q'/x]$.
5. If $N \xRightarrow{\beta\eta} N'$ then $\lambda x. Nx \xRightarrow{\beta\eta} N'$, provided $x \notin FV(N)$.

Teorema 43.14. $M \xRightarrow{\beta\eta} M$.

Prova. Exercise. □

Problema 43.4. Prove **Teorema 43.14**.

Definição 43.15 ($\beta\eta$ -complete development). The $\beta\eta$ -complete development $M^{*\beta\eta}$ of M is defined as follows:

$$x^{*\beta\eta} = x \quad (43.5)$$

$$(\lambda x. N)^{*\beta\eta} = \lambda x. N^{*\beta\eta} \quad (43.6)$$

$$(PQ)^{*\beta\eta} = P^{*\beta\eta}Q^{*\beta\eta} \quad \text{if } P \text{ is not a } \lambda\text{-abstract} \quad (43.7)$$

$$((\lambda x. N)Q)^{*\beta\eta} = N^{*\beta\eta}[Q^{*\beta\eta}/x] \quad (43.8)$$

$$(\lambda x. Nx)^{*\beta\eta} = N^{*\beta\eta} \quad \text{if } x \notin FV(N) \quad (43.9)$$

Lema 43.16. If $M \xRightarrow{\beta\eta} M'$ and $R \xRightarrow{\beta\eta} R'$, then $M[R/y] \xRightarrow{\beta\eta} M'[R'/y]$.

Prova. By induction on the **derivação** of $M \xRightarrow{\beta\eta} M'$.

The first four cases are exactly like those in **Lema 43.6**. If the last rule is (5), then M is $\lambda x. Nx$, M' is N' for some x and N' where $x \notin FV(N)$, and $N \xRightarrow{\beta\eta} N'$. We want to show that $(\lambda x. Nx)[R/y] \xRightarrow{\beta\eta} N'[R'/y]$, O.s., $\lambda x. N[R/y]x \xRightarrow{\beta\eta} N'[R'/y]$. It follows by **Definição 43.13(5)** and the induction hypothesis. □

Lema 43.17. If $M \xRightarrow{\beta\eta} M'$ then $M' \xRightarrow{\beta\eta} M^{*\beta\eta}$.

Prova. By induction on the **derivação** of $M \xRightarrow{\beta\eta} M'$.

The first four cases are like those in **Lema 43.7**. If the last rule is (5), then M is $\lambda x. Nx$ and M' is N' for some x , N , N' where $x \notin FV(N)$ and $N \xRightarrow{\beta\eta} N'$. We want to show that $N' \xRightarrow{\beta\eta} (\lambda x. Nx)^{*\beta\eta}$, O.s., $N' \xRightarrow{\beta\eta} N^{*\beta\eta}$, which is immediate by induction hypothesis. □

Teorema 43.18. $\xRightarrow{\beta\eta}$ has the Church-Rosser property.

Prova. Immediate from [Lema 43.17](#). □

-NoValue-

content/lambda-calculus/church-rosser/beta-eta-reduction.tex

43.5 $\beta\eta$ -reduction

The Church–Rosser property holds for $\beta\eta$ -reduction ($\xrightarrow{\beta\eta}$). lam:cr:be:
sec

Lema 43.19. *If $M \xrightarrow{\beta\eta} M'$, then $M \xRightarrow{\beta\eta} M'$.* lam:cr:be:
lem:one-par

Prova. By induction on the [derivação](#) of $M \xrightarrow{\beta\eta} M'$. If $M \xrightarrow{\beta} M'$ by η -conversion (O.s., [Definição 42.43](#)), we use [Teorema 43.14](#). The other cases are as in [Lema 43.9](#). □

Lema 43.20. *If $M \xRightarrow{\beta\eta} M'$, then $M \xrightarrow{\beta\eta} M'$.* lam:cr:be:
lem:par-red

Prova. Induction on the [derivação](#) of $M \xRightarrow{\beta\eta} M'$.

If the last rule is [\(5\)](#), then M is $\lambda x. Nx$ and M' is N' for some x , N , N' where $x \notin FV(N)$ and $N \xRightarrow{\beta\eta} N'$. Thus we can first reduce $\lambda x. Nx$ to N by η -conversion, followed by the series of $\xrightarrow{\beta\eta}$ steps that show that $N \xrightarrow{\beta\eta} N'$, which holds by induction hypothesis. □

Lema 43.21. $\xrightarrow{\beta\eta}$ is the smallest transitive relation containing $\xRightarrow{\beta\eta}$. lam:cr:be:
lem:str

Prova. As in [Lema 43.11](#) □

Teorema 43.22. $\xrightarrow{\beta\eta}$ satisfies Church–Rosser property. lam:cr:be:
thm:cr

Prova. By [Teorema 43.2](#), [Teorema 43.18](#) and [Lema 43.21](#). □

Capítulo 44

Lambda Definability

This chapter is experimental. It needs more explanation, and the material should be structured better into definitions and propositions with proofs, and more examples.

-NoValue-

<content/lambda-calculus/lambda-definability/introduction.tex>

44.1 Introduction

lam:ldf:int:
sec

At first glance, the lambda calculus is just a very abstract calculus of expressions that represent functions and applications of them to others. Nothing in the syntax of the lambda calculus suggests that these are functions of particular kinds of objects, in particular, the syntax includes no mention of natural numbers. Its basic operations—application and lambda abstractions—are operations that apply to any function, not just functions on natural numbers.

Nevertheless, with some ingenuity, it is possible to define arithmetical functions, O.s., functions on the natural numbers, in the lambda calculus. To do this, we define, for each natural number $n \in \mathbb{N}$, a special λ -term $\bar{n}$, the *Church numeral* for n . (Church numerals are named for Alonzo Church.)

Definição 44.1. If $n \in \mathbb{N}$, the corresponding *Church numeral* $\bar{n}$ represents n :

$$\bar{n} \equiv \lambda f x. f^n(x)$$

Here, $f^n(x)$ stands for the result of applying f to x n times. For example, $\bar{0}$ is $\lambda f x. x$, and $\bar{3}$ is $\lambda f x. f(f(f x))$.

The Church numeral $\bar{n}$ is encoded as a lambda term which represents a function accepting two arguments f and x , and returns $f^n(x)$. Church numerals are evidently in normal form.

A representation of natural numbers in the lambda calculus is only useful, of course, if we can compute with them. Computing with Church numerals in the lambda calculus means applying a λ -term F to such a Church numeral, and reducing the combined term $F \bar{n}$ to a normal form. If it always reduces to a normal form, and the normal form is always a Church numeral $\bar{m}$, we can think of the output of the computation as being the number m . We can then think of F as defining a function $f: \mathbb{N} \rightarrow \mathbb{N}$, namely the function such that $f(n) = m$ iff $F \bar{n} \rightarrow \bar{m}$. Because of the Church–Rosser property, normal forms are unique if they exist. So if $F \bar{n} \rightarrow \bar{m}$, there can be no other term in normal form, in particular no other Church numeral, that $F \bar{n}$ reduces to.

Conversely, given a function $f: \mathbb{N} \rightarrow \mathbb{N}$, we can ask if there is a term F that defines f in this way. In that case we say that F λ -*defin**em* f , and that f is λ -*definível*. We can generalize this to many-place and partial functions.

Definição 44.2. Suppose $f: \mathbb{N}^k \rightarrow \mathbb{N}$. We say that a lambda term F λ -*defin**em* f if for all $n_0, \dots, n_{k-1}$,

$$F \bar{n}_0 \bar{n}_1 \dots \bar{n}_{k-1} \rightarrow \overline{f(n_0, n_1, \dots, n_{k-1})}$$

if $f(n_0, \dots, n_{k-1})$ is defined, and $F \bar{n}_0 \bar{n}_1 \dots \bar{n}_{k-1}$ has no normal form otherwise.

A very simple example are the constant functions. The term $C_k \equiv \lambda x. \bar{k}$ λ -*defin**em* the function $c_k: \mathbb{N} \rightarrow \mathbb{N}$ such that $c(n) = k$. For $C_k \bar{n} \equiv (\lambda x. \bar{k}) \bar{n} \rightarrow \bar{k}$ for any n . The identity function is λ -*definido* by $\lambda x. x$. More complex functions are of course harder to define, and often require a lot of ingenuity. So it is perhaps surprising that every computable function is λ -*definível*. The converse is also true: if a function is λ -*definível*, it is computable.

-NoValue-

<content/lambda-calculus/lambda-definability/arithmetical-functions.tex>

44.2 λ -definível Arithmetical Functions

Proposição 44.3. *The successor function succ is λ -definível.*

lam:rep:arf:
sec
lam:rep:arf:
prop:succ-ld

Prova. A term that λ -*defin**em* the successor function is

$$\text{Succ} \equiv \lambda a. \lambda f x. f(afx).$$

Given our conventions, this is short for

$$\text{Succ} \equiv \lambda a. \lambda f. \lambda x. (f((af)x)).$$

Succ is a function that accepts as argument a number a , and evaluates to another function, $\lambda f x. f(afx)$. That function is not itself a Church numeral. However, if the argument a is a Church numeral, it reduces to one. Consider:

$$(\lambda a. \lambda f x. f(afx)) \bar{n} \rightarrow \lambda f x. f(\bar{n}fx).$$

44.2. λ -DEFINÍVEL ARITHMETICAL FUNCTIONS

The embedded term $\bar{n}fx$ is a redex, since $\bar{n}$ is $\lambda fx. f^n x$. So $\bar{n}fx \rightarrow f^n x$ and so, for the entire term we have

$$\text{Succ } \bar{n} \rightarrow \lambda fx. f(f^n(x)),$$

O.s., $\overline{n+1}$. □

Exemplo 44.4. Let's look at what happens when we apply Succ to $\bar{0}$, O.s., $\lambda fx. x$. We'll spell the terms out in full:

$$\begin{aligned} \text{Succ } \bar{0} &\equiv (\lambda a. \lambda f. \lambda x. (f((af)x)))(\lambda f. \lambda x. x) \\ &\rightarrow \lambda f. \lambda x. (f(((\lambda f. \lambda x. x)f)x)) \\ &\rightarrow \lambda f. \lambda x. (f((\lambda x. x)x)) \\ &\rightarrow \lambda f. \lambda x. (fx) \equiv \bar{1} \end{aligned}$$

Problema 44.1. The term

$$\text{Succ}' \equiv \lambda n. \lambda fx. nf(fx)$$

λ -definem the successor function. Explain why.

lam:rep:arf: prop:add-ld **Proposição 44.5.** The addition function add is λ -definível.

Prova. Addition is λ -defined by the terms

$$\text{Add} \equiv \lambda ab. \lambda fx. af(bfx)$$

or, alternatively,

$$\text{Add}' \equiv \lambda ab. a \text{Succ } b.$$

The first addition works as follows: Add first accept two numbers a and b . The result is a function that accepts f and x and returns $af(bfx)$. If a and b are Church numerals $\bar{n}$ and $\bar{m}$, this reduces to $f^{n+m}(x)$, which is identical to $f^n(f^m(x))$. Or, slowly:

$$\begin{aligned} (\lambda ab. \lambda fx. af(bfx))\bar{n}\bar{m} &\rightarrow \lambda fx. \bar{n} f(\bar{m} fx) \\ &\rightarrow \lambda fx. \bar{n} f(f^m x) \\ &\rightarrow \lambda fx. f^n(f^m x) \equiv \overline{n+m}. \end{aligned}$$

The second representation of addition Add' works differently: Applied to two Church numerals $\bar{n}$ and $\bar{m}$,

$$\text{Add}'\bar{n}\bar{m} \rightarrow \bar{n} \text{Succ } \bar{m}.$$

But $\bar{n}fx$ always reduces to $f^n(x)$. So,

$$\bar{n} \text{Succ } \bar{m} \rightarrow \text{Succ}^n(\bar{m}).$$

And since Succ λ -definable the successor function, and the successor function applied n times to m gives $n + m$, this in turn reduces to $\overline{n + m}$. $\square$

Proposição 44.6. *Multiplication is λ -definível by the term*

*lam:rep:arf:
prop:mult-ld*

$$\text{Mult} \equiv \lambda ab. \lambda fx. a(bf)x$$

Prova. To see how this works, suppose we apply Mult to Church numerals $\bar{n}$ and $\bar{m}$: $\text{Mult } \bar{n} \bar{m}$ reduces to $\lambda fx. \bar{n}(\bar{m}f)x$. The term $\bar{m}f$ defines a function which applies f to its argument m times. Consequently, $\bar{n}(\bar{m}f)x$ applies the function “apply f m times” itself n times to x . In other words, we apply f to x , $n \cdot m$ times. But the resulting normal term is just the Church numeral $\overline{nm}$. $\square$

We can actually simplify this term further by η -reduction:

$$\text{Mult} \equiv \lambda ab. \lambda f. a(bf).$$

But then we first have to explain η -reduction.

Problema 44.2. Multiplication can be λ -defined by the term

$$\text{Mult}' \equiv \lambda ab. a(\text{Add } a)\bar{0}.$$

Explain why this works.

The definition of exponentiation as a λ -term is surprisingly simple:

$$\text{Exp} \equiv \lambda be. eb.$$

The first argument b is the base and the second e is the exponent. Intuitively, ef is f^e by our encoding of numbers. If you find it hard to understand, we can still define exponentiation also by iterated multiplication:

$$\text{Exp}' \equiv \lambda be. e(\text{Mult } b)\bar{1}.$$

Predecessor and subtraction on Church numeral is not as simple as we might think: it requires encoding of pairs.

-NoValue-

[content/lambda-calculus/lambda-definability/pairs.tex](#)

44.3. PAIRS AND PREDECESSOR

44.3 Pairs and Predecessor

lam:ldf:pai:
sec

Definição 44.7. The pair of M and N (written $\langle M, N \rangle$) is defined as follows:

$$\langle M, N \rangle \equiv \lambda f. fMN.$$

Intuitively it is a function that accepts a function, and applies that function to the two elements of the pair. Following this idea we have this constructor, which takes two terms and returns the pair containing them:

$$\text{Pair} \equiv \lambda mn. \lambda f. fmn$$

Given a pair, we also want to recover its elements. For this we need two access functions, which accept a pair as argument and return the first or second elements in it:

$$\text{Fst} \equiv \lambda p. p(\lambda mn. m)$$

$$\text{Snd} \equiv \lambda p. p(\lambda mn. n)$$

Problema 44.3. Explain why the access functions Fst and Snd work.

Now with pairs we can λ -define the predecessor function:

$$\text{Pred} \equiv \lambda n. \text{Fst}(n(\lambda p. \langle \text{Snd } p, \text{Succ}(\text{Snd } p) \rangle)(\bar{0}, \bar{0}))$$

Remember that $\bar{n}fx$ reduces to $f^n(x)$; in this case f is a function that accepts a pair p and returns a new pair containing the second component of p and the successor of the second component; x is the pair $\langle 0, 0 \rangle$. Thus, the result is $\langle 0, 0 \rangle$ for $n = 0$, and $\langle \bar{n} - \bar{1}, \bar{n} \rangle$ otherwise. Pred then returns the first component of the result.

Subtraction can be defined as Pred applied to a, b times:

$$\text{Sub} \equiv \lambda ab. b\text{Pred } a.$$

-NoValue-

[content/lambda-calculus/lambda-definability/truth-values.tex](#)

44.4 Truth Values and Relations

lam:ldf:tvr:
sec

We can encode truth values in the pure lambda calculus as follows:

$$\text{true} \equiv \lambda x. \lambda y. x$$

$$\text{false} \equiv \lambda x. \lambda y. y$$

Truth values are represented as *selectors*, O.s., functions that accept two arguments and returning one of them. The truth value true selects its first argument, and false its second. For example, $\text{true } MN$ always reduces to M , while $\text{false } MN$ always reduces to N .

Definição 44.8. We call a relation $R \subseteq \mathbb{N}^n$ λ -definível if there is a term R such that

$$R \overline{n_1} \dots \overline{n_k} \xrightarrow{\beta} \text{true}$$

whenever $R(n_1, \dots, n_k)$ and

$$R \overline{n_1} \dots \overline{n_k} \xrightarrow{\beta} \text{false}$$

otherwise.

For instance, the relation $\text{IsZero} = \{0\}$ which holds of 0 and 0 only, is λ -definível by

$$\text{IsZero} \equiv \lambda n. n(\lambda x. \text{false}) \text{true}.$$

How does it work? Since Church numerals are defined as iterators (functions which apply their first argument n times to the second), we set the initial value to be true, and for every step of iteration, we return false regardless of the result of the last iteration. This step will be applied to the initial value n times, and the result will be true if and only if the step is not applied at all, O.s., when $n = 0$.

On the basis of this representation of truth values, we can further define some truth functions. Here are two, the representations of negation and conjunction:

$$\text{Not} \equiv \lambda x. x \text{false} \text{true}$$

$$\text{And} \equiv \lambda x. \lambda y. xy \text{false}$$

The function “Not” accepts one argument, and returns true if the argument is false, and false if the argument is true. The function “And” accepts two truth values as arguments, and should return true iff both arguments are true. Truth values are represented as selectors (described above), so when x is a truth value and is applied to two arguments, the result will be the first argument if x is true and the second argument otherwise. Now And takes its two arguments x and y , and in return passes y and false to its first argument x . Assuming x is a truth value, the result will evaluate to y if x is true, and to false if x is false, which is just what is desired.

Note that we assume here that only truth values are used as arguments to And. If it is passed other terms, the result (O.s., the normal form, if it exists) may well not be a truth value.

Problema 44.4. Define the functions Or and Xor representing the truth functions of inclusive and exclusive disjunction using the encoding of truth values as λ -terms.

-NoValue-

[content/lambda-calculus/lambda-definability/primitive-recursive-functions.tex](#)

44.5. PRIMITIVE RECURSIVE FUNCTIONS ARE λ -DEFINÍVEL

44.5 Primitive Recursive Functions are λ -definível

lam:ldf:prf: sec Recall that the primitive recursive functions are those that can be defined from the basic functions zero, succ, and P_i^n by composition and primitive recursion.

lam:ldf:prf: lem:basic **Lema 44.9.** *The basic primitive recursive functions zero, succ, and projections P_i^n are λ -definível.*

Prova. They are λ -defined by the following terms:

$$\begin{aligned} \text{Zero} &\equiv \lambda a. \lambda f x. x \\ \text{Succ} &\equiv \lambda a. \lambda f x. f(afx) \\ \text{Proj}_i^n &\equiv \lambda x_0 \dots x_{n-1}. x_i \end{aligned} \quad \square$$

lam:ldf:prf: lem:comp **Lema 44.10.** *Suppose the k -ary function f , and n -ary functions $g_0, \dots, g_{k-1}$, are λ -definível by terms $F, G_0, \dots, G_k$, and h is defined from them by composition. Then H is λ -definível.*

Prova. h can be λ -defined by the term

$$H \equiv \lambda x_0 \dots x_{n-1}. F(G_0 x_0 \dots x_{n-1}) \dots (G_{k-1} x_0 \dots x_{n-1})$$

We leave verification of this fact as an exercise. $\square$

Problema 44.5. Complete the proof of **Lema 44.10** by showing that $H\overline{n_0} \dots \overline{n_{n-1}} \rightarrow \overline{h(n_0, \dots, n_{n-1})}$.

Note that **Lema 44.10** did not require that f and $g_0, \dots, g_{k-1}$ are primitive recursive; it is only required that they are total and λ -definível.

lam:ldf:prf: lem:prim **Lema 44.11.** *Suppose f is an n -ary function and g is an $n+2$ -ary function, they are λ -definível by terms F and G , and the function h is defined from f and g by primitive recursion. Then h is also λ -definível.*

Prova. Recall that h is defined by

$$\begin{aligned} h(x_1, \dots, x_n, 0) &= f(x_1, \dots, x_n) \\ h(x_1, \dots, x_n, y+1) &= h(x_1, \dots, x_n, y, h(x_1, \dots, x_n, y)). \end{aligned}$$

Informally speaking, the primitive recursive definition iterates the application of the function h y times and applies it to $f(x_1, \dots, x_n)$. This is reminiscent of the definition of Church numerals, which is also defined as an iterator.

For simplicity, we give the definition and proof for a single additional argument x . The function h is λ -defined by:

$$H \equiv \lambda x. \lambda y. \text{Snd}(yD(\overline{0}, Fx))$$

where

$$D \equiv \lambda p. \langle \text{Succ}(\text{Fst } p), (Gx(\text{Fst } p)(\text{Snd } p)) \rangle$$

The iteration state we maintain is a pair, the first of which is the current y and the second is the corresponding value of h . For every step of iteration we create a pair of new values of y and h ; after the iteration is done we return the second part of the pair and that's the final h value. We now prove this is indeed a representation of primitive recursion.

We want to prove that for any n and m , $H \bar{n} \bar{m} \rightarrow \overline{h(n, m)}$. To do this we first show that if $D_n \equiv D[\bar{n}/x]$, then $D_n^m \langle \bar{0}, F \bar{n} \rangle \rightarrow \langle \bar{m}, \overline{h(n, m)} \rangle$. We proceed by induction on m .

If $m = 0$, we want $D_n^0 \langle \bar{0}, F \bar{n} \rangle \rightarrow \langle \bar{0}, \overline{h(n, 0)} \rangle$. But $D_n^0 \langle \bar{0}, F \bar{n} \rangle$ just is $\langle \bar{0}, F \bar{n} \rangle$. Since $F \lambda\text{-definível } f$, this reduces to $\langle \bar{0}, \overline{f(n)} \rangle$, and since $f(n) = h(n, 0)$, this is $\langle \bar{0}, \overline{h(n, 0)} \rangle$.

Now suppose that $D_n^m \langle \bar{0}, F \bar{n} \rangle \rightarrow \langle \bar{m}, \overline{h(n, m)} \rangle$. We want to show that $D_n^{m+1} \langle \bar{0}, F \bar{n} \rangle \rightarrow \langle \overline{m+1}, \overline{h(n, m+1)} \rangle$.

$$\begin{aligned} D_n^{m+1} \langle \bar{0}, F \bar{n} \rangle &\equiv D_n(D_n^m \langle \bar{0}, F \bar{n} \rangle) \\ &\rightarrow D_n \langle \bar{m}, \overline{h(n, m)} \rangle \quad (\text{by IH}) \\ &\equiv (\lambda p. \langle \text{Succ}(\text{Fst } p), (G \bar{n}(\text{Fst } p)(\text{Snd } p)) \rangle) \langle \bar{m}, \overline{h(n, m)} \rangle \\ &\rightarrow \langle \text{Succ}(\text{Fst } \langle \bar{m}, \overline{h(n, m)} \rangle), \\ &\quad (G \bar{n}(\text{Fst } \langle \bar{m}, \overline{h(n, m)} \rangle)(\text{Snd } \langle \bar{m}, \overline{h(n, m)} \rangle)) \rangle \\ &\rightarrow \langle \text{Succ } \bar{m}, (G \bar{n} \bar{m} \overline{h(n, m)}) \rangle \\ &\rightarrow \langle \overline{m+1}, \overline{g(n, m, h(n, m))} \rangle \end{aligned}$$

Since $g(n, m, h(n, m)) = h(n, m+1)$, we are done.

Finally, consider

$$\begin{aligned} H \bar{n} \bar{m} &\equiv \lambda x. \lambda y. \text{Snd}(y(\lambda p. \langle \text{Succ}(\text{Fst } p), (G x(\text{Fst } p)(\text{Snd } p)) \rangle) \langle \bar{0}, Fx \rangle) \\ &\quad \bar{n} \bar{m} \\ &\rightarrow \text{Snd}(\bar{m} \underbrace{(\lambda p. \langle \text{Succ}(\text{Fst } p), (G \bar{n}(\text{Fst } p)(\text{Snd } p)) \rangle)}_{D_n} \langle \bar{0}, F \bar{n} \rangle) \\ &\equiv \text{Snd}(\bar{m} D_n \langle \bar{0}, F \bar{n} \rangle) \\ &\rightarrow \text{Snd}(D_n^m \langle \bar{0}, F \bar{n} \rangle) \\ &\rightarrow \text{Snd} \langle \bar{m}, \overline{h(n, m)} \rangle \\ &\rightarrow \overline{h(n, m)}. \end{aligned} \quad \square$$

Proposição 44.12. *Every primitive recursive function is $\lambda\text{-definível}$.*

Prova. By **Lema 44.9**, all basic functions are $\lambda\text{-definível}$, and by **Lema 44.10** and **Lema 44.11**, the $\lambda\text{-definível}$ functions are closed under composition and primitive recursion. $\square$

-NoValue-

content/lambda-calculus/lambda-definability/fixpoints.tex

44.6 Fixpoints

Suppose we wanted to define the factorial function by recursion as a term `Fac` with the following property:

$$\text{Fac} \equiv \lambda n. \text{IsZero } n \bar{1} (\text{Mult } n (\text{Fac} (\text{Pred } n)))$$

That is, the factorial of n is 1 if $n = 0$, and n times the factorial of $n - 1$ otherwise. Of course, we cannot define the term `Fac` this way since `Fac` itself occurs in the right-hand side. Such recursive definitions involving self-reference are not part of the lambda calculus. Defining a term, p.ex., by

$$\text{Mult} \equiv \lambda ab. a (\text{Add } a) 0$$

only involves previously defined terms in the right-hand side, such as `Add`. We can always remove `Add` by replacing it with its defining term. This would give the term `Mult` as a pure lambda term; if `Add` itself involved defined terms (as, p.ex., `Add'` does), we could continue this process and finally arrive at a pure lambda term.

However this is not true in the case of recursive definitions like the one of `Fac` above. If we replace the occurrence of `Fac` on the right-hand side with the definition of `Fac` itself, we get:

$$\begin{aligned} \text{Fac} &\equiv \lambda n. \text{IsZero } n \bar{1} \\ &\quad (\text{Mult } n ((\lambda n. \text{IsZero } n \bar{1} (\text{Mult } n (\text{Fac} (\text{Pred } n)))) (\text{Pred } n))) \end{aligned}$$

and we still haven't gotten rid of `Fac` on the right-hand side. Clearly, if we repeat this process, the definition keeps growing longer and the process never results in a pure lambda term. Thus this way of defining factorial (or more generally recursive functions) is not feasible.

The recursive definition does tell us something, though: If f were a term representing the factorial function, then the term

$$\text{Fac}' \equiv \lambda g. \lambda n. \text{IsZero } n \bar{1} (\text{Mult } n (g (\text{Pred } n)))$$

applied to the term f , O.s., $\text{Fac}' f$, also represents the factorial function. That is, if we regard Fac' as a function accepting a function and returning a function, the value of $\text{Fac}' f$ is just f , provided f is the factorial. A function f with the property that $\text{Fac}' f \stackrel{\beta}{=} f$ is called a *fixpoint* of Fac' . So, the factorial is a fixpoint of Fac' .

There are terms in the lambda calculus that compute the fixpoints of a given term, and these terms can then be used to turn a term like Fac' into the definition of the factorial.

Definição 44.13. The *Y-combinator* is the term:

lam:ldf:fp:
defn:Turing-Y

$$Y \equiv (\lambda ux. x(ux))(\lambda ux. x(ux)).$$

Teorema 44.14. *Y has the property that $Yg \rightarrow g(Yg)$ for any term g . Thus, Yg is always a fixpoint of g .*

Prova. Let's abbreviate $(\lambda ux. x(ux))$ by U , so that $Y \equiv UU$. Then

$$\begin{aligned} Yg &\equiv (\lambda ux. x(ux))Ug \\ &\rightarrow (\lambda x. x(UUx))g \\ &\rightarrow g(UUg) \equiv g(Yg). \end{aligned}$$

Since $g(Yg)$ and Yg both reduce to $g(Yg)$, $g(Yg) \stackrel{\beta}{=} Yg$, so Yg is a fixpoint of g . $\square$

Of course, since Yg is a redex, the reduction can continue indefinitely:

$$\begin{aligned} Yg &\rightarrow g(Yg) \\ &\rightarrow g(g(Yg)) \\ &\rightarrow g(g(g(Yg))) \\ &\dots \end{aligned}$$

So we can think of Yg as g applied to itself infinitely many times. If we apply g to it one additional time, we—so to speak—aren't doing anything extra; g applied to g applied infinitely many times to Yg is still g applied to Yg infinitely many times.

Note that the above sequence of β -reduction steps starting with Yg is infinite. So if we apply Yg to some term, O.s., consider $(Yg)N$, that term will also reduce to infinitely many different terms, namely $(g(Yg))N$, $(g(g(Yg)))N$, $\dots$. It is nevertheless possible that some *other* sequence of reduction steps does terminate in a normal form.

Take the factorial for instance. Define Fac as $Y \text{Fac}'$ (O.s., a fixpoint of Fac'). Then:

$$\begin{aligned} \text{Fac } \bar{3} &\rightarrow Y \text{Fac}' \bar{3} \\ &\rightarrow \text{Fac}'(Y \text{Fac}') \bar{3} \\ &\equiv (\lambda x. \lambda n. \text{IsZero } n \bar{1} (\text{Mult } n (x(\text{Pred } n)))) \text{Fac } \bar{3} \\ &\rightarrow \text{IsZero } \bar{3} \bar{1} (\text{Mult } \bar{3} (\text{Fac}(\text{Pred } \bar{3}))) \\ &\rightarrow \text{Mult } \bar{3} (\text{Fac } \bar{2}). \end{aligned}$$

Similarly,

$$\begin{aligned} \text{Fac } \bar{2} &\rightarrow \text{Mult } \bar{2} (\text{Fac } \bar{1}) \\ \text{Fac } \bar{1} &\rightarrow \text{Mult } \bar{1} (\text{Fac } \bar{0}) \end{aligned}$$

44.6. FIXPOINTS

but

$$\begin{aligned} \text{Fac } \bar{0} &\rightarrow \text{Fac}'(Y \text{Fac}') \bar{0} \\ &\equiv (\lambda x. \lambda n. \text{IsZero } n \bar{1} (\text{Mult } n (x(\text{Pred } n)))) \text{Fac } \bar{0} \\ &\rightarrow \text{IsZero } \bar{0} \bar{1} (\text{Mult } \bar{0} (\text{Fac}(\text{Pred } \bar{0}))). \\ &\rightarrow \bar{1}. \end{aligned}$$

So together

$$\text{Fac } \bar{3} \rightarrow \text{Mult } \bar{3} (\text{Mult } \bar{2} (\text{Mult } \bar{1} \bar{1})).$$

What goes for Fac' goes for any recursive definition. Suppose we have a recursive equation

$$g x_1 \dots x_n \stackrel{\beta}{=} N$$

where N may contain g and $x_1, \dots, x_n$. Then there is always a term $G \equiv (Y \lambda g. \lambda x_1 \dots x_n. N)$ such that

$$G x_1 \dots x_n \stackrel{\beta}{=} N[G/g].$$

For by the fixpoint theorem,

$$\begin{aligned} G &\equiv (Y \lambda g. \lambda x_1 \dots x_n. N) \rightarrow \lambda g. \lambda x_1 \dots x_n. N(Y \lambda g. \lambda x_1 \dots x_n. N) \\ &\equiv (\lambda g. \lambda x_1 \dots x_n. N) G \end{aligned}$$

and consequently

$$\begin{aligned} G x_1 \dots x_n &\rightarrow (\lambda g. \lambda x_1 \dots x_n. N) G x_1 \dots x_n \\ &\rightarrow (\lambda x_1 \dots x_n. N[G/g]) x_1 \dots x_n \\ &\rightarrow N[G/g]. \end{aligned}$$

The Y combinator of [Definição 44.13](#) is due to Alan Turing. Alonzo Church had proposed a different version which we'll call Y_C :

$$Y_C \equiv \lambda g. (\lambda x. g(xx))(\lambda x. g(xx)).$$

Church's combinator is a bit weaker than Turing's in that $Yg \stackrel{\beta}{=} g(Yg)$ but not $Yg \stackrel{\beta}{\rightarrow} g(Yg)$. Let V be the term $\lambda x. g(xx)$, so that $Y_C \equiv \lambda g. VV$. Then

$$\begin{aligned} VV &\equiv (\lambda x. g(xx))V \rightarrow g(VV) \text{ and thus} \\ Y_C g &\equiv (\lambda g. VV)g \rightarrow VV \rightarrow g(VV), \text{ but also} \\ g(Y_C g) &\equiv g((\lambda g. VV)g) \rightarrow g(VV). \end{aligned}$$

In other words, $Y_C g$ and $g(Y_C g)$ reduce to a common term $g(VV)$; so $Y_C g \stackrel{\beta}{=} g(Y_C g)$. This is often enough for applications.

-NoValue-

[content/lambda-calculus/lambda-definability/minimization.tex](#)

44.7 Minimization

The general recursive functions are those that can be obtained from the basic functions zero, succ, P_i^n by composition, primitive recursion, and regular minimization. To show that all general recursive functions are λ -definível we have to show that any function defined by regular minimization from a λ -definível function is itself λ -definível. lam:ldf:min:
sec

Lema 44.15. *If $f(x_1, \dots, x_k, y)$ is regular and λ -definível, then g defined by* lam:ldf:min:
lem:min

$$g(x_1, \dots, x_k) = \mu y \, f(x_1, \dots, x_k, y) = 0$$

is also λ -definível.

Prova. Suppose the lambda term F λ -defines the regular function $f(\vec{x}, y)$. To λ -define h we use a search function and a fixpoint combinator:

$$\begin{aligned} \text{Search} &\equiv \lambda g. \lambda f \, \vec{x} y. \text{IsZero}(f \, \vec{x} y) y (g \, \vec{x} (\text{Succ } y)) \\ H &\equiv \lambda \vec{x}. (Y \, \text{Search}) F \, \vec{x} \, \bar{0}, \end{aligned}$$

where Y is any fixpoint combinator. Informally speaking, Search is a self-referencing function: starting with y , test whether $f \, \vec{x} y$ is zero: if so, return y , otherwise call itself with $\text{Succ } y$. Thus $(Y \, \text{Search}) F \, \vec{n}_1 \dots \vec{n}_k \, \bar{0}$ returns the least m for which $f(n_1, \dots, n_k, m) = 0$.

Specifically, observe that

$$(Y \, \text{Search}) F \, \vec{n}_1 \dots \vec{n}_k \, \bar{m} \rightarrow \bar{m}$$

if $f(n_1, \dots, n_k, m) = 0$, or

$$\rightarrow (Y \, \text{Search}) F \, \vec{n}_1 \dots \vec{n}_k \, \overline{m + 1}$$

otherwise. Since f is regular, $f(n_1, \dots, n_k, y) = 0$ for some y , and so

$$(Y \, \text{Search}) F \, \vec{n}_1 \dots \vec{n}_k \, \bar{0} \rightarrow \overline{h(n_1, \dots, n_k)}. \quad \square$$

Proposição 44.16. *Every general recursive function is λ -definível.*

Prova. By Lema 44.9, all basic functions are λ -definível, and by Lema 44.10, Lema 44.11, and Lema 44.15, the λ -definível functions are closed under composition, primitive recursion, and regular minimization. □

-NoValue-

content/lambda-calculus/lambda-definability/partial-recursive-functions.tex

44.8 Partial Recursive Functions are λ -definível

lam:ldf:par:
sec

Partial recursive functions are those obtained from the basic functions by composition, primitive recursion, and unbounded minimization. They differ from general recursive function in that the functions used in unbounded search are not required to be regular. Not requiring regularity means that functions defined by minimization may sometimes not be defined.

At first glance it might seem that the same methods used to show that the (total) general recursive functions are all λ -definível can be used to prove that all partial recursive functions are λ -definível. For instance, the composition of f with g is λ -defined by $\lambda x. F(Gx)$ if f and g are λ -defined by terms F and G , respectively. However, when the functions are partial, this is problematic. When $g(x)$ is undefined, meaning Gx has no normal form. In most cases this means that $F(Gx)$ has no normal forms either, which is what we want. But consider when F is $\lambda x. \lambda y. y$, in which case $F(Gx)$ does have a normal form $(\lambda y. y)$.

This problem is not insurmountable, and there are ways to λ -define all partial recursive functions in such a way that undefined values are represented by terms without a normal form. These ways are, however, somewhat more complicated and less intuitive than the approach we have taken for general recursive functions. We record the theorem here without proof:

Teorema 44.17. *All partial recursive functions are λ -definível.*

-NoValue-

<content/lambda-calculus/lambda-definability/lambda-definable-recursive.tex>

44.9 λ -definível Functions are Recursive

lam:df:ldr:
sec

Not only are all partial recursive functions λ -definível, the converse is true, too. That is, all λ -definível functions are partial recursive.

lam:df:ldr:
thm:lambda-computable

Teorema 44.18. *If a partial function f is λ -definível, it is partial recursive.*

Prova. We only sketch the proof. First, we arithmetize λ -terms, O.s., systematically assign Gödel numbers to λ -terms, using the usual power-of-primes coding of sequences. Then we define a partial recursive function $\text{normalize}(t)$ operating on the Gödel number t of a lambda term as argument, and which returns the Gödel number of the normal form if it has one, or is undefined otherwise. Then define two partial recursive functions toChurch and fromChurch that maps natural numbers to and from the Gödel numbers of the corresponding Church numeral.

Using these recursive functions, we can define the function f as a partial recursive function. There is a λ -term F that λ -definem f . To compute $f(n_1, \dots, n_k)$, first obtain the Gödel numbers of the corresponding Church numerals using $\text{toChurch}(n_i)$, append these to ${}^*F^\#$ to obtain the Gödel number of the term $F\overline{n_1} \dots \overline{n_k}$. Now use normalize on this Gödel number. If $f(n_1, \dots, n_k)$

CAPÍTULO 44. LAMBDA DEFINABILITY

is defined, $F\overline{n_1} \dots \overline{n_k}$ has a normal form (which must be a Church numeral), and otherwise it has no normal form (and so

$$\text{normalize}(\#F\overline{n_1} \dots \overline{n_k}\#)$$

is undefined). Finally, use `fromChurch` on the Gödel number of the normalized term. □

Parte X

Many-valued Logic

This part contains draft material on propositional many-valued logics.

Capítulo 45

Syntax and Semantics

-NoValue-

`content/many-valued-logic/syntax-and-semantics/introduction.tex`

45.1 Introduction

mvl:syn:int:
sec In classical logic, we deal with **fórmulas** that are built from **variáveis proposicionais** using the propositional connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$. When we define a semantics for classical logic, we do so using the two truth values $\mathbb{T}$ and $\mathbb{F}$. We interpret **variáveis proposicionais** in a **valoração** $\mathfrak{v}$, which assigns these truth values $\mathbb{T}$, $\mathbb{F}$ to the **variáveis proposicionais**. Any **valoração** then determines a truth value $\bar{\mathfrak{v}}(\varphi)$ for any **fórmula** φ , and **A fórmula** is satisfied in a **valoração** $\mathfrak{v}$, $\mathfrak{v} \models \varphi$, iff $\bar{\mathfrak{v}}(\varphi) = \mathbb{T}$.

Many-valued logics are generalizations of classical two-valued logic by allowing more truth values than just $\mathbb{T}$ and $\mathbb{F}$. So in many-valued logic, a **valoração** $\mathfrak{v}$ is a function assigning to every **variável proposicional** p one of a range of possible truth values. We'll generally call the set of allowed truth values V . Classical logic is a many-valued logic where $V = \{\mathbb{T}, \mathbb{F}\}$, and the truth value $\bar{\mathfrak{v}}(\varphi)$ is computed using the familiar characteristic truth tables for the connectives.

Once we add additional truth values, we have more than one natural option for how to compute $\bar{\mathfrak{v}}(\varphi)$ for the connectives we read as “and,” “or,” “not,” and “if—then.” So a many-valued logic is determined not just by the set of truth

values, but also by the *truth functions* we decide to use for each connective. Once these are selected for a many-valued logic $\mathbf{L}$, however, the truth value $\bar{\mathbf{v}}_{\mathbf{L}}(\varphi)$ is uniquely determined by the valuation, just like in classical logic. Many-valued logics, like classical logic, are *truth functional*.

With this semantic building blocks in hand, we can go on to define the analogs of the semantic concepts of tautology, entailment, and satisfiability. In classical logic, a *fórmula* is a tautology if its truth value $\bar{\mathbf{v}}(\varphi) = \mathbb{T}$ for any $\mathbf{v}$. In many-valued logic, we have to generalize this a bit as well. First of all, there is no requirement that the set of truth values V contains $\mathbb{T}$. For instance, some many-valued logics use numbers, such as all rational numbers between 0 and 1 as their set of truth values. In such a case, 1 usually plays the role of $\mathbb{T}$. In other logics, not just one but several truth values do. So, we require that every many-valued logic have a set V^+ of *designated values*. We can then say that a *fórmula* is satisfied in a *valoração* $\mathbf{v}$, $\mathbf{v} \models_{\mathbf{L}} \varphi$, iff $\bar{\mathbf{v}}_{\mathbf{L}}(\varphi) \in V^+$. A *fórmula* φ is a tautology of the logic, $\models_{\mathbf{L}} \varphi$, iff $\bar{\mathbf{v}}(\varphi) \in V^+$ for any $\mathbf{v}$. And, finally, we say that φ is entailed by a set of *fórmulas*, $\Gamma \models_{\mathbf{L}} \varphi$, if every *valoração* that satisfies all the *fórmulas* in Γ also satisfies φ .

-NoValue-

`content/many-valued-logic/syntax-and-semantics/connectives.tex`

45.2 Languages and Connectives

Classical propositional logic, and many other logics, use a set supply of *propositional constants* and *connectives*. For instance, we use the following as primitives:

mvl:syn:con:
sec

1. The propositional constant for *falsidade* $\perp$.
2. The propositional constant for *verdade* $\top$.
3. The logical connectives: $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\rightarrow$ (*condicional*), $\leftrightarrow$ (*bicondicional*)

The same connectives are used in many-valued logics as well. However, it is often useful to include different versions of, say, conjunction, in the same logic, and that would require different symbols to keep the versions separate. Some many-valued logics also include connectives that have no equivalent in classical logic. So, we'll be a bit more general than usual.

Definição 45.1. A *propositional language* consists of a set $\mathcal{L}$ of *connectives*. Each connective $\star$ has an *arity*; a connective of arity n is said to be *n-place*. Connectives of arity 0 are also called *constants*; connectives of arity 1 are called *unary*, and connectives of arity 2, *binary*.

Exemplo 45.2. The standard language of propositional logic $\mathcal{L}_0$ consists of the following connectives (with associated arities): $\perp$ (0) $\neg$ (1), $\wedge$ (2), $\vee$ (2),

45.3. FÓRMULAS

$\rightarrow$ (2). Most logics we consider will use this language. Some logics by tradition an convention use different symbols for some connectives. For instance, in product logic, the conjunction symbol is often $\odot$ instead of $\wedge$. Sometimes it is convenient to add a new operator, p.ex., the determinateness operator Δ (1-place).

-NoValue-

content/many-valued-logic/syntax-and-semantics/formulas.tex

45.3 Fórmulas

mvl:syn:fml:
mvl:syn:fml:
defn:formulas

Definição 45.3 (Formula). The set $\text{Frm}(\mathcal{L})$ of *fórmulas* of a propositional language $\mathcal{L}$ is defined inductively as follows:

1. Every *variável proposicional* p_i is an atomic *fórmula*.
2. Every 0-place connective (propositional constant) of $\mathcal{L}$ is an atomic *fórmula*.
3. If $\star$ is an n -place connective of $\mathcal{L}$, and $\varphi_1, \dots, \varphi_n$ are *fórmulas*, then $\star(\varphi_1, \dots, \varphi_n)$ is a *fórmula*.
4. Nothing else is a *fórmula*.

If $\star$ is 1-place, then $\star(\varphi_1)$ will often be written simply as $\star\varphi_1$. If $\star$ is 2-place $\star(\varphi_1, \varphi_2)$ will often be written as $(\varphi_1 \star \varphi_2)$.

As usual, we will often silently leave out the outermost parentheses.

Exemplo 45.4. In the standard language $\mathcal{L}_0$, $p_1 \rightarrow (p_1 \wedge \neg p_2)$ is a formula. In the language of product logic, it would be written instead as $p_1 \rightarrow (p_1 \odot \neg p_2)$. If we add the 1-place Δ to the language, we would also have formulas such as $\Delta(p_1 \wedge p_2) \rightarrow (\Delta p_1 \wedge \Delta p_2)$.

-NoValue-

content/many-valued-logic/syntax-and-semantics/matrices.tex

45.4 Matrices

mvl:syn:mat:
sec

A many-valued logic is defined by its language, its set of truth values V , a subset of designated truth values, and truth functions for its connective. Together, these elements are called a *matrix*.

mvl:syn:mat:
defn:matrix

Definição 45.5 (Matrix). A *matrix* for the logic $\mathbf{L}$ consists of:

1. a set of connectives making up a language $\mathcal{L}$;

$\simeq$		$\tilde{\wedge}$	T	F	$\tilde{\vee}$	T	F	$\tilde{\rightarrow}$	T	F
T	F	T	T	F	T	T	T	T	T	F
F	T	F	F	F	F	T	F	F	T	T

Figura 45.1: Truth functions for classical logic **C**.

mv1:syn:mat:
fig:tf-CL

2. a set $V \neq \emptyset$ of truth values;
3. a set $V^+ \subseteq V$ of designated truth values;
4. for each n -place connective $\star$ in $\mathcal{L}$, a truth function $\tilde{\star}: V^n \rightarrow V$. If $n = 0$, then $\tilde{\star}$ is just an element of V .

Exemplo 45.6. The matrix for classical logic **C** consists of:

1. The standard propositional language $\mathcal{L}_0$ with $\perp, \neg, \wedge, \vee, \rightarrow$.
2. The set of truth values $V = \{\mathbb{T}, \mathbb{F}\}$.
3. $\mathbb{T}$ is the only designated value, O.s., $V^+ = \{\mathbb{T}\}$.
4. For $\perp$, we have $\tilde{\perp} = \mathbb{F}$. The other truth functions are given by the usual truth tables (see [Figura 45.1](#)).

-NoValue-

[content/many-valued-logic/syntax-and-semantics/valuations-sat.tex](#)

45.5 Valorações and Satisfaction

Definição 45.7 (Valorações). Let V be a set of truth values. A *valoração* for $\mathcal{L}$ into V is a function $\mathbf{v}$ assigning *um membro* of V to the *variáveis proposicionais* of the language, O.s., $\mathbf{v}: \text{At}_0 \rightarrow V$.

mv1:syn:val:
sec

Definição 45.8. Given a *valoração* $\mathbf{v}$ into the set of truth values V of a many-valued logic **L**, define the evaluation function $\bar{\mathbf{v}}: \text{Frm}(\mathcal{L}) \rightarrow V$ inductively by:

mv1:syn:val:
defn:pValue

1. $\bar{\mathbf{v}}(p_n) = \mathbf{v}(p_n)$;
2. If $\star$ is a 0-place connective, then $\bar{\mathbf{v}}(\star) = \tilde{\star}_{\mathbf{L}}$;
3. If $\star$ is an n -place connective, then

$$\bar{\mathbf{v}}(\star(\varphi_1, \dots, \varphi_n)) = \tilde{\star}_{\mathbf{L}}(\bar{\mathbf{v}}(\varphi_1), \dots, \bar{\mathbf{v}}(\varphi_n)).$$

45.6. SEMANTIC NOTIONS

myl:syn:val:
defn:satisfaction

Definição 45.9 (Satisfaction). The fórmula φ is *satisfied* by a *valoração* $\mathbf{v}$, $\mathbf{v} \models_{\mathbf{L}} \varphi$, iff $\bar{\mathbf{v}}_{\mathbf{L}}(\varphi) \in V^+$, where V^+ is the set of designated truth values of $\mathbf{L}$.

We write $\mathbf{v} \not\models_{\mathbf{L}} \varphi$ to mean “not $\mathbf{v} \models_{\mathbf{L}} \varphi$.” If Γ is a set of fórmulas, $\mathbf{v} \models_{\mathbf{L}} \Gamma$ iff $\mathbf{v} \models_{\mathbf{L}} \varphi$ for every $\varphi \in \Gamma$.

-NoValue-

content/many-valued-logic/syntax-and-semantics/semantic-notions.tex

45.6 Semantic Notions

myl:syn:sem:
sec

Suppose a many-valued logic $\mathbf{L}$ is given by a matrix. Then we can define the usual semantic notions for $\mathbf{L}$.

- Definição 45.10.**
1. A fórmula φ is *satisfiable* if for some $\mathbf{v}$, $\mathbf{v} \models \varphi$; it is *unsatisfiable* if for no $\mathbf{v}$, $\mathbf{v} \models \varphi$;
 2. A fórmula φ is a *tautology* if $\mathbf{v} \models \varphi$ for all *valorações* $\mathbf{v}$;
 3. If Γ is a set of fórmulas, $\Gamma \models \varphi$ (“ Γ entails φ ”) if and only if $\mathbf{v} \models \varphi$ for every *valoração* $\mathbf{v}$ for which $\mathbf{v} \models \Gamma$.
 4. If Γ is a set of fórmulas, Γ is *satisfiable* if there is a *valoração* $\mathbf{v}$ for which $\mathbf{v} \models \Gamma$, and Γ is *unsatisfiable* otherwise.

We have some of the same facts for these notions as we do for the case of classical logic:

myl:syn:sem:
prop:semanticalfacts

Proposição 45.11.

1. φ is a tautology if and only if $\emptyset \models \varphi$;
2. If Γ is satisfiable then every finite subset of Γ is also satisfiable;
3. Monotonicity: if $\Gamma \subseteq \Delta$ and $\Gamma \models \varphi$ then also $\Delta \models \varphi$;
4. Transitivity: if $\Gamma \models \varphi$ and $\Delta \cup \{\varphi\} \models \psi$ then $\Gamma \cup \Delta \models \psi$;

myl:syn:sem:
def:monotonicity
myl:syn:sem:
def:Cut

Prova. Exercise. □

Problema 45.1. Prove **Proposição 45.11**

In classical logic we can connect entailment and the conditional. For instance, we have the validity of *modus ponens*: If $\Gamma \models \varphi$ and $\Gamma \models \varphi \rightarrow \psi$ then $\Gamma \models \psi$. Another important relationship between $\models$ and $\rightarrow$ in classical logic is the semantic deduction theorem: $\Gamma \models \varphi \rightarrow \psi$ if and only if $\Gamma \cup \{\varphi\} \models \psi$. These results *do not* always hold in many-valued logics. Whether they do depends on the truth function $\widetilde{\rightarrow}$.

-NoValue-

content/many-valued-logic/syntax-and-semantics/sublogics.tex

45.7 Many-valued logics as sublogics of $\mathbf{C}$

The usual many-valued logics are all defined using matrices in which the value of a truth-function for arguments in $\{\mathbb{T}, \mathbb{F}\}$ agrees with the classical truth functions. Specifically, in these logics, if $x \in \{\mathbb{T}, \mathbb{F}\}$, then $\sim_{\mathbf{L}}(x) = \sim_{\mathbf{C}}(x)$, and for $\star$ any one of $\wedge, \vee, \rightarrow$, if $x, y \in \{\mathbb{T}, \mathbb{F}\}$, then $\star_{\mathbf{L}}(x, y) = \star_{\mathbf{C}}(x, y)$. In other words, the truth functions for $\neg, \wedge, \vee, \rightarrow$ restricted to $\{\mathbb{T}, \mathbb{F}\}$ are exactly the classical truth functions.

*mvl:syn:sub:
sec*

Proposição 45.12. *Suppose that a many-valued logic $\mathbf{L}$ contains the connectives $\neg, \wedge, \vee, \rightarrow$ in its language, $\mathbb{T}, \mathbb{F} \in V$, and its truth functions satisfy:*

*mvl:syn:sub:
prop:mvl-cl*

1. $\sim_{\mathbf{L}}(x) = \sim_{\mathbf{C}}(x)$ if $x = \mathbb{T}$ or $x = \mathbb{F}$;
2. $\tilde{\wedge}_{\mathbf{L}}(x, y) = \tilde{\wedge}_{\mathbf{C}}(x, y)$,
3. $\tilde{\vee}_{\mathbf{L}}(x, y) = \tilde{\vee}_{\mathbf{C}}(x, y)$,
4. $\tilde{\rightarrow}_{\mathbf{L}}(x, y) = \tilde{\rightarrow}_{\mathbf{C}}(x, y)$, if $x, y \in \{\mathbb{T}, \mathbb{F}\}$.

*mvl:syn:sub:
prop:not
mvl:syn:sub:
prop:land
mvl:syn:sub:
prop:lor
mvl:syn:sub:
prop:lif*

Then, for any valuation $\mathbf{v}$ into V such that $\mathbf{v}(p) \in \{\mathbb{T}, \mathbb{F}\}$, $\bar{\mathbf{v}}_{\mathbf{L}}(\varphi) = \bar{\mathbf{v}}_{\mathbf{C}}(\varphi)$.

Prova. By induction on φ .

1. If $\varphi \equiv p$ is atomic, we have $\bar{\mathbf{v}}_{\mathbf{L}}(\varphi) = \mathbf{v}(p) = \bar{\mathbf{v}}_{\mathbf{C}}(\varphi)$.
2. If $\varphi \equiv \neg B$, we have

$$\begin{aligned} \bar{\mathbf{v}}_{\mathbf{L}}(\varphi) &= \sim_{\mathbf{L}}(\bar{\mathbf{v}}_{\mathbf{L}}(\psi)) && \text{by Definição 45.8} \\ &= \sim_{\mathbf{L}}(\bar{\mathbf{v}}_{\mathbf{C}}(\psi)) && \text{by inductive hypothesis} \\ &= \sim_{\mathbf{C}}(\bar{\mathbf{v}}_{\mathbf{C}}(\psi)) && \text{by assumption (1),} \\ & && \text{since } \bar{\mathbf{v}}_{\mathbf{C}}(\psi) \in \{\mathbb{T}, \mathbb{F}\}, \\ &= \bar{\mathbf{v}}_{\mathbf{C}}(\varphi) && \text{by Definição 45.8.} \end{aligned}$$

3. If $\varphi \equiv (\psi \wedge \chi)$, we have

$$\begin{aligned} \bar{\mathbf{v}}_{\mathbf{L}}(\varphi) &= \tilde{\wedge}_{\mathbf{L}}(\bar{\mathbf{v}}_{\mathbf{L}}(\psi), \bar{\mathbf{v}}_{\mathbf{L}}(\chi)) && \text{by Definição 45.8} \\ &= \tilde{\wedge}_{\mathbf{L}}(\bar{\mathbf{v}}_{\mathbf{C}}(\psi), \bar{\mathbf{v}}_{\mathbf{C}}(\chi)) && \text{by inductive hypothesis} \\ &= \tilde{\wedge}_{\mathbf{C}}(\bar{\mathbf{v}}_{\mathbf{C}}(\psi), \bar{\mathbf{v}}_{\mathbf{C}}(\chi)) && \text{by assumption (2),} \\ & && \text{since } \bar{\mathbf{v}}_{\mathbf{C}}(\psi), \bar{\mathbf{v}}_{\mathbf{C}}(\chi) \in \{\mathbb{T}, \mathbb{F}\}, \\ &= \bar{\mathbf{v}}_{\mathbf{C}}(\varphi) && \text{by Definição 45.8.} \end{aligned}$$

The cases where $\varphi \equiv (\psi \vee \chi)$ and $\varphi \equiv (\psi \rightarrow \chi)$ are similar. $\square$

Corolário 45.13. *If a many-valued logic satisfies the conditions of [Proposição 45.12](#), $\mathbb{T} \in V^+$ and $\mathbb{F} \notin V^+$, then $\models_{\mathbf{L}} \subseteq \models_{\mathbf{C}}$. O.s., if $\Gamma \models_{\mathbf{L}} \psi$ then $\Gamma \models_{\mathbf{C}} \psi$. In particular, every tautology of $\mathbf{L}$ is also a classical tautology.*

Prova. We prove the contrapositive. Suppose $\Gamma \not\models_{\mathbf{C}} \psi$. Then there is some **valoração** $\mathbf{v}: \text{At}_0 \rightarrow \{\mathbb{T}, \mathbb{F}\}$ such that $\bar{\mathbf{v}}_{\mathbf{C}}(\varphi) = \mathbb{T}$ for all $\varphi \in \Gamma$ and $\bar{\mathbf{v}}_{\mathbf{C}}(\psi) = \mathbb{F}$. Since $\mathbb{T}, \mathbb{F} \in V$, the **valoração** $\mathbf{v}$ is also a **valoração** for $\mathbf{L}$. By **Proposição 45.12**, $\bar{\mathbf{v}}_{\mathbf{L}}(\varphi) = \mathbb{T}$ for all $\varphi \in \Gamma$ and $\bar{\mathbf{v}}_{\mathbf{L}}(\psi) = \mathbb{F}$. Since $\mathbb{T} \in V^+$ and $\mathbb{F} \notin V^+$ that means $\mathbf{v} \models_{\mathbf{L}} \Gamma$ and $\mathbf{v} \not\models_{\mathbf{L}} \psi$, O.s., $\Gamma \not\models_{\mathbf{L}} \psi$. $\square$

Capítulo 46

Three-valued Logics

-NoValue-

`content/many-valued-logic/three-valued-logics/introduction.tex`

46.1 Introduction

mvl:thr:int:
sec If we just add one more value $\mathbb{U}$ to $\mathbb{T}$ and $\mathbb{F}$, we get a three-valued logic. Even though there is only one more truth value, the possibilities for defining the truth-functions for $\neg$, $\wedge$, $\vee$, and $\rightarrow$ are quite numerous. Then a logic might use any combination of these truth functions, and you also have a choice of making only $\mathbb{T}$ designated, or both $\mathbb{T}$ and $\mathbb{U}$.

We present here a selection of the most well-known three-valued logics, their motivations, and some of their properties.

-NoValue-

`content/many-valued-logic/three-valued-logics/lukasiewicz.tex`

46.2 Łukasiewicz logic

mvl:thr:luk:
sec One of the first published, worked out proposals for a many-valued logic is due to the Polish philosopher Jan Łukasiewicz in 1921. Łukasiewicz was motivated by Aristotle's sea battle problem: It seems that, *today*, the sentence "There will be a sea battle tomorrow" is neither true nor false: its truth value is not yet settled. Łukasiewicz proposed to introduce a third truth value, to such "future contingent" sentences.

I can assume without contradiction that my presence in Warsaw at a certain moment of next year, p.ex., at noon on 21 December, is at the present time determined neither positively nor negatively. Hence it is possible, but not necessary, that I shall be present in Warsaw at the given time. On this assumption the proposition “I shall be in Warsaw at noon on 21 December of next year,” can at the present time be neither true nor false. For if it were true now, my future presence in Warsaw would have to be necessary, which is contradictory to the assumption. If it were false now, on the other hand, my future presence in Warsaw would have to be impossible, which is also contradictory to the assumption. Therefore the proposition considered is at the moment neither true nor false and must possess a third value, different from “0” or falsity and “1” or truth. This value we can designate by “ $\frac{1}{2}$.” It represents “the possible,” and joins “the true” and “the false” as a third value.

We will use $\mathbb{U}$ for Łukasiewicz’s third truth value.¹

The truth functions for the connectives $\neg$, $\wedge$, and $\vee$ are easy to determine on this interpretation: the negation of a future contingent sentence is also a future contingent sentence, so $\neg(\mathbb{U}) = \mathbb{U}$. If one conjunct of a conjunction is undetermined and the other is true, the conjunction is also undetermined—after all, depending on how the future contingent conjunct turns out, the conjunction might turn out to be true, and it might turn out to be false. So

$$\widetilde{\wedge}(\mathbb{T}, \mathbb{U}) = \widetilde{\wedge}(\mathbb{U}, \mathbb{T}) = \mathbb{U}.$$

If the other conjunct is false, however, it cannot turn out true, so

$$\widetilde{\wedge}(\mathbb{F}, \mathbb{U}) = \widetilde{\wedge}(\mathbb{U}, \mathbb{F}) = \mathbb{F}.$$

The other values (if the arguments are settled truth values, $\mathbb{T}$ or $\mathbb{F}$, are like in classical logic.

For the conditional, the situation is a little trickier. Suppose q is a future contingent statement. If p is false, then $p \rightarrow q$ will be true, regardless of how q turns out, so we should set $\widetilde{\rightarrow}(\mathbb{F}, \mathbb{U}) = \mathbb{T}$. And if p is true, then $q \rightarrow p$ will be true, regardless of what q turns out to be, so $\widetilde{\rightarrow}(\mathbb{U}, \mathbb{T}) = \mathbb{T}$. If p is true, then $p \rightarrow q$ might turn out to be true or false, so $\widetilde{\rightarrow}(\mathbb{T}, \mathbb{U}) = \mathbb{U}$. Similarly, if p is false, then $q \rightarrow p$ might turn out to be true or false, so $\widetilde{\rightarrow}(\mathbb{U}, \mathbb{F}) = \mathbb{U}$. This leaves the case where p and q are both future contingents. On the basis of the motivation, we should really assign $\mathbb{U}$ in this case. However, this would make $\varphi \rightarrow \varphi$ *not* a tautology. Łukasiewicz had not trouble giving up $\varphi \vee \neg\varphi$ and $\neg(\varphi \wedge \neg\varphi)$, but balked at giving up $\varphi \rightarrow \varphi$. So he stipulated $\widetilde{\rightarrow}(\mathbb{U}, \mathbb{U}) = \mathbb{T}$.

Definição 46.1. Three-valued Łukasiewicz logic is defined using the matrix:

mvl:thr:luk:
def:lukasiewicz

1. The standard propositional language $\mathcal{L}_0$ with $\neg$, $\wedge$, $\vee$, $\rightarrow$.

¹Łukasiewicz here uses “possible” in a way that is uncommon today, namely to mean possible but not necessary.

46.2. LUKASIEWICZ LOGIC

2. The set of truth values $V = \{\mathbb{T}, \mathbb{U}, \mathbb{F}\}$.
3. $\mathbb{T}$ is the only designated value, O.s., $V^+ = \{\mathbb{T}\}$.
4. Truth functions are given by the following tables:

$\neg$		$\tilde{\wedge}_{\mathbf{L}_3}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$

$\tilde{\vee}_{\mathbf{L}_3}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$	$\tilde{\rightarrow}_{\mathbf{L}_3}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{U}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$

As can easily be seen, any **fórmula** φ containing only $\neg$, $\wedge$, and $\vee$ will take the truth value $\mathbb{U}$ if all its **variáveis proposicionais** are assigned $\mathbb{U}$. So for instance, the classical tautologies $p \vee \neg p$ and $\neg(p \wedge \neg p)$ are not tautologies in $\mathbf{L}_3$, since $\bar{\mathbf{v}}(\varphi) = \mathbb{U}$ whenever $\mathbf{v}(p) = \mathbb{U}$.

On **valorações** where $\mathbf{v}(p) = \mathbb{T}$ or $\mathbb{F}$, $\bar{\mathbf{v}}(\varphi)$ will coincide with its classical truth value.

Proposição 46.2. *If $\mathbf{v}(p) \in \{\mathbb{T}, \mathbb{F}\}$ for all p in φ , then $\bar{\mathbf{v}}_{\mathbf{L}_3}(\varphi) = \bar{\mathbf{v}}_{\mathbf{C}}(\varphi)$.*

Problema 46.1. Suppose we define $\bar{\mathbf{v}}(\varphi \leftrightarrow \psi) = \bar{\mathbf{v}}((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi))$ in $\mathbf{L}_3$. What truth table would $\leftrightarrow$ have?

Many classical tautologies *are* also tautologies in $\mathbf{L}_3$, e.g., $\neg p \rightarrow (p \rightarrow q)$. Just like in classical logic, we can use truth tables to verify this:

p	q	$\neg$	p	$\rightarrow$	$(p$	$\rightarrow$	$q)$
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{U}$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{U}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{T}$	$\mathbb{U}$
$\mathbb{U}$	$\mathbb{F}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{U}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{U}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$

Problema 46.2. Show that the following are tautologies in $\mathbf{L}_3$:

1. $p \rightarrow (q \rightarrow p)$
2. $\neg(p \wedge q) \leftrightarrow (\neg p \vee \neg q)$
3. $\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$

(In (2) and (3), take $\varphi \leftrightarrow \psi$ as an abbreviation for $(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$, or refer to your solution to Problema 46.1.)

Problema 46.3. Show that the following classical tautologies are not tautologies in $\mathbf{L}_3$:

1. $(\neg p \wedge p) \rightarrow q$
2. $((p \rightarrow q) \rightarrow p) \rightarrow p$
3. $(p \rightarrow (p \rightarrow q)) \rightarrow (p \rightarrow q)$

One might therefore perhaps think that although not all classical tautologies are tautologies in $\mathbf{L}_3$, they should at least take either the value $\mathbb{T}$ or the value $\mathbb{U}$ on every *valoração*. This is not the case. A counterexample is given by

$$\neg(p \rightarrow \neg p) \vee \neg(\neg p \rightarrow p)$$

which is $\mathbb{F}$ if p is $\mathbb{U}$.

Problema 46.4. Which of the following relations hold in Łukasiewicz logic? Give a truth table for each.

1. $p, p \rightarrow q \models q$
2. $\neg\neg p \models p$
3. $p \wedge q \models p$
4. $p \models p \wedge p$
5. $p \models p \vee q$

Łukasiewicz hoped to build a logic of possibility on the basis of his three-valued system, by introducing a one-place connective $\Diamond\varphi$ (for “ φ is possible”) and a corresponding $\Box\varphi$ (for “ φ is necessary”):

$\tilde{\Diamond}$		$\tilde{\Box}$	
$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{T}$
$\mathbb{U}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$

In other words, p is possible iff it is not already settled as false; and p is necessary iff it is already settled as true.

Problema 46.5. Show that $\Box p \leftrightarrow \neg\Diamond\neg p$ and $\Diamond p \leftrightarrow \neg\Box\neg p$ are tautologies in $\mathbf{L}_3$, extended with the truth tables for $\Box$ and $\Diamond$.

46.3. KLEENE LOGICS

However, the shortcomings of this proposed modal logic soon became evident: However things turn out, $p \wedge \neg p$ can never turn out to be true. So even if it is not now settled (and therefore undetermined), it should count as impossible, O.s., $\neg \Diamond(p \wedge \neg p)$ should be a tautology. However, if $\mathbf{v}(p) = \mathbb{U}$, then $\bar{\mathbf{v}}(\neg \Diamond(p \wedge \neg p)) = \mathbb{U}$. Although Łukasiewicz was correct that two truth values will not be enough to accommodate modal distinctions such as possibility and necessity, introducing a third truth value is also not enough.

-NoValue-

content/many-valued-logic/three-valued-logics/kleene.tex

46.3 Kleene logics

mvl:thr:skl:
sec

Stephen Kleene introduced two three-valued logics motivated by a logic in which truth values are thought of the outcomes of computational procedures: a procedure may yield $\mathbb{T}$ or $\mathbb{F}$, but it may also fail to terminate. In that case the corresponding truth value is undefined, represented by the truth value $\mathbb{U}$.

To compute the negation of a proposition φ , you would first compute the value of φ , and then return the opposite of the result. If the computation of φ does not terminate, then the entire procedure does not either: so the negation of $\mathbb{U}$ is $\mathbb{U}$.

To compute a conjunction $\varphi \wedge \psi$, there are two options: one can first compute φ , then ψ , and then the result would be $\mathbb{T}$ if the outcome of both is $\mathbb{T}$, and $\mathbb{F}$ otherwise. If either computation fails to halt, the entire procedure does as well. So in this case, the if one conjunct is undefined, the conjunction is as well. The same goes for disjunction.

However, if we can evaluate φ and ψ in parallel, we can do better. Then, if one of the two procedures halts and returns $\mathbb{F}$, we can stop, as the answer must be false. So in that case a conjunction with one false conjunct is false, even if the other conjunct is undefined. Similarly, when computing a disjunction in parallel, we can stop once the procedure for one of the two disjuncts has returned true: then the disjunction must be true. So in this case we can know what the outcome of a compound claim is, even if one of the components is undefined. On this interpretation, we might read $\mathbb{U}$ as “unknown” rather than “undefined.”

The two interpretations give rise to Kleene’s strong and weak logic. The conditional is defined as equivalent to $\neg \varphi \vee \psi$.

Definição 46.3. *Strong Kleene logic* **Ks** is defined using the matrix:

1. The standard propositional language $\mathcal{L}_0$ with $\neg, \wedge, \vee, \rightarrow$.
2. The set of truth values $V = \{\mathbb{T}, \mathbb{U}, \mathbb{F}\}$.
3. $\mathbb{T}$ is the only designated value, O.s., $V^+ = \{\mathbb{T}\}$.
4. Truth functions are given by the following tables:

$\approx$		$\tilde{\wedge}_{\mathbf{Ks}}$	T	U	F
T	F	T	T	U	F
U	U	U	U	U	F
F	T	F	F	F	F

$\tilde{\vee}_{\mathbf{Ks}}$	T	U	F	$\tilde{\rightarrow}_{\mathbf{Ks}}$	T	U	F
T	T	T	T	T	T	U	F
U	T	U	U	U	T	U	U
F	T	U	F	F	T	T	T

Definição 46.4. *Weak Kleene logic $\mathbf{Kw}$* is defined using the matrix:

1. The standard propositional language $\mathcal{L}_0$ with $\neg, \wedge, \vee, \rightarrow$.
2. The set of truth values $V = \{\mathbf{T}, \mathbf{U}, \mathbf{F}\}$.
3. $\mathbf{T}$ is the only designated value, O.s., $V^+ = \{\mathbf{T}\}$.
4. Truth functions are given by the following tables:

$\approx$		$\tilde{\wedge}_{\mathbf{Kw}}$	T	U	F
T	F	T	T	U	F
U	U	U	U	U	U
F	T	F	F	U	F

$\tilde{\vee}_{\mathbf{Kw}}$	T	U	F	$\tilde{\rightarrow}_{\mathbf{Kw}}$	T	U	F
T	T	U	T	T	T	U	F
U	U	U	U	U	U	U	U
F	T	U	F	F	T	U	T

Proposição 46.5. *$\mathbf{Ks}$ and $\mathbf{Kw}$ have no tautologies.*

Prova. If $\mathbf{v}(p) = \mathbf{U}$ for all **variáveis proposicionais** p , then any formula φ will have truth value $\mathbf{v}(\varphi) = \mathbf{U}$, since

$$\approx(\mathbf{U}) = \tilde{\vee}(\mathbf{U}, \mathbf{U}) = \tilde{\wedge}(\mathbf{U}, \mathbf{U}) = \tilde{\rightarrow}(\mathbf{U}, \mathbf{U}) = \mathbf{U}$$

in both logics. As $\mathbf{U} \notin V^+$ for either $\mathbf{Ks}$ or $\mathbf{Kw}$, on this **valoração**, φ will not be designated. $\square$

Although both weak and strong Kleene logic have no tautologies, they have non-trivial consequence relations.

Problema 46.6. Which of the following relations hold in (a) strong and (b) weak Kleene logic? Give a truth table for each.

1. $p, p \rightarrow q \models q$
2. $p \vee q, \neg p \models q$

46.4. GÖDEL LOGICS

- 3. $p \wedge q \models p$
- 4. $p \models p \wedge p$
- 5. $p \models p \vee q$

Dmitry Bochvar interpreted $\mathbb{U}$ as “meaningless” and attempted to use it to solve paradoxes such as the Liar paradox by stipulating that paradoxical sentences take the value $\mathbb{U}$. He introduced a logic which is essentially weak Kleene logic extended by additional connectives, two of which are “external negation” and the “is undefined” operator:

$\sim$		$\tilde{+}$	
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{F}$

Problema 46.7. Can you define $\sim$ in Bochvar’s logic in terms of $\neg$ and $+$, O.s., find a fórmula with only the variável proposicional p and not involving $\sim$ which always takes the same truth value as $\sim p$? Give a truth table to show you’re right.

-NoValue-

content/many-valued-logic/three-valued-logics/goedel.tex

46.4 Gödel logics

mvl:thr:god:sec Kurt Gödel introduced a sequence of n -valued logics that each contain all fórmulas valid in intuitionistic logic, and are contained in classical logic. Here is the first interesting one:

mvl:thr:god:defn:goedel **Definição 46.6.** 3-valued Gödel logic $\mathbf{G}$ is defined using the matrix:

- 1. The standard propositional language $\mathcal{L}_0$ with $\perp, \neg, \wedge, \vee, \rightarrow$.
- 2. The set of truth values $V = \{\mathbb{T}, \mathbb{U}, \mathbb{F}\}$.
- 3. $\mathbb{T}$ is the only designated value, O.s., $V^+ = \{\mathbb{T}\}$.
- 4. For $\perp$, we have $\tilde{\perp} = \mathbb{F}$. Truth functions for the remaining connectives are given by the following tables:

$\tilde{\neg}_{\mathbf{G}}$		$\tilde{\wedge}_{\mathbf{G}}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{T}$	$\mathbb{F}$	$\mathbb{T}$	$\mathbb{T}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{F}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{U}$	$\mathbb{F}$
$\mathbb{F}$	$\mathbb{T}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$	$\mathbb{F}$

$\tilde{\vee}_{\mathbf{G}}$	T	U	F	$\tilde{\rightarrow}_{\mathbf{G}}$	T	U	F
T	T	T	T	T	T	U	F
U	T	U	U	U	T	T	F
F	T	U	F	F	T	T	T

You'll notice that the truth tables for $\wedge$ and $\vee$ are the same as in Łukasiewicz and strong Kleene logic, but the truth tables for $\neg$ and $\rightarrow$ differ for each. In Gödel logic, $\tilde{\neg}(U) = F$. In contrast to Łukasiewicz logic and Kleene logic, $\tilde{\rightarrow}(U, F) = F$; in contrast to Kleene logic (but as in Łukasiewicz logic), $\tilde{\rightarrow}(U, U) = T$.

As the connection to intuitionistic logic alluded to above suggests, $\mathbf{G}_3$ is close to intuitionistic logic. All intuitionistic truths are tautologies in $\mathbf{G}_3$, and many classical tautologies that are not valid intuitionistically also fail to be tautologies in $\mathbf{G}_3$. For instance, the following are not tautologies:

$$\begin{array}{ll}
 p \vee \neg p & (p \rightarrow q) \rightarrow (\neg p \vee q) \\
 \neg \neg p \rightarrow p & \neg(\neg p \wedge \neg q) \rightarrow (p \vee q) \\
 ((p \rightarrow q) \rightarrow p) \rightarrow p & \neg(p \rightarrow q) \rightarrow (p \wedge \neg q)
 \end{array}$$

However, not every tautology of $\mathbf{G}_3$ is also intuitionistically valid, p.ex., $\neg \neg p \vee \neg p$ or $(p \rightarrow q) \vee (q \rightarrow p)$.

Problema 46.8. Give truth tables to show that the following are tautologies of $\mathbf{G}_3$:

$$\begin{array}{l}
 \neg \neg p \vee \neg p \\
 (p \rightarrow q) \vee (q \rightarrow p) \\
 \neg(p \wedge q) \rightarrow (\neg p \vee \neg q) \\
 (p \rightarrow q) \vee (q \rightarrow r) \vee (r \rightarrow s)
 \end{array}$$

Problema 46.9. Give truth tables that show that the following are not tautologies of $\mathbf{G}_3$

$$\begin{array}{l}
 (p \rightarrow q) \rightarrow (\neg p \vee q) \\
 \neg(\neg p \wedge \neg q) \rightarrow (p \vee q) \\
 ((p \rightarrow q) \rightarrow p) \rightarrow p \\
 \neg(p \rightarrow q) \rightarrow (p \wedge \neg q)
 \end{array}$$

Problema 46.10. Which of the following relations hold in Gödel logic? Give a truth table for each.

1. $p, p \rightarrow q \models q$
2. $p \vee q, \neg p \models q$
3. $p \wedge q \models p$

46.5. DESIGNATING NOT JUST $\mathbb{T}$

$$4. p \models p \wedge p$$

$$5. p \models p \vee q$$

-NoValue-

`content/many-valued-logic/three-valued-logics/multiple-designation.tex`

46.5 Designating not just $\mathbb{T}$

mul:thr:mvl:sec

So far the logics we've seen all had the set of designated truth values $V^+ = \{\mathbb{T}\}$, O.s., something counts as true iff its truth value is $\mathbb{T}$. But one might also count something as true if it's just not $\mathbb{F}$. Then one would get a logic by stipulating in the matrix, p.ex., that $V^+ = \{\mathbb{T}, \mathbb{U}\}$.

Definição 46.7. The *logic of paradox* **LP** is defined using the matrix:

1. The standard propositional language $\mathcal{L}_0$ with $\neg, \wedge, \vee, \rightarrow$.
2. The set of truth values $V = \{\mathbb{T}, \mathbb{U}, \mathbb{F}\}$.
3. $\mathbb{T}$ and $\mathbb{U}$ are designated, O.s., $V^+ = \{\mathbb{T}, \mathbb{U}\}$.
4. Truth functions are the same as in strong Kleene logic.

Definição 46.8. Halldén's *logic of nonsense* **Hal** is defined using the matrix:

1. The standard propositional language $\mathcal{L}_0$ with $\neg, \wedge, \vee, \rightarrow$ and a 1-place connective $+$.
2. The set of truth values $V = \{\mathbb{T}, \mathbb{U}, \mathbb{F}\}$.
3. $\mathbb{T}$ and $\mathbb{U}$ are designated, O.s., $V^+ = \{\mathbb{T}, \mathbb{U}\}$.
4. Truth functions are the same as weak Kleene logic, plus the “is meaningless” operator:

$\tilde{+}$	
$\mathbb{T}$	$\mathbb{F}$
$\mathbb{U}$	$\mathbb{T}$
$\mathbb{F}$	$\mathbb{F}$

By contrast to the Kleene logics with which they share truth tables, these *do* have tautologies.

mul:thr:mvl:prop:LP-taut-CL

Proposição 46.9. *The tautologies of **LP** are the same as the tautologies of classical propositional logic.*

Prova. By **Proposição 45.12**, if $\models_{\mathbf{LP}} \varphi$ then $\models_{\mathbf{C}} \varphi$. To show the reverse, we show that if there is a **valoração** $\mathbf{v}: \text{At}_0 \rightarrow \{\mathbb{F}, \mathbb{T}, \mathbb{U}\}$ such that $\bar{\mathbf{v}}_{\mathbf{Ks}}(\varphi) = \mathbb{F}$ then there is a **valoração** $\mathbf{v}': \text{At}_0 \rightarrow \{\mathbb{F}, \mathbb{T}\}$ such that $\bar{\mathbf{v}}'_{\mathbf{C}}(\varphi) = \mathbb{F}$. This establishes the result for **LP**, since **Ks** and **LP** have the same characteristic truth functions, and $\mathbb{F}$ is the only truth value of **LP** that is not designated (that is the only difference between **LP** and **Ks**). Thus, if $\not\models_{\mathbf{LP}} \varphi$, for some **valoração** $\mathbf{v}$, $\bar{\mathbf{v}}_{\mathbf{LP}}(\varphi) = \bar{\mathbf{v}}_{\mathbf{Ks}}(\varphi) = \mathbb{F}$. By the claim we're proving, $\bar{\mathbf{v}}'_{\mathbf{C}}(\varphi) = \mathbb{F}$, O.s., $\not\models_{\mathbf{C}} \varphi$.

To establish the claim, we first define $\mathbf{v}'$ as

$$\mathbf{v}'(p) = \begin{cases} \mathbb{T} & \text{if } \mathbf{v}(p) \in \{\mathbb{T}, \mathbb{U}\} \\ \mathbb{F} & \text{otherwise} \end{cases}$$

We now show by induction on φ that (a) if $\bar{\mathbf{v}}_{\mathbf{Ks}}(\varphi) = \mathbb{F}$ then $\bar{\mathbf{v}}'_{\mathbf{C}}(\varphi) = \mathbb{F}$, and (b) if $\bar{\mathbf{v}}_{\mathbf{Ks}}(\varphi) = \mathbb{T}$ then $\bar{\mathbf{v}}'_{\mathbf{C}}(\varphi) = \mathbb{T}$

1. Induction basis: $\varphi \equiv p$. By **Definição 45.8**, $\bar{\mathbf{v}}_{\mathbf{Ks}}(\varphi) = \mathbf{v}(p) = \bar{\mathbf{v}}'_{\mathbf{C}}(\varphi)$, which implies both (a) and (b).

For the induction step, consider the cases:

2. $\varphi \equiv \neg\psi$.
 - a) Suppose $\bar{\mathbf{v}}_{\mathbf{Ks}}(\neg\psi) = \mathbb{F}$. By the definition of $\tilde{\mathbf{v}}_{\mathbf{Ks}}$, $\bar{\mathbf{v}}_{\mathbf{Ks}}(\psi) = \mathbb{T}$. By inductive hypothesis, case (b), we get $\bar{\mathbf{v}}'_{\mathbf{C}}(\psi) = \mathbb{T}$, so $\bar{\mathbf{v}}'_{\mathbf{C}}(\neg\psi) = \mathbb{F}$.
 - b) Suppose $\bar{\mathbf{v}}_{\mathbf{Ks}}(\neg\psi) = \mathbb{T}$. By the definition of $\tilde{\mathbf{v}}_{\mathbf{Ks}}$, $\bar{\mathbf{v}}_{\mathbf{Ks}}(\psi) = \mathbb{F}$. By inductive hypothesis, case (a), we get $\bar{\mathbf{v}}'_{\mathbf{C}}(\psi) = \mathbb{F}$, so $\bar{\mathbf{v}}'_{\mathbf{C}}(\neg\psi) = \mathbb{T}$.
3. $\varphi \equiv (\psi \wedge \chi)$.
 - a) Suppose $\bar{\mathbf{v}}_{\mathbf{Ks}}(\psi \wedge \chi) = \mathbb{F}$. By the definition of $\tilde{\wedge}_{\mathbf{Ks}}$, $\bar{\mathbf{v}}_{\mathbf{Ks}}(\psi) = \mathbb{F}$ or $\bar{\mathbf{v}}_{\mathbf{Ks}}(\chi) = \mathbb{F}$. By inductive hypothesis, case (a), we get $\bar{\mathbf{v}}'_{\mathbf{C}}(\psi) = \mathbb{F}$ or $\bar{\mathbf{v}}'_{\mathbf{C}}(\chi) = \mathbb{F}$, so $\bar{\mathbf{v}}'_{\mathbf{C}}(\psi \wedge \chi) = \mathbb{F}$.
 - b) Suppose $\bar{\mathbf{v}}_{\mathbf{Ks}}(\psi \wedge \chi) = \mathbb{T}$. By the definition of $\tilde{\wedge}_{\mathbf{Ks}}$, $\bar{\mathbf{v}}_{\mathbf{Ks}}(\psi) = \mathbb{T}$ and $\bar{\mathbf{v}}_{\mathbf{Ks}}(\chi) = \mathbb{T}$. By inductive hypothesis, case (b), we get $\bar{\mathbf{v}}'_{\mathbf{C}}(\psi) = \mathbb{T}$ and $\bar{\mathbf{v}}'_{\mathbf{C}}(\chi) = \mathbb{T}$, so $\bar{\mathbf{v}}'_{\mathbf{C}}(\psi \wedge \chi) = \mathbb{T}$.

The other two cases are similar, and left as exercises. Alternatively, the proof above establishes the result for all **fórmulas** only containing $\neg$ and $\wedge$. One may now appeal to the facts that in both **Ks** and **C**, for any $\mathbf{v}$, $\bar{\mathbf{v}}(\psi \vee \chi) = \bar{\mathbf{v}}(\neg(\neg\psi \wedge \neg\chi))$ and $\bar{\mathbf{v}}(\psi \rightarrow \chi) = \bar{\mathbf{v}}(\neg(\psi \wedge \neg\chi))$. $\square$

Problema 46.11. Complete the proof **Proposição 46.9**, O.s., establish (a) and (b) for the cases where $\varphi \equiv (\psi \vee \chi)$ and $\varphi \equiv (\psi \rightarrow \chi)$.

Problema 46.12. Prove that every classical tautology is a tautology in **Hal**.

Although they have the same tautologies as classical logic, their consequence relations are different. **LP**, for instance, is *paraconsistent* in that $\neg p, p \not\models q$, and so the principle of explosion $\neg\varphi, \varphi \models \psi$ does not hold in general. (It holds for some cases of φ and ψ , p.ex., if ψ is a tautology.)

Problema 46.13. Which of the following relations hold in (a) **LP** and in (b) **Hal**? Give a truth table for each.

1. $p, p \rightarrow q \models q$
2. $\neg q, p \rightarrow q \models \neg p$
3. $p \vee q, \neg p \models q$
4. $\neg p, p \models q$
5. $p \models p \vee q$
6. $p \rightarrow q, q \rightarrow r \models p \rightarrow r$

What if you make $\mathbb{U}$ designated in **L**₃?

Definição 46.10. The logic *3-valued R-Mingle* **RM**₃ is defined using the matrix:

1. The standard propositional language $\mathcal{L}_0$ with $\perp, \neg, \wedge, \vee, \rightarrow$.
2. The set of truth values $V = \{\mathbb{T}, \mathbb{U}, \mathbb{F}\}$.
3. $\mathbb{T}$ and $\mathbb{U}$ are designated, O.s., $V^+ = \{\mathbb{T}, \mathbb{U}\}$.
4. Truth functions are the same as Łukasiewicz logic **L**₃.

Problema 46.14. Which of the following relations hold in **RM**₃?

1. $p, p \rightarrow q \models q$
2. $p \vee q, \neg p \models q$
3. $\neg p, p \models q$
4. $p \models p \vee q$

Different truth tables can sometimes generate the same logic (entailment relation) just by changing the designated values. P.ex., this happens if in Gödel logic we take $V^+ = \{\mathbb{T}, \mathbb{U}\}$ instead of $\{\mathbb{T}\}$.

mul:thr:mul:
prop:gl-udes **Proposição 46.11.** *The matrix with $V = \{\mathbb{F}, \mathbb{U}, \mathbb{T}\}$, $V^+ = \{\mathbb{T}, \mathbb{U}\}$, and the truth functions of 3-valued Gödel logic defines classical logic.*

Prova. Exercise. □

Problema 46.15. Prove **Proposição 46.11** by showing that for the logic **L** defined just like Gödel logic but with $V^+ = \{\mathbb{T}, \mathbb{U}\}$, if $\Gamma \not\models_{\mathbf{L}} \psi$ then $\Gamma \not\models_{\mathbf{C}} \psi$. Use the ideas of **Proposição 46.9**, except instead of proving properties (a) and (b), show that $\bar{\mathbf{v}}_{\mathbf{G}}(\varphi) = \mathbb{F}$ iff $\bar{\mathbf{v}}'_{\mathbf{C}}(\varphi) = \mathbb{F}$ (and hence that $\bar{\mathbf{v}}_{\mathbf{G}}(\varphi) \in \{\mathbb{T}, \mathbb{U}\}$ iff $\bar{\mathbf{v}}'_{\mathbf{C}}(\varphi) = \mathbb{T}$). Explain why this establishes the proposition.

Capítulo 47

Infinite-valued Logics

-NoValue-

`content/many-valued-logic/infinite-valued-logics/introduction.tex`

47.1 Introduction

The number of truth values of a matrix need not be finite. An obvious choice for a set of infinitely many truth values is the set of rational numbers between 0 and 1, $V_\infty = [0, 1] \cap \mathbb{Q}$, O.s., mvl:inf:int:sec

$$V_\infty = \left\{ \frac{n}{m} : n, m \in \mathbb{N} \text{ and } n \leq m \right\}.$$

When considering this infinite truth value set, it is often useful to also consider the subsets

$$V_m = \left\{ \frac{n}{m-1} : n \in \mathbb{N} \text{ and } n \leq m \right\}$$

For instance, V_5 is the set with 5 evenly spaced truth values,

$$V_5 = \left\{ 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1 \right\}.$$

In logics based on these truth value sets, usually only 1 is designated, O.s., $V^+ = \{1\}$. In other words, we let 1 play the role of (absolute) truth, 0 as absolute falsity, but fórmulas may take any intermediate value in V .

One can also consider the set $V_{[0,1]} = [0, 1]$ of all *real* numbers between 0 and 1, or other infinite subsets of $[0, 1]$, however. Logics with this truth value set are often called *fuzzy*.

-NoValue-

`content/many-valued-logic/infinite-valued-logics/lukasiewicz.tex`

47.2. LUKASIEWICZ LOGIC

47.2 Łukasiewicz logic

mv:inf:luk:
sec

This is a short “stub” of a section on infinite-valued Łukasiewicz logic.

mv:inf:luk:
def:lukasiewicz

Definição 47.1. Infinite-valued Łukasiewicz logic $\mathbf{L}_\infty$ is defined using the matrix:

1. The standard propositional language $\mathcal{L}_0$ with $\neg, \wedge, \vee, \rightarrow$.
2. The set of truth values V_∞ .
3. 1 is the only designated value, O.s., $V^+ = \{1\}$.
4. Truth functions are given by the following functions:

$$\begin{aligned}\neg_{\mathbf{L}}(x) &= 1 - x \\ \wedge_{\mathbf{L}}(x, y) &= \min(x, y) \\ \vee_{\mathbf{L}}(x, y) &= \max(x, y) \\ \rightarrow_{\mathbf{L}}(x, y) &= \min(1, 1 - (x - y)) = \begin{cases} 1 & \text{if } x \leq y \\ 1 - (x - y) & \text{otherwise.} \end{cases}\end{aligned}$$

m -valued Łukasiewicz logic is defined the same, except $V = V_m$.

Proposição 47.2. The logic $\mathbf{L}_3$ defined by *Definição 46.1* is the same as $\mathbf{L}_3$ defined by *Definição 47.1*.

Prova. This can be seen by comparing the truth tables for the connectives given in *Definição 46.1* with the truth tables determined by the equations in *Definição 47.1*:

$\neg$		$\wedge_{\mathbf{L}_3}$	1	1/2	0
1	0	1	1	1/2	0
1/2	1/2	1/2	1/2	1/2	0
0	1	0	0	0	0

$\vee_{\mathbf{L}_3}$	1	1/2	0	$\rightarrow_{\mathbf{L}_3}$	1	1/2	0
1	1	1	1	1	1	1/2	0
1/2	1	1/2	1/2	1/2	1	1	1/2
0	1	1/2	0	0	1	1	1

□

mv:inf:luk:
prop:luk-infity-m

Proposição 47.3. If $\Gamma \models_{\mathbf{L}_\infty} \psi$ then $\Gamma \models_{\mathbf{L}_m} \psi$ for all $m \geq 2$.

Prova. Exercise.

□

Problema 47.1. Prove [Proposição 47.3](#).

In fact, the converse holds as well.

Infinite-valued Łukasiewicz logic is the most popular fuzzy logic. In the fuzzy logic literature, the conditional is often defined as $\neg\varphi \vee \psi$. The result would be an infinite-valued strong Kleene logic.

Problema 47.2. Show that $(p \rightarrow q) \vee (q \rightarrow p)$ is a tautology of $\mathbf{L}_\infty$.

-NoValue-

[content/many-valued-logic/infinite-valued-logics/goedel.tex](#)

47.3 Gödel logics

[mvl:inf:god:](#)
[sec](#)

This is a short “stub” of a section on infinite-valued Gödel logic.

Definição 47.4. Infinite-valued Gödel logic $\mathbf{G}_\infty$ is defined using the matrix:

[mvl:inf:god:](#)
[def:goedel](#)

1. The standard propositional language $\mathcal{L}_0$ with $\perp, \neg, \wedge, \vee, \rightarrow$.
2. The set of truth values V_∞ .
3. 1 is the only designated value, O.s., $V^+ = \{1\}$.
4. Truth functions are given by the following functions:

$$\begin{aligned}\tilde{\perp} &= 0 \\ \tilde{\neg}_{\mathbf{G}}(x) &= \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases} \\ \tilde{\wedge}_{\mathbf{G}}(x, y) &= \min(x, y) \\ \tilde{\vee}_{\mathbf{G}}(x, y) &= \max(x, y) \\ \tilde{\rightarrow}_{\mathbf{G}}(x, y) &= \begin{cases} 1 & \text{if } x \leq y \\ y & \text{otherwise.} \end{cases}\end{aligned}$$

m -valued Gödel logic is defined the same, except $V = V_m$.

Proposição 47.5. The logic $\mathbf{G}_3$ defined by [Definição 46.6](#) is the same as $\mathbf{G}_3$ defined by [Definição 47.4](#).

Prova. This can be seen by comparing the truth tables for the connectives given in [Definição 46.6](#) with the truth tables determined by the equations in [Definição 47.4](#):

$$\begin{array}{c|c} \widetilde{\neg}_{\mathbf{G}_3} & \\ \hline 1 & 0 \\ 1/2 & 0 \\ 0 & 1 \end{array} \quad \begin{array}{c|c|c} \widetilde{\neg}_{\mathbf{G}} & 1 & 1/2 & 0 \\ \hline 1 & 1 & 1/2 & 0 \\ 1/2 & 1/2 & 1/2 & 0 \\ 0 & 0 & 0 & 0 \end{array}$$

mul:inf:god: **Proposição 47.6.** *If $\Gamma \models_{\mathbf{G}_\infty} \psi$ then $\Gamma \models_{\mathbf{G}_m} \psi$ for all $m \geq 2$.*
prop:god:infy-m

Prova. Exercise.

Problema 47.3. Prove **Proposição 47.6**.

In fact, the converse holds as well.

Like $\mathbf{G}_3$, $\mathbf{G}_\infty$ has all intuitionistically valid **fórmulas** as tautologies, and the same examples of non-tautologies are non-tautologies of $\mathbf{G}_\infty$:

$$\begin{array}{ll} p \vee \neg p & (p \rightarrow q) \rightarrow (\neg p \vee q) \\ \neg \neg p \rightarrow p & \neg(\neg p \wedge \neg q) \rightarrow (p \vee q) \\ ((p \rightarrow q) \rightarrow p) \rightarrow p & \neg(p \rightarrow q) \rightarrow (p \wedge \neg q) \end{array}$$

The example of an intuitionistically invalid **fórmula** that is nevertheless a tautology of $\mathbf{G}_3$, $(p \rightarrow q) \vee (q \rightarrow p)$, is also a tautology in $\mathbf{G}_\infty$. In fact, $\mathbf{G}_\infty$ can be characterized as intuitionistic logic to which the schema $(\varphi \rightarrow \psi) \vee (\psi \rightarrow \varphi)$ is added. This was shown by Michael Dummett, and so $\mathbf{G}_\infty$ is often referred to as Gödel–Dummett logic **LC**.

Problema 47.4. Show that $(p \rightarrow q) \vee (q \rightarrow p)$ is a tautology of $\mathbf{G}_\infty$.

Problem 47.5. Show that $(p \rightarrow q) \vee (q \rightarrow r) \vee (r \rightarrow s)$, which is a tautology of $\mathbf{G}_3$, is not a tautology of $\mathbf{G}_\infty$.

Capítulo 48

Sequent Calculus

-NoValue-

content/many-valued-logic/sequent-calculus/introduction.tex

48.1 Introduction

The sequent calculus for classical logic is an efficient and simple **derivação** system. If a many-valued logic is defined by a matrix with finitely many truth values, O.s., V is finite, it is possible to provide a sequent calculus for it. The idea for how to do this comes from considering the meanings of sequents and the form of inference rules in the classical case.

mvl:seq:int:
sec

Now recall that a sequent

$$\varphi_1, \dots, \varphi_n \Rightarrow \psi_1, \dots, \psi_n$$

can be interpreted as the **fórmula**

$$(\varphi_1 \wedge \dots \wedge \varphi_n) \rightarrow (\psi_1 \vee \dots \vee \psi_n)$$

In other words, **A valoração** $\mathbf{v}$ satisfies a sequent $\Gamma \Rightarrow \Delta$ iff either $\bar{\mathbf{v}}(\varphi) = \mathbb{F}$ for some $\varphi \in \Gamma$ or $\bar{\mathbf{v}}(\varphi) = \mathbb{T}$ for some $\varphi \in \Delta$. On this interpretation, initial sequents $\varphi \Rightarrow \varphi$ are always satisfied, because either $\bar{\mathbf{v}}(\varphi) = \mathbb{T}$ or $\bar{\mathbf{v}}(\varphi) = \mathbb{F}$.

Here are the inference rules for the conditional in **LK**, with side formulas Γ, Δ left out:

$\frac{\Rightarrow \varphi \quad \psi \Rightarrow}{\varphi \rightarrow \psi \Rightarrow} \rightarrow L$	$\frac{\varphi \Rightarrow \psi}{\Rightarrow \varphi \rightarrow \psi} \rightarrow R$

If we apply the above semantic interpretation of a sequent, we can read the $\rightarrow L$ rule as saying that if $\bar{\mathbf{v}}(\varphi) = \mathbb{T}$ and $\bar{\mathbf{v}}(\psi) = \mathbb{F}$, then $\bar{\mathbf{v}}(\varphi \rightarrow \psi) = \mathbb{F}$. Similarly, the $\rightarrow R$ rule says that if either $\bar{\mathbf{v}}(\varphi) = \mathbb{F}$ or $\bar{\mathbf{v}}(\psi) = \mathbb{T}$, then $\bar{\mathbf{v}}(\varphi \rightarrow \psi) = \mathbb{T}$. And in fact, these conditionals are actually biconditionals. In the case of the $\wedge L$ and $\vee R$ rules in their standard formulation, the corresponding conditionals would not be biconditionals. But there are alternative versions of these rules where they are:

$\frac{\varphi, \psi, \Gamma \Rightarrow \Delta}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta} \wedge L$	$\frac{\Gamma \Rightarrow \Delta, \varphi, \psi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi} \vee R$

This basic idea, applied to an n -valued logic, then results in a sequent calculus with n instead of two places, one for each truth value. For a three-valued logic with $V = \{\mathbb{F}, \mathbb{U}, \mathbb{T}\}$, a sequent is an expression $\Gamma \mid \Pi \mid \Delta$. It is satisfied in **a valoração** $\mathbf{v}$ iff either $\bar{\mathbf{v}}(\varphi) = \mathbb{F}$ for some $\varphi \in \Gamma$ or $\bar{\mathbf{v}}(\varphi) = \mathbb{T}$ for some $\varphi \in \Delta$ or $\bar{\mathbf{v}}(\varphi) = \mathbb{U}$ for some $\varphi \in \Pi$. Consequently, initial sequents $\varphi \mid \varphi \mid \varphi$ are always satisfied.

-NoValue-

content/many-valued-logic/sequent-calculus/rules-and-proofs.tex

48.2 Rules and Derivações

mvl:seq:rul:
sec

For the following, let Γ, Δ, Π, A represent finite sequences of **sentenças**.

Definição 48.1 (Sequent). An n -sided *sequent* is an expression of the form

$$\Gamma_1 \mid \dots \mid \Gamma_n$$

where each Γ_i is a finite (possibly empty) sequences of **sentenças** of the language $\mathcal{L}$.

Definição 48.2 (Initial Sequent). An n -sided *initial sequent* is an n -sided sequent of the form $\varphi \mid \dots \mid \varphi$ for any **sentença** φ in the language.

If the language contains a 0-place connective $\star$, O.s., a propositional constant, then we also take the sequent $\dots \mid \star \mid \dots$ where $\star$ appears in the space for the truth value associated with $\tilde{\star} \in V$, and is empty otherwise.

For each connective of an n -valued logic $\mathbf{L}$, there is a logical rule for each truth value that this connective can take in $\mathbf{L}$. **Derivações** in an n -sided sequent calculus for $\mathbf{L}$ are trees of sequents, where the topmost sequents are initial sequents, and if a sequent stands below one or more other sequents, it must follow correctly by a rule of inference for the connectives of $\mathbf{L}$.

Definição 48.3 (Theorems). A **sentença** φ is a *theorem* of an n -valued logic $\mathbf{L}$ if there is a **derivação** of the n -sequent containing φ in each position corresponding to a designated truth value of $\mathbf{L}$. We write $\vdash_{\mathbf{L}} \varphi$ if φ is a theorem and $\not\vdash_{\mathbf{L}} \varphi$ if it is not.

Definição 48.4 (Derivabilidade). A **sentença** φ is *derivável* from a set of **sentenças** Γ in an n -valued logic $\mathbf{L}$, $\Gamma \vdash_{\mathbf{L}} \varphi$, iff there is a finite subset $\Gamma_0 \subseteq \Gamma$ and a sequence Γ'_0 of the **sentenças** in Γ_0 such that the following sequent has a **derivação**:

$$A_1 \mid \dots \mid A_n$$

where A_i is φ if position i corresponds to a designated truth value, and Γ'_0 otherwise. If φ is not **derivável** from Γ we write $\Gamma \not\vdash \varphi$.

For instance, 3-valued Łukasiewicz logic has a 3-sided sequent calculus. In a 3-sided sequent $\Gamma \mid \Pi \mid \Delta$, Γ corresponds to $\mathbb{F}$, Δ to $\mathbb{T}$, and Π to $\mathbb{U}$. Axioms are $\varphi \mid \varphi \mid \varphi$. Since only $\mathbb{T}$ is designated, $\Gamma \vdash_{\mathbf{L}_3} \varphi$ iff the sequent $\Gamma \mid \Gamma \mid \varphi$ has a **derivação**. (If $\mathbb{U}$ were also designated, we would need a **derivação** of $\Gamma \mid \varphi \mid \varphi$.)

-NoValue-

content/many-valued-logic/sequent-calculus/structural-rules.tex

48.3 Structural Rules

The structural rules for n -sided sequent calculus operate as in the classical case, except for each position i . mvl:seq:str:
sec

$$\frac{\Gamma_1 \mid \dots \mid \Gamma_i \mid \dots \mid \Gamma_n}{\Gamma_1 \mid \dots \mid \varphi, \Gamma_i \mid \dots \mid \Gamma_n} W_i$$

$$\frac{\Gamma_1 \mid \dots \mid \varphi, \varphi, \Gamma_i \mid \dots \mid \Gamma_n}{\Gamma_1 \mid \dots \mid \varphi, \Gamma_i \mid \dots \mid \Gamma_n} C_i$$

$$\frac{\Gamma_1 \mid \dots \mid \Gamma_i, \varphi, \psi, \Gamma'_i \mid \dots \mid \Gamma_n}{\Gamma_1 \mid \dots \mid \Gamma_i, \psi, \varphi, \Gamma'_i \mid \dots \mid \Gamma_n} X_i$$

A series of weakening, contraction, and exchange inferences will often be indicated by double inference lines.

The Cut rule comes in several forms, one for every combination of distinct positions in the sequent $i \neq j$:

$$\frac{\Gamma_1 \mid \dots \mid \varphi, \Gamma_i \mid \dots \mid \Gamma_n \quad \Delta_1 \mid \dots \mid \varphi, \Delta_j \mid \dots \mid \Delta_n}{\Gamma_1, \Delta_1 \mid \dots \mid \Gamma_n, \Delta_n} \text{Cut}_{i,j}$$

-NoValue-

content/many-valued-logic/sequent-calculus/propositional-rules.tex

48.4 Propositional Rules for Selected Logics

The inference rules for a connective in an n -sided sequent calculus only depend on the characteristic truth function for the connective. Thus, if some connective is defined by the same truth function in different logics, these n -sided sequent rules for the connective are the same in those logics. mvl:seq:prl:
sec

Rules for $\neg$

The following rules for $\neg$ apply to Łukasiewicz and Kleene logics, and their variants.

48.4. PROPOSITIONAL RULES FOR SELECTED LOGICS

$$\frac{\Gamma \mid \Pi \mid \Delta, \varphi}{\neg\varphi, \Gamma \mid \Pi \mid \Delta} \neg\mathbb{F}$$

$$\frac{\Gamma \mid \varphi, \Pi \mid \Delta}{\Gamma \mid \neg\varphi, \Pi \mid \Delta} \neg\mathbb{U}$$

$$\frac{\varphi, \Gamma \mid \Pi \mid \Delta}{\Gamma \mid \Pi \mid \Delta, \neg\varphi} \neg\mathbb{T}$$

The following rules for $\neg$ apply to Gödel logic.

$$\frac{\Gamma \mid \varphi, \Pi \mid \Delta, \varphi}{\neg\varphi, \Gamma \mid \Pi \mid \Delta} \neg_{\mathbf{G}}\mathbb{F} \qquad \frac{\varphi, \Gamma \mid \Pi \mid \Delta}{\Gamma \mid \Pi \mid \Delta, \neg\varphi} \neg_{\mathbf{G}}\mathbb{T}$$

(In Gödel logic, $\neg\varphi$ can never take the value $\mathbb{U}$, so there is no rule for the middle position.)

Rules for $\wedge$

These are the rules for $\wedge$ in Łukasiewicz, strong Kleene, and Gödel logic.

$$\frac{\varphi, \psi, \Gamma \mid \Pi \mid \Delta}{\varphi \wedge \psi, \Gamma \mid \Pi \mid \Delta} \wedge\mathbb{F}$$

$$\frac{\Gamma \mid \varphi, \Pi \mid \varphi, \Delta \quad \Gamma \mid \psi, \Pi \mid \psi, \Delta \quad \Gamma \mid \varphi, \psi, \Pi \mid \Delta}{\Gamma \mid \varphi \wedge \psi, \Pi \mid \Delta} \wedge\mathbb{U}$$

$$\frac{\Gamma \mid \Pi \mid \Delta, \varphi \quad \Gamma \mid \Pi \mid \Delta, \psi}{\Gamma \mid \Pi \mid \Delta, \varphi \wedge \psi} \wedge\mathbb{T}$$

Rules for $\vee$

These are the rules for $\vee$ in Łukasiewicz, strong Kleene, and Gödel logic.

$$\frac{\varphi, \Gamma \mid \Pi \mid \Delta \quad \psi, \Gamma \mid \Pi \mid \Delta}{\varphi \vee \psi, \Gamma \mid \Pi \mid \Delta} \vee\mathbb{F}$$

$$\frac{\varphi, \Gamma \mid \varphi, \Pi \mid \Delta \quad \psi, \Gamma \mid \psi, \Pi \mid \Delta \quad \Gamma \mid \varphi, \psi, \Pi \mid \Delta}{\Gamma \mid \varphi \vee \psi, \Pi \mid \Delta} \vee\mathbb{U}$$

$$\frac{\Gamma \mid \Pi \mid \Delta, \varphi, \psi}{\Gamma \mid \Pi \mid \Delta, \varphi \vee \psi} \vee \mathbb{T}$$

Rules for $\rightarrow$

These are the rules for $\rightarrow$ in Łukasiewicz logic.

$$\begin{array}{c} \frac{\Gamma \mid \Pi \mid \Delta, \varphi \quad \psi, \Gamma \mid \Pi \mid \Delta}{\varphi \rightarrow \psi, \Gamma \mid \Pi \mid \Delta} \rightarrow_{\mathbf{L}_3} \mathbb{F} \\[10pt] \frac{\Gamma \mid \varphi, \psi, \Pi \mid \Delta \quad \psi, \Gamma \mid \Pi \mid \Delta, \varphi}{\Gamma \mid \varphi \rightarrow \psi, \Pi \mid \Delta} \rightarrow_{\mathbf{L}_3} \mathbb{U} \\[10pt] \frac{\varphi, \Gamma \mid \psi, \Pi \mid \Delta, \psi \quad \varphi, \Gamma \mid \varphi, \Pi \mid \Delta, \psi}{\Gamma \mid \Pi \mid \Delta, \varphi \rightarrow \psi} \rightarrow_{\mathbf{L}_3} \mathbb{T} \end{array}$$

These are the rules for $\rightarrow$ in strong Kleene logic.

$$\begin{array}{c} \frac{\Gamma \mid \Pi \mid \Delta, \varphi \quad \psi, \Gamma \mid \Pi \mid \Delta}{\varphi \rightarrow \psi, \Gamma \mid \Pi \mid \Delta} \rightarrow_{\mathbf{Ks}} \mathbb{F} \\[10pt] \frac{\psi, \Gamma \mid \psi, \Pi \mid \Delta \quad \Gamma \mid \varphi, \psi, \Pi \mid \Delta \quad \Gamma \mid \varphi, \Pi \mid \Delta, \varphi}{\Gamma \mid \varphi \rightarrow \psi, \Pi \mid \Delta} \rightarrow_{\mathbf{Ks}} \mathbb{U} \\[10pt] \frac{\varphi, \Gamma \mid \Pi \mid \Delta, \psi}{\Gamma \mid \Pi \mid \Delta, \varphi \rightarrow \psi} \rightarrow_{\mathbf{Ks}} \mathbb{T} \end{array}$$

These are the rules for $\rightarrow$ in Gödel logic.

$$\begin{array}{c} \frac{\Gamma \mid \varphi, \Pi \mid \Delta, \varphi \quad \psi, \Gamma \mid \Pi \mid \Delta}{\varphi \rightarrow \psi, \Gamma \mid \Pi \mid \Delta} \rightarrow_{\mathbf{G}_3} \mathbb{F} \\[10pt] \frac{\Gamma \mid \psi, \Pi \mid \Delta \quad \Gamma \mid \Pi \mid \Delta, \varphi}{\Gamma \mid \varphi \rightarrow \psi, \Pi \mid \Delta} \rightarrow_{\mathbf{G}_3} \mathbb{U} \\[10pt] \frac{\varphi, \Gamma \mid \psi, \Pi \mid \Delta, \psi \quad \varphi, \Gamma \mid \varphi, \Pi \mid \Delta, \psi}{\Gamma \mid \Pi \mid \Delta, \varphi \rightarrow \psi} \rightarrow_{\mathbf{G}_3} \mathbb{T} \end{array}$$

Release : 51c2271 (2026-07-12)

677

Figura 48.1: Example derivação in L_3

Parte XI

Normal Modal Logics

This part covers the metatheory of normal modal logics. It currently consists of Aldo Antonelli's notes on classical correspondence theory for basic modal logic.

Capítulo 49

Syntax and Semantics

-NoValue-

[content/normal-modal-logic/syntax-and-semantics/introduction.tex](#)

49.1 Introduction

Modal logic deals with *modal propositions* and the entailment relations among them. Examples of modal propositions are the following: nml:syn:int:
sec

1. It is necessary that $2 + 2 = 4$.
2. It is necessarily possible that it will rain tomorrow.
3. If it is necessarily possible that φ then it is possible that φ .

Possibility and necessity are not the only modalities: other unary connectives are also classified as modalities, for instance, “it ought to be the case that φ ,” “It will be the case that φ ,” “Dana knows that φ ,” or “Dana believes that φ .”

Modal logic makes its first appearance in Aristotle's *De Interpretatione*: he was the first to notice that necessity implies possibility, but not vice versa; that possibility and necessity are inter-definable; that If $\varphi \wedge \psi$ is possibly true then φ is possibly true and ψ is possibly true, but not conversely; and that if $\varphi \rightarrow \psi$ is necessary, then if φ is necessary, so is ψ .

49.1. INTRODUCTION

The first modern approach to modal logic was the work of C. I. Lewis, culminating with Lewis and Langford, *Symbolic Logic* (1932). Lewis & Langford were unhappy with the representation of implication by means of the material conditional: $\varphi \rightarrow \psi$ is a poor substitute for “ φ implies ψ .” Instead, they proposed to characterize implication as “Necessarily, if φ then ψ ,” symbolized as $\varphi \rightarrow \psi$. In trying to sort out the different properties, Lewis identified five different modal systems, **S1**, $\dots$, **S4**, **S5**, the last two of which are still in use.

The approach of Lewis and Langford was purely *syntactical*: they identified reasonable axioms and rules and investigated what was provable with those means. A semantic approach remained elusive for a long time, until a first attempt was made by Rudolf Carnap in *Meaning and Necessity* (1947) using the notion of a *state description*, O.s., a collection of atomic sentences (those that are “true” in that state description). After lifting the truth definition to arbitrary sentences φ , Carnap defines φ to be *necessarily true* if it is true in all state descriptions. Carnap’s approach could not handle *iterated* modalities, in that sentences of the form “Possibly necessarily $\dots$ possibly φ ” always reduce to the innermost modality.

The major breakthrough in modal semantics came with Saul Kripke’s article “A Completeness Theorem in Modal Logic” (JSL 1959). Kripke based his work on Leibniz’s idea that a statement is necessarily true if it is true “at all possible worlds.” This idea, though, suffers from the same drawbacks as Carnap’s, in that the truth of statement at a world w (or a state description s) does not depend on w at all. So Kripke assumed that worlds are related by an *accessibility relation* R , and that a statement of the form “Necessarily φ ” is true at a world w if and only if φ is true at all worlds w' *accessible from* w . Semantics that provide some version of this approach are called Kripke semantics and made possible the tumultuous development of modal logics (in the plural).

When interpreted by the Kripke semantics, modal logic shows us what *relational structures* look like “from the inside.” A relational structure is just a set equipped with a binary relation (for instance, the set of students in the class ordered by their social security number is a relational structure). But in fact relational structures come in all sorts of domains: besides relative possibility of states of the world, we can have epistemic states of some agent related by epistemic possibility, or states of a dynamical system with their state transitions, etc. Modal logic can be used to model all of these: the first gives us ordinary, alethic, modal logic; the others give us epistemic logic, dynamic logic, etc.

We focus on one particular angle, known to modal logicians as “correspondence theory.” One of the most significant early discoveries of Kripke’s is that many properties of the accessibility relation R (whether it is transitive, symmetric, etc.) can be characterized *in the modal language* itself by means of appropriate “modal schemas.” Modal logicians say, for instance, that the reflexivity of R “corresponds” to the schema “If necessarily φ , then φ ”. We explore mainly the correspondence theory of a number of classical systems of modal logic (p.ex., **S4** and **S5**) obtained by a combination of the schemas D,

T, B, 4, and 5.

-NoValue-

content/normal-modal-logic/syntax-and-semantics/language-modal-logic.tex

49.2 The Language of Basic Modal Logic

nml:syn:lan:
sec

Definição 49.1. The basic language of modal logic contains

1. The propositional constant for *falsidade* $\perp$.
2. The propositional constant for *verdade* $\top$.
3. A *infinitos enumeráveis* set of *variáveis proposicionais*: $p_0, p_1, p_2, \dots$
4. The propositional connectives: $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\rightarrow$ (*condicional*), $\leftrightarrow$ (*bicondicional*).
5. The modal operator $\Box$.
6. The modal operator $\Diamond$.

Definição 49.2. *Fórmulas* of the basic modal language are inductively defined as follows:

1. $\perp$ is an atomic *fórmula*.
2. $\top$ is an atomic *fórmula*.
3. Every *variável proposicional* p_i is an (atomic) *fórmula*.
4. If φ is a *fórmula*, then $\neg\varphi$ is a *fórmula*.
5. If φ and ψ are *fórmulas*, then $(\varphi \wedge \psi)$ is a *fórmula*.
6. If φ and ψ are *fórmulas*, then $(\varphi \vee \psi)$ is a *fórmula*.
7. If φ and ψ are *fórmulas*, then $(\varphi \rightarrow \psi)$ is a *fórmula*.
8. If φ and ψ are *fórmulas*, then $(\varphi \leftrightarrow \psi)$ is a *fórmula*.
9. If φ is a *fórmula*, then $\Box\varphi$ is a *fórmula*.
10. If φ is a *fórmula*, then $\Diamond\varphi$ is a *fórmula*.
11. Nothing else is a *fórmula*.

If a *fórmula* φ does not contain $\Box$ or $\Diamond$, we say it is *modal-free*.

-NoValue-

content/normal-modal-logic/syntax-and-semantics/substitution.tex

49.3. SIMULTANEOUS SUBSTITUTION

49.3 Simultaneous Substitution

nml:syn:sub:sec An *instance* of a **fórmula** φ is the result of replacing all occurrences of a **variável proposicional** in φ by some other **fórmula**. We will refer to instances of **fórmulas** often, both when discussing validity and when discussing **derivabilidade**. It therefore is useful to define the notion precisely.

nml:syn:sub: def:subst-inst **Definição 49.3.** Where φ is a modal **fórmula** all of whose **variáveis proposicionais** are among $p_1, \dots, p_n$, and $\theta_1, \dots, \theta_n$ are also modal **fórmulas**, we define $\varphi[\theta_1/p_1, \dots, \theta_n/p_n]$ as the result of simultaneously substituting each θ_i for p_i in φ . Formally, this is a definition by induction on φ :

1. $\varphi \equiv \perp$: $\varphi[\theta_1/p_1, \dots, \theta_n/p_n]$ is $\perp$.
2. $\varphi \equiv \top$: $\varphi[\theta_1/p_1, \dots, \theta_n/p_n]$ is $\top$.
3. $\varphi \equiv q$: $\varphi[\theta_1/p_1, \dots, \theta_n/p_n]$ is q , provided $q \neq p_i$ for $i = 1, \dots, n$.
4. $\varphi \equiv p_i$: $\varphi[\theta_1/p_1, \dots, \theta_n/p_n]$ is θ_i .
5. $\varphi \equiv \neg\psi$: $\varphi[\theta_1/p_1, \dots, \theta_n/p_n]$ is $\neg\psi[\theta_1/p_1, \dots, \theta_n/p_n]$.
6. $\varphi \equiv (\psi \wedge \chi)$: $\varphi[\theta_1/p_1, \dots, \theta_n/p_n]$ is

$$(\psi[\theta_1/p_1, \dots, \theta_n/p_n] \wedge \chi[\theta_1/p_1, \dots, \theta_n/p_n]).$$
7. $\varphi \equiv (\psi \vee \chi)$: $\varphi[\theta_1/p_1, \dots, \theta_n/p_n]$ is

$$(\psi[\theta_1/p_1, \dots, \theta_n/p_n] \vee \chi[\theta_1/p_1, \dots, \theta_n/p_n]).$$
8. $\varphi \equiv (\psi \rightarrow \chi)$: $\varphi[\theta_1/p_1, \dots, \theta_n/p_n]$ is

$$(\psi[\theta_1/p_1, \dots, \theta_n/p_n] \rightarrow \chi[\theta_1/p_1, \dots, \theta_n/p_n]).$$
9. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\varphi[\theta_1/p_1, \dots, \theta_n/p_n]$ is

$$(\psi[\theta_1/p_1, \dots, \theta_n/p_n] \leftrightarrow \chi[\theta_1/p_1, \dots, \theta_n/p_n]).$$
10. $\varphi \equiv \Box\psi$: $\varphi[\theta_1/p_1, \dots, \theta_n/p_n]$ is $\Box\psi[\theta_1/p_1, \dots, \theta_n/p_n]$.
11. $\varphi \equiv \Diamond\psi$: $\varphi[\theta_1/p_1, \dots, \theta_n/p_n]$ is $\Diamond\psi[\theta_1/p_1, \dots, \theta_n/p_n]$.

The **fórmula** $\varphi[\theta_1/p_1, \dots, \theta_n/p_n]$ is called a *substitution instance* of φ .

Exemplo 49.4. Suppose φ is $p_1 \rightarrow \Box(p_1 \wedge p_2)$, θ_1 is $\Diamond(p_2 \rightarrow p_3)$ and θ_2 is $\neg\Box p_1$. Then $\varphi[\theta_1/p_1, \theta_2/p_2]$ is

$$\Diamond(p_2 \rightarrow p_3) \rightarrow \Box(\Diamond(p_2 \rightarrow p_3) \wedge \neg\Box p_1)$$

while $\varphi[\theta_2/p_1, \theta_1/p_2]$ is

$$\neg\Box p_1 \rightarrow \Box(\neg\Box p_1 \wedge \Diamond(p_2 \rightarrow p_3))$$

Note that simultaneous substitution is in general not the same as iterated substitution, p.ex., compare $\varphi[\theta_1/p_1, \theta_2/p_2]$ above with $(\varphi[\theta_1/p_1])[\theta_2/p_2]$, which is:

$$\begin{aligned} \Diamond(p_2 \rightarrow p_3) &\rightarrow \Box(\Diamond(p_2 \rightarrow p_3) \wedge p_2)[\neg\Box p_1/p_2], \text{ O.s.,} \\ \Diamond(\neg\Box p_1 \rightarrow p_3) &\rightarrow \Box(\Diamond(\neg\Box p_1 \rightarrow p_3) \wedge \neg\Box p_1) \end{aligned}$$

and with $(\varphi[\theta_2/p_2])[\theta_1/p_1]$:

$$\begin{aligned} p_1 &\rightarrow \Box(p_1 \wedge \neg\Box p_1)[\Diamond(p_2 \rightarrow p_3)/p_1], \text{ O.s.,} \\ \Diamond(p_2 \rightarrow p_3) &\rightarrow \Box(\Diamond(p_2 \rightarrow p_3) \wedge \neg\Box\Diamond(p_2 \rightarrow p_3)). \end{aligned}$$

-NoValue-

`content/normal-modal-logic/syntax-and-semantics/relational-models.tex`

49.4 Relational Models

The basic concept of semantics for normal modal logics is that of a *relational model*. It consists of a set of worlds, which are related by a binary “accessibility relation,” together with an assignment which determines which **variáveis proposicionais** count as “true” at which worlds. nml:syn:rel:sec

Definição 49.5. A *model* for the basic modal language is a triple $\mathfrak{M} = \langle W, R, V \rangle$, where

1. W is a nonempty set of “worlds,”
2. R is a binary accessibility relation on W , and
3. V is a function assigning to each **variável proposicional** p a set $V(p)$ of possible worlds.

When Rww' holds, we say that w' is *accessible from* w . When $w \in V(p)$ we say p is *true at* w .

The great advantage of relational semantics is that models can be represented by means of simple diagrams, such as the one in **Figura 49.1**. Worlds are represented by nodes, and world w' is accessible from w precisely when there is an arrow from w to w' . Moreover, we label a node (world) by p when $w \in V(p)$, and otherwise by $\neg p$. **Figura 49.1** represents the model with $W = \{w_1, w_2, w_3\}$, $R = \{\langle w_1, w_2 \rangle, \langle w_1, w_3 \rangle\}$, $V(p) = \{w_1, w_2\}$, and $V(q) = \{w_2\}$.

-NoValue-

`content/normal-modal-logic/syntax-and-semantics/truth-at-w.tex`

49.5. TRUTH AT A WORLD

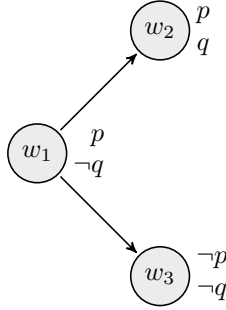


Figura 49.1: A simple model.

nml:syn:rel:
fig:simple

49.5 Truth at a World

nml:syn:trw:
sec

Every modal model determines which modal **fórmulas** count as true at which worlds in it. The relation “model $\mathfrak{M}$ makes **fórmula** φ true at world w ” is the basic notion of relational semantics. The relation is defined inductively and coincides with the usual characterization using truth tables for the non-modal operators.

nml:syn:trw:
defn:mmodels

Definição 49.6. *Truth of a **fórmula** φ at w in a $\mathfrak{M}$, in symbols: $\mathfrak{M}, w \Vdash \varphi$, is defined inductively as follows:*

1. $\varphi \equiv \perp$: Never $\mathfrak{M}, w \Vdash \perp$.
2. $\varphi \equiv \top$: Always $\mathfrak{M}, w \Vdash \top$.
3. $\mathfrak{M}, w \Vdash p$ iff $w \in V(p)$.
4. $\varphi \equiv \neg\psi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \not\Vdash \psi$.
5. $\varphi \equiv (\psi \wedge \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ and $\mathfrak{M}, w \Vdash \chi$.
6. $\varphi \equiv (\psi \vee \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ or $\mathfrak{M}, w \Vdash \chi$ (or both).
7. $\varphi \equiv (\psi \rightarrow \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \not\Vdash \psi$ or $\mathfrak{M}, w \Vdash \chi$.
8. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff either both $\mathfrak{M}, w \Vdash \psi$ and $\mathfrak{M}, w \Vdash \chi$ or neither $\mathfrak{M}, w \Vdash \psi$ nor $\mathfrak{M}, w \Vdash \chi$.
9. $\varphi \equiv \Box\psi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w' \Vdash \psi$ for all $w' \in W$ with Rww' .
10. $\varphi \equiv \Diamond\psi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w' \Vdash \psi$ for at least one $w' \in W$ with Rww' .

nml:syn:trw:
defn:sub:mmodels-box
nml:syn:trw:
defn:sub:mmodels-diamond

Note that by clause (9), a **fórmula** $\Box\psi$ is true at w whenever there are no w' with Rww' . In such a case $\Box\psi$ is *vacuously* true at w . Also, $\Box\psi$ may be satisfied at w even if ψ is not. The truth of ψ at w does not guarantee the truth of $\Diamond\psi$ at w . This holds, however, if Rww , p.ex., if R is reflexive. If there is no w' such that Rww' , then $\mathfrak{M}, w \not\Vdash \Diamond\varphi$, for any φ .

Problema 49.1. Consider the model of [Figura 49.1](#). Which of the following hold?

1. $\mathfrak{M}, w_1 \Vdash q$;
2. $\mathfrak{M}, w_3 \Vdash \neg q$;
3. $\mathfrak{M}, w_1 \Vdash p \vee q$;
4. $\mathfrak{M}, w_1 \Vdash \Box(p \vee q)$;
5. $\mathfrak{M}, w_3 \Vdash \Box q$;
6. $\mathfrak{M}, w_3 \Vdash \Box \perp$;
7. $\mathfrak{M}, w_1 \Vdash \Diamond q$;
8. $\mathfrak{M}, w_1 \Vdash \Box q$;
9. $\mathfrak{M}, w_1 \Vdash \neg \Box \Box \neg q$.

Proposição 49.7.

nml:syn:trw:
prop:dual

1. $\mathfrak{M}, w \Vdash \Box \varphi$ iff $\mathfrak{M}, w \Vdash \neg \Diamond \neg \varphi$.
2. $\mathfrak{M}, w \Vdash \Diamond \varphi$ iff $\mathfrak{M}, w \Vdash \neg \Box \neg \varphi$.

Prova. 1. $\mathfrak{M}, w \Vdash \neg \Diamond \neg \varphi$ iff $\mathfrak{M}, w \not\Vdash \Diamond \neg \varphi$ by definition of $\mathfrak{M}, w \Vdash$. $\mathfrak{M}, w \not\Vdash \Diamond \neg \varphi$ iff for some w' with Rww' , $\mathfrak{M}, w' \Vdash \neg \varphi$. Hence, $\mathfrak{M}, w \not\Vdash \Diamond \neg \varphi$ iff for all w' with Rww' , $\mathfrak{M}, w' \not\Vdash \neg \varphi$. We also have $\mathfrak{M}, w' \not\Vdash \neg \varphi$ iff $\mathfrak{M}, w' \Vdash \varphi$. Together we have $\mathfrak{M}, w \Vdash \neg \Diamond \neg \varphi$ iff for all w' with Rww' , $\mathfrak{M}, w' \Vdash \varphi$. Again by definition of $\mathfrak{M}, w \Vdash$, that is the case iff $\mathfrak{M}, w \Vdash \Box \varphi$.

2. $\mathfrak{M}, w \Vdash \neg \Box \neg \varphi$ iff $\mathfrak{M} \not\Vdash \Box \neg \varphi$. $\mathfrak{M}, w \Vdash \Box \neg \varphi$ iff for all w' with Rww' , $\mathfrak{M}, w' \Vdash \neg \varphi$. Hence, $\mathfrak{M}, w \not\Vdash \Box \neg \varphi$ iff for some w' with Rww' , $\mathfrak{M}, w' \not\Vdash \neg \varphi$. We also have $\mathfrak{M}, w' \not\Vdash \neg \varphi$ iff $\mathfrak{M}, w' \Vdash \varphi$. Together we have $\mathfrak{M}, w \Vdash \neg \Box \neg \varphi$ iff for some w' with Rww' , $\mathfrak{M}, w' \Vdash \varphi$. Again by definition of $\mathfrak{M}, w \Vdash$, that is the case iff $\mathfrak{M}, w \Vdash \Diamond \varphi$. $\square$

Problema 49.2. Complete the proof of [Proposição 49.7](#).

Problema 49.3. Let $\mathfrak{M} = \langle W, R, V \rangle$ be a model, and suppose $w_1, w_2 \in W$ are such that:

1. $w_1 \in V(p)$ if and only if $w_2 \in V(p)$ (for every [variável proposicional](#) p); and
2. for all $w \in W$: Rw_1w if and only if Rw_2w .

Using induction on [fórmulas](#), show that for all [fórmulas](#) φ : $\mathfrak{M}, w_1 \Vdash \varphi$ if and only if $\mathfrak{M}, w_2 \Vdash \varphi$.

49.6. TRUTH IN A MODEL

Problema 49.4. Let $\mathfrak{M} = \langle W, R, V \rangle$. Show that $\mathfrak{M}, w \Vdash \neg \Diamond \varphi$ if and only if $\mathfrak{M}, w \Vdash \Box \neg \varphi$.

-NoValue-

`content/normal-modal-logic/syntax-and-semantics/truth-in-model.tex`

49.6 Truth in a Model

nml:syn:tru: sec Sometimes we are interested in which **fórmulas** are true at every world in a given model. Let's introduce a notation for this.

Definição 49.8. A **fórmula** φ is *true in a model* $M = \langle W, R, V \rangle$, written $\mathfrak{M} \Vdash \varphi$, if and only if $\mathfrak{M}, w \Vdash \varphi$ for every $w \in W$.

nml:syn:tru: prop:truthfacts **Proposição 49.9.**

1. If $\mathfrak{M} \Vdash \varphi$ then $\mathfrak{M} \nVdash \neg \varphi$, but not vice-versa.
2. If $\mathfrak{M} \Vdash \varphi \rightarrow \psi$ then $\mathfrak{M} \Vdash \varphi$ only if $\mathfrak{M} \Vdash \psi$, but not vice-versa.

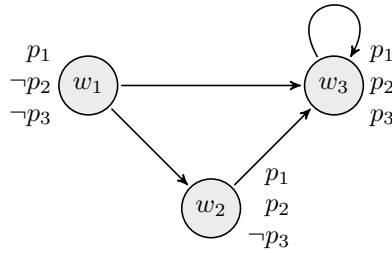
Prova. 1. If $\mathfrak{M} \Vdash \varphi$ then φ is true at all worlds in W , and since $W \neq \emptyset$, it can't be that $\mathfrak{M} \Vdash \neg \varphi$, or else φ would have to be both true and false at some world.

On the other hand, if $\mathfrak{M} \nVdash \neg \varphi$ then φ is true at some world $w \in W$. It does not follow that $\mathfrak{M}, w \Vdash \varphi$ for *every* $w \in W$. For instance, in the model of **Figura 49.1**, $\mathfrak{M} \nVdash \neg p$, and also $\mathfrak{M} \nVdash p$.

2. Assume $\mathfrak{M} \Vdash \varphi \rightarrow \psi$ and $\mathfrak{M} \Vdash \varphi$; to show $\mathfrak{M} \Vdash \psi$ let $w \in W$ be an arbitrary world. Then $\mathfrak{M}, w \Vdash \varphi \rightarrow \psi$ and $\mathfrak{M}, w \Vdash \varphi$, so $\mathfrak{M}, w \Vdash \psi$, and since w was arbitrary, $\mathfrak{M} \Vdash \psi$.

To show that the converse fails, we need to find a model $\mathfrak{M}$ such that $\mathfrak{M} \Vdash \varphi$ only if $\mathfrak{M} \Vdash \psi$, but $\mathfrak{M} \nVdash \varphi \rightarrow \psi$. Consider again the model of **Figura 49.1**: $\mathfrak{M} \nVdash p$ and hence (vacuously) $\mathfrak{M} \Vdash p$ only if $\mathfrak{M} \Vdash q$. However, $\mathfrak{M} \nVdash p \rightarrow q$, as p is true but q false at w_1 . $\square$

Problema 49.5. Consider the following model $\mathfrak{M}$ for the language comprising p_1, p_2, p_3 as the only **variáveis proposicionais**:



Are the following **fórmulas** and schemas true in the model $\mathfrak{M}$, O.s., true at every world in $\mathfrak{M}$? Explain.

1. $p \rightarrow \Diamond p$ (for p atomic);
2. $\varphi \rightarrow \Diamond \varphi$ (for φ arbitrary);
3. $\Box p \rightarrow p$ (for p atomic);
4. $\neg p \rightarrow \Diamond \Box p$ (for p atomic);
5. $\Diamond \Box \varphi$ (for φ arbitrary);
6. $\Box \Diamond p$ (for p atomic).

-NoValue-

content/normal-modal-logic/syntax-and-semantics/modal-validity.tex

49.7 Validity

explanation **Fórmulas** that are true in all models, O.s., true at every world in every model, are particularly interesting. They represent those modal propositions which are true regardless of how $\Box$ and $\Diamond$ are interpreted, as long as the interpretation is “normal” in the sense that it is generated by some accessibility relation on possible worlds. We call such **fórmulas** *valid*. For instance, $\Box(p \wedge q) \rightarrow \Box p$ is valid. Some **fórmulas** one might expect to be valid on the basis of the alethic interpretation of $\Box$, such as $\Box p \rightarrow p$, are not valid, however. Part of the interest of relational models is that different interpretations of $\Box$ and $\Diamond$ can be captured by different kinds of accessibility relations. This suggests that we should define validity not just relative to *all* models, but relative to all models *of a certain kind*. It will turn out, p.ex., that $\Box p \rightarrow p$ is true in all models where every world is accessible from itself, O.s., R is reflexive. Defining validity relative to classes of models enables us to formulate this succinctly: $\Box p \rightarrow p$ is valid in the class of reflexive models.

nml:syn:val:
sec

Definição 49.10. A **fórmula** φ is *valid* in a class $\mathcal{C}$ of models if it is true in every model in $\mathcal{C}$ (O.s., true at every world in every model in $\mathcal{C}$). If φ is valid in $\mathcal{C}$, we write $\mathcal{C} \models \varphi$, and we write $\models \varphi$ if φ is valid in the class of *all* models.

Proposição 49.11. If φ is valid in $\mathcal{C}$ it is also valid in each class $\mathcal{C}' \subseteq \mathcal{C}$.

nml:syn:val:
prop:subset-class

Proposição 49.12. If φ is valid, then so is $\Box \varphi$.

nml:syn:val:
prop:Nec-rule

Prova. Assume $\models \varphi$. To show $\models \Box \varphi$ let $\mathfrak{M} = \langle W, R, V \rangle$ be a model and $w \in W$. If Rww' then $\mathfrak{M}, w' \models \varphi$, since φ is valid, and so also $\mathfrak{M}, w \models \Box \varphi$. Since $\mathfrak{M}$ and w were arbitrary, $\models \Box \varphi$. $\square$

Problema 49.6. Show that the following are valid:

49.8. TAUTOLOGICAL INSTANCES

1. $\Box p \rightarrow \Box(q \rightarrow p)$;
2. $\Box \neg \perp$;
3. $\Box p \rightarrow (\Box q \rightarrow \Box p)$.

Problema 49.7. Show that $\varphi \rightarrow \Box \varphi$ is valid in the class $\mathcal{C}$ of models $\mathfrak{M} = \langle W, R, V \rangle$ where $W = \{w\}$. Similarly, show that $\psi \rightarrow \Box \varphi$ and $\Diamond \varphi \rightarrow \psi$ are valid in the class of models $\mathfrak{M} = \langle W, R, V \rangle$ where $R = \emptyset$.

-NoValue-

content/normal-modal-logic/syntax-and-semantics/tautological-instances.tex

49.8 Tautological Instances

nml:syn:tau:
sec

A modal-free formula is a tautology if it is true under every truth-value assignment. Clearly, every tautology is true at every world in every model. But for **fórmulas** involving $\Box$ and $\Diamond$, the notion of tautology is not defined. Is it the case, p.ex., that $\Box p \vee \neg \Box p$ —an instance of the principle of excluded middle—is valid? The notion of a *tautological instance* helps: a **fórmula** that is a substitution instance of a (non-modal) tautology. It is not surprising, but still requires proof, that every tautological instance is valid.

explanation

Definição 49.13. A modal **fórmula** ψ is a *tautological instance* if and only if there is a modal-free tautology φ with **variáveis proposicionais** $p_1, \dots, p_n$ and **fórmulas** $\theta_1, \dots, \theta_n$ such that $\psi \equiv \varphi[\theta_1/p_1, \dots, \theta_n/p_n]$.

nml:syn:tau:
lem:valid-taut

Lema 49.14. Suppose φ is a modal-free **fórmula** whose **variáveis proposicionais** are $p_1, \dots, p_n$, and let $\theta_1, \dots, \theta_n$ be modal **fórmulas**. Then for any assignment $\mathbf{v}$, any model $\mathfrak{M} = \langle W, R, V \rangle$, and any $w \in W$ such that $\mathbf{v}(p_i) = \mathbb{T}$ if and only if $\mathfrak{M}, w \Vdash \theta_i$ we have that $\mathbf{v} \models \varphi$ if and only if $\mathfrak{M}, w \Vdash \varphi[\theta_1/p_1, \dots, \theta_n/p_n]$.

Prova. By induction on φ .

1. $\varphi \equiv \perp$: Both $\mathbf{v} \not\models \perp$ and $\mathfrak{M}, w \not\Vdash \perp$.
2. $\varphi \equiv \top$: Both $\mathbf{v} \models \top$ and $\mathfrak{M}, w \Vdash \top$.
3. $\varphi \equiv p_i$:

$$\begin{aligned}
 \mathbf{v} \models p_i &\Leftrightarrow \mathbf{v}(p_i) = \mathbb{T} \\
 &\quad \text{by definition of } \mathbf{v} \models p_i \\
 &\Leftrightarrow \mathfrak{M}, w \Vdash \theta_i \\
 &\quad \text{by assumption} \\
 &\Leftrightarrow \mathfrak{M}, w \Vdash p_i[\theta_1/p_1, \dots, \theta_n/p_n] \\
 &\quad \text{since } p_i[\theta_1/p_1, \dots, \theta_n/p_n] \equiv \theta_i.
 \end{aligned}$$

4. $\varphi \equiv \neg\psi$:

$$\begin{aligned}
 \mathfrak{v} \models \neg\psi &\Leftrightarrow \mathfrak{v} \not\models \psi \\
 &\text{by definition of } \mathfrak{v} \models; \\
 &\Leftrightarrow \mathfrak{M}, w \not\models \psi[\theta_1/p_1, \dots, \theta_n/p_n] \\
 &\text{by induction hypothesis} \\
 &\Leftrightarrow \mathfrak{M}, w \models \neg\psi[\theta_1/p_1, \dots, \theta_n/p_n] \\
 &\text{by definition of } \mathfrak{v} \models.
 \end{aligned}$$

5. $\varphi \equiv (\psi \wedge \chi)$:

$$\begin{aligned}
 \mathfrak{v} \models \psi \wedge \chi &\Leftrightarrow \mathfrak{v} \models \psi \text{ and } \mathfrak{v} \models \chi \\
 &\text{by definition of } \mathfrak{v} \models \\
 &\Leftrightarrow \mathfrak{M}, w \models \psi[\theta_1/p_1, \dots, \theta_n/p_n] \text{ and } \\
 &\quad \mathfrak{M}, w \models \chi[\theta_1/p_1, \dots, \theta_n/p_n] \\
 &\text{by induction hypothesis} \\
 &\Leftrightarrow \mathfrak{M}, w \models (\psi \wedge \chi)[\theta_1/p_1, \dots, \theta_n/p_n] \\
 &\text{by definition of } \mathfrak{M}, w \models.
 \end{aligned}$$

6. $\varphi \equiv (\psi \vee \chi)$:

$$\begin{aligned}
 \mathfrak{v} \models \psi \vee \chi &\Leftrightarrow \mathfrak{v} \models \psi \text{ or } \mathfrak{v} \models \chi \\
 &\text{by definition of } \mathfrak{v} \models; \\
 &\Leftrightarrow \mathfrak{M}, w \models \psi[\theta_1/p_1, \dots, \theta_n/p_n] \text{ or } \\
 &\quad \mathfrak{M}, w \models \chi[\theta_1/p_1, \dots, \theta_n/p_n] \\
 &\text{by induction hypothesis} \\
 &\Leftrightarrow \mathfrak{M}, w \models (\psi \vee \chi)[\theta_1/p_1, \dots, \theta_n/p_n] \\
 &\text{by definition of } \mathfrak{M}, w \models.
 \end{aligned}$$

7. $\varphi \equiv (\psi \rightarrow \chi)$:

$$\begin{aligned}
 \mathfrak{v} \models \psi \rightarrow \chi &\Leftrightarrow \mathfrak{v} \not\models \psi \text{ or } \mathfrak{v} \models \chi \\
 &\text{by definition of } \mathfrak{v} \models \\
 &\Leftrightarrow \mathfrak{M}, w \not\models \psi[\theta_1/p_1, \dots, \theta_n/p_n] \text{ or } \\
 &\quad \mathfrak{M}, w \models \chi[\theta_1/p_1, \dots, \theta_n/p_n] \\
 &\text{by induction hypothesis} \\
 &\Leftrightarrow \mathfrak{M}, w \models (\psi \rightarrow \chi)[\theta_1/p_1, \dots, \theta_n/p_n] \\
 &\text{by definition of } \mathfrak{M}, w \models.
 \end{aligned}$$

49.9. SCHEMAS AND VALIDITY

8. $\varphi \equiv (\psi \leftrightarrow \chi)$:

$$\begin{aligned}
 \mathfrak{v} \models \psi \rightarrow \chi &\Leftrightarrow \text{either } \mathfrak{v} \models \psi \text{ and } \mathfrak{v} \models \chi \\
 &\quad \text{or } \mathfrak{v} \not\models \psi \text{ and } \mathfrak{v} \not\models \chi \\
 &\quad \text{by definition of } \mathfrak{v} \models \\
 &\Leftrightarrow \text{either } \mathfrak{M}, w \Vdash \psi[\theta_1/p_1, \dots, \theta_n/p_n] \text{ and} \\
 &\quad \mathfrak{M}, w \Vdash \chi[\theta_1/p_1, \dots, \theta_n/p_n] \\
 &\quad \text{or } \mathfrak{M}, w \nVdash \psi[\theta_1/p_1, \dots, \theta_n/p_n] \text{ and} \\
 &\quad \mathfrak{M}, w \nVdash \chi[\theta_1/p_1, \dots, \theta_n/p_n] \\
 &\quad \text{by induction hypothesis} \\
 &\Leftrightarrow \mathfrak{M}, w \Vdash (\psi \leftrightarrow \chi)[\theta_1/p_1, \dots, \theta_n/p_n] \\
 &\quad \text{by definition of } \mathfrak{M}, w \Vdash. \quad \square
 \end{aligned}$$

*nml:syn:tau:
prop:valid-taut*

Proposição 49.15. *All tautological instances are valid.*

Prova. Contrapositively, suppose φ is such that $\mathfrak{M}, w \nVdash \varphi[\theta_1/p_1, \dots, \theta_n/p_n]$, for some model $\mathfrak{M}$ and world w . Define an assignment $\mathfrak{v}$ such that $\mathfrak{v}(p_i) = \mathbb{T}$ if and only if $\mathfrak{M}, w \Vdash \theta_i$ (and $\mathfrak{v}$ assigns arbitrary values to $q \notin \{p_1, \dots, p_n\}$). Then by [Lema 49.14](#), $\mathfrak{v} \not\models \varphi$, so φ is not a tautology. $\square$

-NoValue-

[content/normal-modal-logic/syntax-and-semantics/schemas.tex](#)

49.9 Schemas and Validity

*nml:syn:sch:
sec*

Definição 49.16. A *schema* is a set of *fórmulas* comprising all and only the substitution instances of some modal *fórmula* χ , O.s.,

$$\{\psi : \exists \theta_1, \dots, \exists \theta_n (\psi = \chi[\theta_1/p_1, \dots, \theta_n/p_n])\}.$$

The *fórmula* χ is called the *characteristic fórmula* of the schema, and it is unique up to a renaming of the *variáveis proposicionais*. A *fórmula* φ is an *instance* of a schema if it is a member of the set.

It is convenient to denote a schema by the meta-linguistic expression obtained by substituting ‘ φ ’, ‘ ψ ’, ..., for the atomic components of χ . So, for instance, the following denote schemas: ‘ φ ’, ‘ $\varphi \rightarrow \Box \varphi$ ’, ‘ $\varphi \rightarrow (\psi \rightarrow \varphi)$ ’. They correspond to the characteristic *fórmulas* p , $p \rightarrow \Box p$, $p \rightarrow (q \rightarrow p)$. The schema ‘ φ ’ denotes the set of *all fórmulas*.

Definição 49.17. A schema is *true* in a model if and only if all of its instances are; and a schema is *valid* if and only if it is true in every model.

Proposição 49.18. *The following schema K is valid*

*nml:syn:sch:
prop:Kvalid*

$$\Box(\varphi \rightarrow \psi) \rightarrow (\Box\varphi \rightarrow \Box\psi). \quad (\text{K})$$

Prova. We need to show that all instances of the schema are true at every world in every model. So let $\mathfrak{M} = \langle W, R, V \rangle$ and $w \in W$ be arbitrary. To show that a conditional is true at a world we assume the antecedent is true to show that the consequent is true as well. In this case, let $\mathfrak{M}, w \Vdash \Box(\varphi \rightarrow \psi)$ and $\mathfrak{M}, w \Vdash \Box\varphi$. We need to show $\mathfrak{M}, w \Vdash \Box\psi$. So let w' be arbitrary such that Rww' . Then by the first assumption $\mathfrak{M}, w' \Vdash \varphi \rightarrow \psi$ and by the second assumption $\mathfrak{M}, w' \Vdash \varphi$. It follows that $\mathfrak{M}, w' \Vdash \psi$. Since w' was arbitrary, $\mathfrak{M}, w \Vdash \Box\psi$. $\square$

Proposição 49.19. *The following schema DUAL is valid*

*nml:syn:sch:
prop:Dual-valid*

$$\Diamond\varphi \leftrightarrow \neg\Box\neg\varphi. \quad (\text{DUAL})$$

Prova. Exercise. $\square$

Problema 49.8. Prove **Proposição 49.19**.

Proposição 49.20. *If φ and $\varphi \rightarrow \psi$ are true at a world in a model then so is ψ . Hence, the valid **fórmulas** are closed under modus ponens.*

*nml:syn:sch:
prop:soundMP*

Proposição 49.21. *A **fórmula** φ is valid iff all its substitution instances are. In other words, a schema is valid iff its characteristic **fórmula** is.*

*nml:syn:sch:
prop:valid-instances*

Prova. The “if” direction is obvious, since φ is a substitution instance of itself.

To prove the “only if” direction, we show the following: Suppose $\mathfrak{M} = \langle W, R, V \rangle$ is a modal model, and $\psi \equiv \varphi[\theta_1/p_1, \dots, \theta_n/p_n]$ is a substitution instance of φ . Define $\mathfrak{M}' = \langle W, R, V' \rangle$ by $V'(p_i) = \{w : \mathfrak{M}, w \Vdash \theta_i\}$. Then $\mathfrak{M}, w \Vdash \psi$ iff $\mathfrak{M}', w \Vdash \varphi$, for any $w \in W$. (We leave the proof as an exercise.) Now suppose that φ was valid, but some substitution instance ψ of φ was not valid. Then for some $\mathfrak{M} = \langle W, R, V \rangle$ and some $w \in W$, $\mathfrak{M}, w \not\Vdash \psi$. But then $\mathfrak{M}', w \not\Vdash \varphi$ by the claim, and φ is not valid, a contradiction. $\square$

Problema 49.9. Prove the claim in the “only if” part of the proof of **Proposição 49.21**. (Hint: use induction on φ .)

Note, however, that it is not true that a schema is true in a model iff its characteristic formula is. Of course, the “only if” direction holds: if every instance of φ is true in $\mathfrak{M}$, φ itself is true in $\mathfrak{M}$. But it may happen that φ is true in $\mathfrak{M}$ but some instance of φ is false at some world in $\mathfrak{M}$. For a very simple counterexample consider p in a model with only one world w and $V(p) = \{w\}$, so that p is true at w . But $\perp$ is an instance of p , and not true at w .

Problema 49.10. Show that none of the following **fórmulas** are valid:

49.10. ENTAILMENT

Valid Schemas	Invalid Schemas
$\Box(\varphi \rightarrow \psi) \rightarrow (\Diamond\varphi \rightarrow \Diamond\psi)$	$\Box(\varphi \vee \psi) \rightarrow (\Box\varphi \vee \Box\psi)$
$\Diamond(\varphi \rightarrow \psi) \rightarrow (\Box\varphi \rightarrow \Diamond\psi)$	$(\Diamond\varphi \wedge \Diamond\psi) \rightarrow \Diamond(\varphi \wedge \psi)$
$\Box(\varphi \wedge \psi) \leftrightarrow (\Box\varphi \wedge \Box\psi)$	$\varphi \rightarrow \Box\varphi$
$\Box\varphi \rightarrow \Box(\psi \rightarrow \varphi)$	$\Box\Diamond\varphi \rightarrow \psi$
$\neg\Diamond\varphi \rightarrow \Box(\varphi \rightarrow \psi)$	$\Box\Box\varphi \rightarrow \Box\varphi$
$\Diamond(\varphi \vee \psi) \leftrightarrow (\Diamond\varphi \vee \Diamond\psi)$	$\Box\Diamond\varphi \rightarrow \Diamond\Box\varphi.$

Tabela 49.1: Valid and (or?) invalid schemas.

nml:syn:sch:
tab:valid-invalidSchemas

- D: $\Box p \rightarrow \Diamond p$;
T: $\Box p \rightarrow p$;
B: $p \rightarrow \Box\Diamond p$;
4: $\Box p \rightarrow \Box\Box p$;
5: $\Diamond p \rightarrow \Box\Diamond p$.

Problema 49.11. Prove that the schemas in the first column of [Tabela 49.1](#) are valid and those in the second column are not valid.

Problema 49.12. Decide whether the following schemas are valid or invalid:

- $(\Diamond\varphi \rightarrow \Box\psi) \rightarrow (\Box\varphi \rightarrow \Box\psi)$;
- $\Diamond(\varphi \rightarrow \psi) \vee \Box(\psi \rightarrow \varphi)$.

Problema 49.13. For each of the following schemas find a model $\mathfrak{M}$ such that every instance of the [fórmula](#) is true in $\mathfrak{M}$:

- $p \rightarrow \Diamond\Diamond p$;
- $\Diamond p \rightarrow \Box p$.

-NoValue-

[content/normal-modal-logic/syntax-and-semantics/entailment.tex](#)

49.10 Entailment

nml:syn:ent:
sec

With the definition of truth at a world, we can define an entailment relation [explanation](#) between [fórmulas](#). A [fórmula](#) ψ entails φ iff, whenever ψ is true, φ is true as well. Here, “whenever” means both “whichever model we consider” as well as “whichever world in that model we consider.”

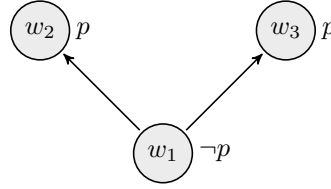


Figura 49.2: Counterexample to $p \rightarrow \Diamond p \models \Box p \rightarrow p$.

nml:syn:ent:
fig:counterex

Definição 49.22. If Γ is a set of **fórmulas** and φ a **fórmula**, then Γ entails φ , in symbols: $\Gamma \models \varphi$, if and only if for every model $\mathfrak{M} = \langle W, R, V \rangle$ and world $w \in W$, if $\mathfrak{M}, w \Vdash \psi$ for every $\psi \in \Gamma$, then $\mathfrak{M}, w \Vdash \varphi$. If Γ contains a single **fórmula** ψ , then we write $\psi \models \varphi$.

Exemplo 49.23. To show that a **fórmula** entails another, we have to reason about all models, using the definition of $\mathfrak{M}, w \Vdash$. For instance, to show $p \rightarrow \Diamond p \models \Box \neg p \rightarrow \neg p$, we might argue as follows: Consider a model $\mathfrak{M} = \langle W, R, V \rangle$ and $w \in W$, and suppose $\mathfrak{M}, w \Vdash p \rightarrow \Diamond p$. We have to show that $\mathfrak{M}, w \Vdash \Box \neg p \rightarrow \neg p$. Suppose not. Then $\mathfrak{M}, w \Vdash \Box \neg p$ and $\mathfrak{M}, w \nVdash \neg p$. Since $\mathfrak{M}, w \nVdash \neg p$, $\mathfrak{M}, w \Vdash p$. By assumption, $\mathfrak{M}, w \Vdash p \rightarrow \Diamond p$, hence $\mathfrak{M}, w \Vdash \Diamond p$. By definition of $\mathfrak{M}, w \Vdash \Diamond p$, there is some w' with Rww' such that $\mathfrak{M}, w' \Vdash p$. Since also $\mathfrak{M}, w \Vdash \Box \neg p$, $\mathfrak{M}, w' \Vdash \neg p$, a contradiction.

To show that a **fórmula** ψ does not entail another φ , we have to give a counterexample, O.s., a model $\mathfrak{M} = \langle W, R, V \rangle$ where we show that at some world $w \in W$, $\mathfrak{M}, w \Vdash \psi$ but $\mathfrak{M}, w \nVdash \varphi$. Let's show that $p \rightarrow \Diamond p \nmodels \Box p \rightarrow p$. Consider the model in **Figura 49.2**. We have $\mathfrak{M}, w_1 \Vdash \Diamond p$ and hence $\mathfrak{M}, w_1 \Vdash p \rightarrow \Diamond p$. However, since $\mathfrak{M}, w_1 \Vdash \Box p$ but $\mathfrak{M}, w_1 \nVdash p$, we have $\mathfrak{M}, w_1 \nVdash \Box p \rightarrow p$.

Often very simple counterexamples suffice. The model $\mathfrak{M}' = \{W', R', V'\}$ with $W' = \{w\}$, $R' = \emptyset$, and $V'(p) = \emptyset$ is also a counterexample: Since $\mathfrak{M}', w \nVdash p$, $\mathfrak{M}', w \Vdash p \rightarrow \Diamond p$. As no worlds are accessible from w , we have $\mathfrak{M}', w \Vdash \Box p$, and so $\mathfrak{M}', w \nVdash \Box p \rightarrow p$.

Problema 49.14. Show that $\Box(\varphi \wedge \psi) \models \Box\varphi$.

Problema 49.15. Show that $\Box(p \rightarrow q) \nmodels p \rightarrow \Box q$ and $p \rightarrow \Box q \nmodels \Box(p \rightarrow q)$.

Capítulo 50

Frame Definability

-NoValue-

`content/normal-modal-logic/frame-definability/introduction.tex`

50.1 Introduction

nml:fd:int:
sec

One question that interests modal logicians is the relationship between the accessibility relation and the truth of certain *fórmulas* in models with that accessibility relation. For instance, suppose the accessibility relation is reflexive, O.s., for every $w \in W$, Rww . In other words, every world is accessible from itself. That means that when $\Box\varphi$ is true at a world w , w itself is among the accessible worlds at which φ must therefore be true. So, if the accessibility relation R of $\mathfrak{M}$ is reflexive, then whatever world w and formula φ we take, $\Box\varphi \rightarrow \varphi$ will be true there (in other words, the schema $\Box p \rightarrow p$ and all its substitution instances are true in $\mathfrak{M}$).

The converse, however, is false. It's not the case, p.ex., that if $\Box p \rightarrow p$ is true in $\mathfrak{M}$, then R is reflexive. For we can easily find a non-reflexive model $\mathfrak{M}$ where $\Box p \rightarrow p$ is true at all worlds: take the model with a single world w , not accessible from itself, but with $w \in V(p)$. By picking the truth value of p suitably, we can make $\Box\varphi \rightarrow \varphi$ true in a model that is not reflexive.

The solution is to remove the variable assignment V from the equation. If we require that $\Box p \rightarrow p$ is true at all worlds in $\mathfrak{M}$, regardless of which worlds are in $V(p)$, then it is necessary that R is reflexive. For in any non-reflexive model, there will be at least one world w such that not Rww . If we set $V(p) = W \setminus \{w\}$, then p will be true at all worlds other than w , and so at all worlds accessible from w (since w is guaranteed not to be accessible from w , and w is the only world where p is false). On the other hand, p is false at w , so $\Box p \rightarrow p$ is false at w .

This suggests that we should introduce a notation for model structures without a valuation: we call these *frames*. A frame $\mathfrak{F}$ is simply a pair $\langle W, R \rangle$ consisting of a set of worlds with an accessibility relation. Every model $\langle W, R, V \rangle$

If R is ...	then ... is true in $\mathfrak{M}$:
<i>serial</i> : $\forall u \exists v Ruv$	$\Box p \rightarrow \Diamond p$ (D)
<i>reflexive</i> : $\forall w Rww$	$\Box p \rightarrow p$ (T)
<i>symmetric</i> : $\forall u \forall v (Ruv \rightarrow Rvu)$	$p \rightarrow \Box \Diamond p$ (B)
<i>transitive</i> : $\forall u \forall v \forall w ((Ruv \wedge Rvw) \rightarrow Ruw)$	$\Box p \rightarrow \Box \Box p$ (4)
<i>euclidean</i> : $\forall w \forall u \forall v ((Rwu \wedge Rvw) \rightarrow Ruw)$	$\Diamond p \rightarrow \Box \Diamond p$ (5)

Tabela 50.1: Five correspondence facts.

is then, as we say, *based on* the frame $\langle W, R \rangle$. Conversely, a frame determines the class of models based on it; and a class of frames determines the class of models which are based on any frame in the class. And we can define $\mathfrak{F} \models \varphi$, the notion of a **fórmula** being *valid* in a frame as: $\mathfrak{M} \Vdash \varphi$ for all $\mathfrak{M}$ based on $\mathfrak{F}$.

With this notation, we can establish correspondence relations between **fórmulas** and classes of frames: p.ex., $\mathfrak{F} \models \Box p \rightarrow p$ if, and only if, $\mathfrak{F}$ is reflexive.

-NoValue-

nml:frd:acc:
tab:five

content/normal-modal-logic/frame-definability/properties-accessibility.tex

50.2 Properties of Accessibility Relations

Many modal **fórmulas** turn out to be characteristic of simple, and even familiar, properties of the accessibility relation. In one direction, that means that any model that has a given property makes a corresponding **fórmula** (and all its substitution instances) true. We begin with five classical examples of kinds of accessibility relations and the formulas the truth of which they guarantee.

nml:frd:acc:
sec

Teorema 50.1. *Let $\mathfrak{M} = \langle W, R, V \rangle$ be a model. If R has the property on the left side of **Tabela 50.1**, every instance of the **fórmula** on the right side is true in $\mathfrak{M}$.*

nml:frd:acc:
thm:soundschemas

Prova. Here is the case for B: to show that the schema is true in a model we need to show that all of its instances are true at all worlds in the model. So let $\varphi \rightarrow \Box \Diamond \varphi$ be a given instance of B, and let $w \in W$ be an arbitrary world. Suppose the antecedent φ is true at w , in order to show that $\Box \Diamond \varphi$ is true at w . So we need to show that $\Diamond \varphi$ is true at all w' accessible from w . Now, for any w' such that Rww' we have, using the hypothesis of symmetry, that also $Rw'w$ (see **Figura 50.1**). Since $\mathfrak{M}, w \Vdash \varphi$, we have $\mathfrak{M}, w' \Vdash \Diamond \varphi$. Since w' was an arbitrary world such that Rww' , we have $\mathfrak{M}, w \Vdash \Box \Diamond \varphi$.

We leave the other cases as exercises. □

Problema 50.1. Complete the proof of **Teorema 50.1**.

50.2. PROPERTIES OF ACCESSIBILITY RELATIONS

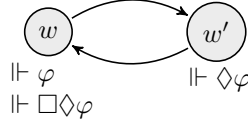


Figura 50.1: The argument from symmetry.

nml:frd:acc:
fig:Bsymm

Notice that the converse implications of **Teorema 50.1** do not hold: it's not true that if a model verifies a schema, then the accessibility relation of that model has the corresponding property. In the case of T and reflexive models, it is easy to give an example of a model in which T itself fails: let $W = \{w\}$ and $V(p) = \emptyset$. Then R is not reflexive, but $\mathfrak{M}, w \Vdash \Box p$ and $\mathfrak{M}, w \not\Vdash p$. But here we have just a single instance of T that fails in $\mathfrak{M}$; other instances, p.ex., $\Box \neg p \rightarrow \neg p$ are true. It is harder to give examples where *every substitution instance* of T is true in $\mathfrak{M}$ and $\mathfrak{M}$ is not reflexive. But there are such models, too:

nml:frd:acc:
prop:reflexive

Proposição 50.2. *Let $\mathfrak{M} = \langle W, R, V \rangle$ be a model such that $W = \{u, v\}$, where worlds u and v are related by R : O.s., both Ruv and Rvu . Suppose that for all p : $u \in V(p) \Leftrightarrow v \in V(p)$. Then:*

1. *For all φ : $\mathfrak{M}, u \Vdash \varphi$ if and only if $\mathfrak{M}, v \Vdash \varphi$ (use induction on φ).*
2. *Every instance of T is true in $\mathfrak{M}$.*

Since $\mathfrak{M}$ is not reflexive (it is, in fact, irreflexive), the converse of **Teorema 50.1** fails in the case of T (similar arguments can be given for some—though not all—the other schemas mentioned in **Teorema 50.1**).

Problema 50.2. Prove the claims in **Proposição 50.2**.

Although we will focus on the five classical fórmulas D, T, B, 4, and 5, we record in **Tabela 50.2** a few more properties of accessibility relations. The accessibility relation R is partially functional, if from every world at most one world is accessible. If it is the case that from every world exactly one world is accessible, we call it functional. (Thus the functional relations are precisely those that are both serial and partially functional). They are called “functional” because the accessibility relation operates like a (partial) function. A relation is weakly dense if whenever Ruv , there is a w “between” u and v . So weakly dense relations are in a sense the opposite of transitive relations: in a transitive relation, whenever you can reach v from u by a detour via w , you can reach v from u directly; in a weakly dense relation, whenever you can reach v from u directly, you can also reach it by a detour via some w . A relation is weakly directed if whenever you can reach worlds u and v from some world w , you can reach a single world t from both u and v —this is sometimes called the “diamond property” or “confluence.”

If R is ...	then ... is true in $\mathfrak{M}$:
<i>partially functional</i> : $\forall w \forall u \forall v ((Rwu \wedge Ruv) \rightarrow u = v)$	$\Diamond p \rightarrow \Box p$
<i>functional</i> : $\forall w \exists v \forall u (Rwu \leftrightarrow u = v)$	$\Diamond p \leftrightarrow \Box p$
<i>weakly dense</i> : $\forall u \forall v (Ruv \rightarrow \exists w (Ruw \wedge Ruv))$	$\Box \Box p \rightarrow \Box p$
<i>weakly connected</i> : $\forall w \forall u \forall v ((Rwu \wedge Ruv) \rightarrow (Ruv \vee u = v \vee Rvu))$	$\Box((p \wedge \Box p) \rightarrow q) \vee \Box((q \wedge \Box q) \rightarrow p)$ (L)
<i>weakly directed</i> : $\forall w \forall u \forall v ((Rwu \wedge Ruv) \rightarrow \exists t (Rut \wedge Rvt))$	$\Diamond \Box p \rightarrow \Box \Diamond p$ (G)

Tabela 50.2: Five more correspondence facts.

Problema 50.3. Let $\mathfrak{M} = \langle W, R, V \rangle$ be a model. Show that if R satisfies the left-hand properties of [Tabela 50.2](#), every instance of the corresponding right-hand formula is true in $\mathfrak{M}$.

nml:frd:acc:
tab:anotherfive

-NoValue-

content/normal-modal-logic/frame-definability/frames.tex

50.3 Frames

Definição 50.3. A *frame* is a pair $\mathfrak{F} = \langle W, R \rangle$ where W is a non-empty set of worlds and R a binary relation on W . A model $\mathfrak{M}$ is *based on* a frame $\mathfrak{F} = \langle W, R \rangle$ if and only if $\mathfrak{M} = \langle W, R, V \rangle$ for some valuation V .

nml:frd:fra:
sec

Definição 50.4. If $\mathfrak{F}$ is a frame, we say that φ is *valid in* $\mathfrak{F}$, $\mathfrak{F} \models \varphi$, if $\mathfrak{M} \models \varphi$ for every model $\mathfrak{M}$ based on $\mathfrak{F}$.

If $\mathcal{F}$ is a class of frames, we say φ is *valid in* $\mathcal{F}$, $\mathcal{F} \models \varphi$, iff $\mathfrak{F} \models \varphi$ for every frame $\mathfrak{F} \in \mathcal{F}$.

The reason frames are interesting is that correspondence between schemas and properties of the accessibility relation R is at the level of frames, *not of models*. For instance, although T is true in all reflexive models, not every model in which T is true is reflexive. However, it *is* true that not only is T *valid* on all reflexive *frames*, also every frame in which T is valid is reflexive.

Observação 6. Validity in a class of frames is a special case of the notion of validity in a class of models: $\mathcal{F} \models \varphi$ iff $\mathcal{C} \models \varphi$ where $\mathcal{C}$ is the class of all models based on a frame in $\mathcal{F}$.

50.4. FRAME DEFINABILITY

Obviously, if a fórmula or a schema is valid, O.s., valid with respect to the class of *all* models, it is also valid with respect to any class $\mathcal{F}$ of frames.

-NoValue-

content/normal-modal-logic/frame-definability/definability.tex

50.4 Frame Definability

nml:frd:def:
sec

Even though the converse implications of [Teorema 50.1](#) fail, they hold if we replace “model” by “frame”: for the properties considered in [Teorema 50.1](#), it is true that if a fórmula is valid in a *frame* then the accessibility relation of that frame has the corresponding property. So, the fórmulas considered *define* the classes of frames that have the corresponding property.

Definição 50.5. If $\mathcal{F}$ is a class of frames, we say φ *defines* $\mathcal{F}$ iff $\mathfrak{F} \models \varphi$ for all and only frames $\mathfrak{F} \in \mathcal{F}$.

We now proceed to establish the full definability results for frames.

nml:frd:def:
thm:fullCorrespondence

Teorema 50.6. *If the fórmula on the right side of [Tabela 50.1](#) is valid in a frame $\mathfrak{F}$, then $\mathfrak{F}$ has the property on the left side.*

- Prova.*
1. Suppose D is valid in $\mathfrak{F} = \langle W, R \rangle$, O.s., $\mathfrak{F} \models \Box p \rightarrow \Diamond p$. Let $\mathfrak{M} = \langle W, R, V \rangle$ be a model based on $\mathfrak{F}$, and $w \in W$. We have to show that there is a v such that Rwv . Suppose not: then both $\mathfrak{M} \Vdash \Box \varphi$ and $\mathfrak{M}, w \nVdash \Diamond \varphi$ for any φ , including p . But then $\mathfrak{M}, w \nVdash \Box p \rightarrow \Diamond p$, contradicting the assumption that $\mathfrak{F} \models \Box p \rightarrow \Diamond p$.
 2. Suppose T is valid in $\mathfrak{F}$, O.s., $\mathfrak{F} \models \Box p \rightarrow p$. Let $w \in W$ be an arbitrary world; we need to show Rww . Let $u \in V(p)$ if and only if Rwu (when q is other than p , $V(q)$ is arbitrary, say $V(q) = \emptyset$). Let $\mathfrak{M} = \langle W, R, V \rangle$. By construction, for all u such that Rwu : $\mathfrak{M}, u \Vdash p$, and hence $\mathfrak{M}, w \Vdash \Box p$. But by hypothesis $\Box p \rightarrow p$ is true at w , so that $\mathfrak{M}, w \Vdash p$, but by definition of V this is possible only if Rww .
 3. We prove the contrapositive: Suppose $\mathfrak{F}$ is not symmetric, we show that B, O.s., $p \rightarrow \Box \Diamond p$ is not valid in $\mathfrak{F} = \langle W, R \rangle$. If $\mathfrak{F}$ is not symmetric, there are $u, v \in W$ such that Ruv but not Rvu . Define V such that $w \in V(p)$ if and only if not Rvw (and V is arbitrary otherwise). Let $\mathfrak{M} = \langle W, R, V \rangle$. Now, by definition of V , $\mathfrak{M}, w \Vdash p$ for all w such that not Rvw , in particular, $\mathfrak{M}, u \Vdash p$ since not Rvu . Also, since Rvw iff $w \notin V(p)$, there is no w such that Rvw and $\mathfrak{M}, w \Vdash p$, and hence $\mathfrak{M}, v \nVdash \Diamond p$. Since Ruv , also $\mathfrak{M}, u \nVdash \Box \Diamond p$. It follows that $\mathfrak{M}, u \nVdash p \rightarrow \Box \Diamond p$, and so B is not valid in $\mathfrak{F}$.

4. Suppose 4 is valid in $\mathfrak{F} = \langle W, R \rangle$, O.s., $\mathfrak{F} \models \Box p \rightarrow \Box \Box p$, and let $u, v, w \in W$ be arbitrary worlds such that Ruv and Rvw ; we need to show that Ruw . Define V such that $z \in V(p)$ if and only if Ruz (and V is arbitrary otherwise). Let $\mathfrak{M} = \langle W, R, V \rangle$. By definition of V , $\mathfrak{M}, z \Vdash p$ for all z such that Ruz , and hence $\mathfrak{M}, u \Vdash \Box p$. But by hypothesis 4, $\Box p \rightarrow \Box \Box p$, is true at u , so that $\mathfrak{M}, u \Vdash \Box \Box p$. Since Ruv and Rvw , we have $\mathfrak{M}, w \Vdash p$, but by definition of V this is possible only if Ruw , as desired.
5. We proceed contrapositively, assuming that the frame $\mathfrak{F} = \langle W, R \rangle$ is not euclidean, and show that it falsifies 5, O.s., $\mathfrak{F} \not\models \Diamond p \rightarrow \Box \Diamond p$. Suppose there are worlds $u, v, w \in W$ such that Rwu and Rvw but not Ruv . Define V such that for all worlds z , $z \in V(p)$ if and only if it is *not* the case that Ruz . Let $\mathfrak{M} = \langle W, R, V \rangle$. Then by hypothesis $\mathfrak{M}, v \Vdash p$ and since Rvw also $\mathfrak{M}, w \Vdash \Diamond p$. However, there is no world y such that Ruy and $\mathfrak{M}, y \Vdash p$ so $\mathfrak{M}, u \not\models \Diamond p$. Since Rwu , it follows that $\mathfrak{M}, w \not\models \Box \Diamond p$, so that 5, $\Diamond p \rightarrow \Box \Diamond p$, fails at w . $\square$

You'll notice a difference between the proof for D and the other cases: no mention was made of the valuation V . In effect, we proved that if $\mathfrak{M} \Vdash D$ then $\mathfrak{M}$ is serial. So D defines the class of serial *models*, not just frames.

Corolário 50.7. *Any model where D is true is serial.*

*nml:frd:def:
prop:D-serial*

Corolário 50.8. *Each **fórmula** on the right side of **Tabela 50.1** defines the class of frames which have the property on the left side.*

Prova. In **Teorema 50.1**, we proved that if a model has the property on the left, the **fórmula** on the right is true in it. Thus, if a frame $\mathfrak{F}$ has the property on the left, the **fórmula** on the right is valid in $\mathfrak{F}$. In **Teorema 50.6**, we proved the converse implications: if a **fórmula** on the right is valid in $\mathfrak{F}$, $\mathfrak{F}$ has the property on the left. $\square$

Problema 50.4. Show that if the **fórmula** on the right side of **Tabela 50.2** is valid in a frame $\mathfrak{F}$, then $\mathfrak{F}$ has the property on the left side. To do this, consider a frame that does *not* satisfy the property on the left, and define a suitable V such that the **fórmula** on the right is false at some world.

Teorema 50.6 also shows that the properties can be combined: for instance if both B and 4 are valid in $\mathfrak{F}$ then the frame is both symmetric and transitive, etc. Many important modal logics are characterized as the set of **fórmulas** valid in all frames that combine some frame properties, and so we can characterize them as the set of **fórmulas** valid in all frames in which the corresponding defining **fórmulas** are valid. For instance, the classical system **S4** is the set of all **fórmulas** valid in all reflexive and transitive frames, O.s., in all those where both T and 4 are valid. **S5** is the set of all formulas valid in all reflexive, symmetric, and euclidean frames, O.s., all those where all of T, B, and 5 are valid.

50.5. FIRST-ORDER DEFINABILITY

Logical relationships between properties of R in general correspond to relationships between the corresponding defining **fórmulas**. For instance, every reflexive relation is serial; hence, whenever T is valid in a frame, so is D . (Note that this relationship is *not* that of entailment. It is not the case that whenever $\mathfrak{M}, w \Vdash T$ then $\mathfrak{M}, w \Vdash D$.) We record some such relationships.

*nml:frd:def:
prop:relation-facts*

Proposição 50.9. *Let R be a binary relation on a set W ; then:*

1. *If R is reflexive, then it is serial.*
2. *If R is symmetric, then it is transitive if and only if it is euclidean.*
3. *If R is symmetric or euclidean then it is weakly directed (it has the “diamond property”).*
4. *If R is euclidean then it is weakly connected.*
5. *If R is functional then it is serial.*

Problema 50.5. Prove **Proposição 50.9**.

-NoValue-

`content/normal-modal-logic/frame-definability/first-order-definability.tex`

50.5 First-order Definability

*nml:frd:fol:
sec*

We’ve seen that a number of properties of accessibility relations of frames can be defined by modal **fórmulas**. For instance, symmetry of frames can be defined by the **fórmula** $B, p \rightarrow \Box \Diamond p$. The conditions we’ve encountered so far can all be expressed by first-order **fórmulas** in a **linguagem** involving a single two-place **símbolo de predicado**. For instance, symmetry is defined by $\forall x \forall y (Q(x, y) \rightarrow Q(y, x))$ in the sense that a first-order **estrutura** $\mathfrak{M}$ with $|\mathfrak{M}| = W$ and $Q^{\mathfrak{M}} = R$ satisfies the preceding formula iff R is symmetric. This suggests the following definition:

Definição 50.10. A class $\mathcal{F}$ of frames is *first-order definable* if there is a **sentença** φ in the first-order language with a single two-place **símbolo de predicado** Q such that $\mathfrak{F} = \langle W, R \rangle \in \mathcal{F}$ iff $\mathfrak{M} \models \varphi$ in the first-order **estrutura** $\mathfrak{M}$ with $|\mathfrak{M}| = W$ and $Q^{\mathfrak{M}} = R$.

It turns out that the properties and modal **fórmulas** that define them considered so far are exceptional. Not every **fórmula** defines a first-order definable class of frames, and not every first-order definable class of frames is definable by a modal formula.

A counterexample to the first is given by the Löb formula:

$$\Box(\Box p \rightarrow p) \rightarrow \Box p. \quad (W)$$

W defines the class of transitive and converse well-founded frames. A relation is well-founded if there is no infinite sequence $w_1, w_2, \dots$ such that $Rw_2w_1, Rw_3w_2, \dots$. For instance, the relation $<$ on $\mathbb{N}$ is well-founded, whereas the relation $<$ on $\mathbb{Z}$ is not. A relation is converse well-founded iff its converse is well-founded. So converse well-founded relations are those where there is no infinite sequence $w_1, w_2, \dots$ such that $Rw_1w_2, Rw_2w_3, \dots$.

There is, however, no first-order **fórmula** defining transitive converse well-founded relations. For suppose $\mathfrak{M} \models \beta$ iff $R = Q^{\mathfrak{M}}$ is transitive converse well-founded. Let φ_n be the **fórmula**

$$(Q(a_1, a_2) \wedge \dots \wedge Q(a_{n-1}, a_n))$$

Now consider the set of **fórmulas**

$$\Gamma = \{\beta, \varphi_1, \varphi_2, \dots\}.$$

Every finite subset of Γ is satisfiable: Let k be largest such that φ_k is in the subset, $|\mathfrak{M}_k| = \{1, \dots, k\}$, $a_i^{\mathfrak{M}_k} = i$, and $Q^{\mathfrak{M}_k} = <$. Since $<$ on $\{1, \dots, k\}$ is transitive and converse well-founded, $\mathfrak{M}_k \models \beta$. $\mathfrak{M}_k \models \varphi_i$ by construction, for all $i \leq k$. By the Compactness Theorem for first-order logic, Γ is satisfiable in some **estrutura** $\mathfrak{M}$. By hypothesis, since $\mathfrak{M} \models \beta$, the relation $Q^{\mathfrak{M}}$ is converse well-founded. But clearly, $a_1^{\mathfrak{M}}, a_2^{\mathfrak{M}}, \dots$ would form an infinite sequence of the kind ruled out by converse well-foundedness.

A counterexample to the second claim is given by the property of universality: for every u and v , Ruv . Universal frames are first-order definable by the formula $\forall x \forall y Q(x, y)$. However, no modal formula is valid in all and only the universal frames. This is a consequence of a result that is independently interesting: the formulas valid in universal frames are exactly the same as those valid in reflexive, symmetric, and transitive frames. There are reflexive, symmetric, and transitive frames that are not universal, hence every **fórmula** valid in all universal frames is also valid in some non-universal frames.

-NoValue-

[content/normal-modal-logic/frame-definability/equivalence-S5.tex](#)

50.6 Equivalence Relations and S5

The modal logic **S5** is characterized as the set of **fórmulas** valid on all universal frames, O.s., every world is accessible from every world, including itself. In such a scenario, $\Box$ corresponds to necessity and $\Diamond$ to possibility: $\Box\varphi$ is true if φ is true at *every* world, and $\Diamond\varphi$ is true if φ is true at *some* world. It turns out that **S5** can also be characterized as the **fórmulas** valid on all reflexive, symmetric, and transitive frames, O.s., on all *equivalence relations*.

nml:frd:es5:
sec

Definição 50.11. A binary relation R on W is an *equivalence relation* if and only if it is reflexive, symmetric and transitive. A relation R on W is *universal* if and only if Ruv for all $u, v \in W$.

50.6. EQUIVALENCE RELATIONS AND S5

Since T, B, and 4 characterize the reflexive, symmetric, and transitive frames, the frames where the accessibility relation is an equivalence relation are exactly those in which all three *fórmulas* are valid. It turns out that the equivalence relations can also be characterized by other combinations of *fórmulas*, since the conditions with which we've defined equivalence relations are equivalent to combinations of other familiar conditions on R .

*nml:frd:es5:
prop:equivalences*

Proposição 50.12. *The following are equivalent:*

1. R is an equivalence relation;
2. R is reflexive and euclidean;
3. R is serial, symmetric, and euclidean;
4. R is serial, symmetric, and transitive.

Prova. Exercise. □

Problema 50.6. Prove *Proposição 50.12* by showing:

1. If R is symmetric and transitive, it is euclidean.
2. If R is reflexive, it is serial.
3. If R is reflexive and euclidean, it is symmetric.
4. If R is symmetric and euclidean, it is transitive.
5. If R is serial, symmetric, and transitive, it is reflexive.

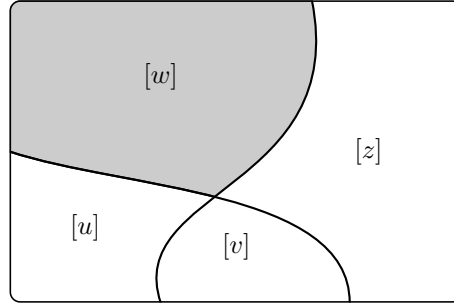
Explain why this suffices for the proof that the conditions are equivalent.

Proposição 50.12 is the semantic counterpart to *Proposição 51.30*, in that it gives an equivalent characterization of the modal logic of frames over which R is an equivalence relation (the logic traditionally referred to as **S5**).

What is the relationship between universal and equivalence relations? Although every universal relation is an equivalence relation, clearly not every equivalence relation is universal. However, the *fórmulas* valid on all universal relations are exactly the same as those valid on all equivalence relations.

Proposição 50.13. *Let R be an equivalence relation, and for each $w \in W$ define the equivalence class of w as the set $[w] = \{w' \in W : Rww'\}$. Then:*

1. $w \in [w]$;
2. R is universal on each equivalence class $[w]$;
3. The collection of equivalence classes partitions W into mutually exclusive and jointly exhaustive subsets.

Figura 50.2: A partition of W in equivalence classes.

Proposição 50.14. *A fórmula φ is valid in all frames $\mathfrak{F} = \langle W, R \rangle$ where R is an equivalence relation, if and only if it is valid in all frames $\mathfrak{F} = \langle W, R \rangle$ where R is universal. Hence, the logic of universal frames is just **S5**.*

nml:frd:es5:
fig:partition
nml:frd:es5:
prop:S5=univ

Prova. It's immediate to verify that a universal relation R on W is an equivalence. Hence, if φ is valid in all frames where R is an equivalence it is valid in all universal frames. For the other direction, we argue contrapositively: suppose ψ is a fórmula that fails at a world w in a model $\mathfrak{M} = \langle W, R, V \rangle$ based on a frame $\langle W, R \rangle$, where R is an equivalence on W . So $\mathfrak{M}, w \not\models \psi$. Define a model $\mathfrak{M}' = \langle W', R', V' \rangle$ as follows:

1. $W' = [w]$;
2. R' is universal on W' ;
3. $V'(p) = V(p) \cap W'$.

(So the set W' of worlds in $\mathfrak{M}'$ is represented by the shaded area in **Figura 50.2**.) It is easy to see that R and R' agree on W' . Then one can show by induction on fórmulas that for all $w' \in W'$: $\mathfrak{M}', w' \models \varphi$ if and only if $\mathfrak{M}, w' \models \varphi$ for each φ (this makes sense since $W' \subseteq W$). In particular, $\mathfrak{M}', w \not\models \psi$, and ψ fails in a model based on a universal frame. $\square$

-NoValue-

content/normal-modal-logic/frame-definability/second-order-definability.tex

50.7 Second-order Definability

Not every frame property definable by modal formulas is first-order definable. However, if we allow quantification over one-place predicates (O.s., monadic second-order quantification), we define all modally definable frame properties. The trick is to exploit a systematic way in which the conditions under which a modal fórmula is true at a world are related to first-order fórmulas. This is

nml:frd:st:
sec

50.7. SECOND-ORDER DEFINABILITY

the so-called standard translation of modal formulas into first-order formulas in a language containing not just a two-place **símbolo de predicado** Q for the accessibility relation, but also a one-place **símbolo de predicado** P_i for the **variáveis proposicionais** p_i occurring in φ .

Definição 50.15. The *standard translation* $ST_x(\varphi)$ is inductively defined as follows:

1. $\varphi \equiv \perp$: $ST_x(\varphi) = \perp$.
2. $\varphi \equiv \top$: $ST_x(\varphi) = \top$.
3. $\varphi \equiv p_i$: $ST_x(\varphi) = P_i(x)$.
4. $\varphi \equiv \neg\psi$: $ST_x(\varphi) = \neg ST_x(\psi)$.
5. $\varphi \equiv (\psi \wedge \chi)$: $ST_x(\varphi) = (ST_x(\psi) \wedge ST_x(\chi))$.
6. $\varphi \equiv (\psi \vee \chi)$: $ST_x(\varphi) = (ST_x(\psi) \vee ST_x(\chi))$.
7. $\varphi \equiv (\psi \rightarrow \chi)$: $ST_x(\varphi) = (ST_x(\psi) \rightarrow ST_x(\chi))$.
8. $\varphi \equiv (\psi \leftrightarrow \chi)$: $ST_x(\varphi) = (ST_x(\psi) \leftrightarrow ST_x(\chi))$.
9. $\varphi \equiv \Box\psi$: $ST_x(\varphi) = \forall y (Q(x, y) \rightarrow ST_y(\psi))$.
10. $\varphi \equiv \Diamond\psi$: $ST_x(\varphi) = \exists y (Q(x, y) \wedge ST_y(\psi))$.

For instance, $ST_x(\Box p \rightarrow p)$ is $\forall y (Q(x, y) \rightarrow P(y)) \rightarrow P(x)$. Any **estrutura** for the language of $ST_x(\varphi)$ requires a domain, a two-place relation assigned to Q , and subsets of the domain assigned to the one-place **símbolos de predicado** P_i . In other words, the components of such a **estrutura** are exactly those of a model for φ : the domain is the set of worlds, the two-place relation assigned to Q is the accessibility relation, and the subsets assigned to P_i are just the assignments $V(p_i)$. It won't surprise that satisfaction of φ in a modal model and of $ST_x(\varphi)$ in the corresponding **estrutura** agree:

Proposição 50.16. *Let $\mathfrak{M} = \langle W, R, V \rangle$, $\mathfrak{M}'$ be the first-order **estrutura** with $|\mathfrak{M}'| = W$, $Q^{\mathfrak{M}'} = R$, and $P_i^{\mathfrak{M}'} = V(p_i)$, and $s(x) = w$. Then*

$$\mathfrak{M}, w \Vdash \varphi \text{ iff } \mathfrak{M}', s \models ST_x(\varphi)$$

Prova. By induction on φ . □

Proposição 50.17. *Suppose φ is a modal **fórmula** and $\mathfrak{F} = \langle W, R \rangle$ is a frame. Let $\mathfrak{F}'$ be the first-order **estrutura** with $|\mathfrak{F}'| = W$ and $Q^{\mathfrak{F}'} = R$, and let φ' be the second-order formula*

$$\forall X_1 \dots \forall X_n \forall x ST_x(\varphi)[X_1/P_1, \dots, X_n/P_n],$$

*where $P_1, \dots, P_n$ are all one-place **símbolos de predicado** in $ST_x(\varphi)$. Then*

$$\mathfrak{F} \models \varphi \text{ iff } \mathfrak{F}' \models \varphi'$$

Prova. $\mathfrak{F}' \models \varphi'$ iff for every *estrutura* $\mathfrak{M}'$ where $P_i^{\mathfrak{M}'} \subseteq W$ for $i = 1, \dots, n$, and for every s with $s(x) \in W$, $\mathfrak{M}', s \models \text{ST}_x(\varphi)$. By **Proposição 50.16**, that is the case iff for all models $\mathfrak{M}$ based on $\mathfrak{F}$ and every world $w \in W$, $\mathfrak{M}, w \Vdash \varphi$, O.s., $\mathfrak{F} \models \varphi$. $\square$

Definição 50.18. A class $\mathcal{F}$ of frames is *second-order definable* if there is a *sentença* φ in the second-order language with a single two-place *símbolo de predicado* P and quantifiers only over monadic set variables such that $\mathfrak{F} = \langle W, R \rangle \in \mathcal{F}$ iff $\mathfrak{M} \models \varphi$ in the *estrutura* $\mathfrak{M}$ with $|\mathfrak{M}| = W$ and $P^{\mathfrak{M}} = R$.

Corolário 50.19. If a class of frames is definable by a *fórmula* φ , the corresponding class of accessibility relations is definable by a monadic second-order sentence.

Prova. The monadic second-order sentence φ' of the preceding proof has the required property. $\square$

As an example, consider again the *fórmula* $\Box p \rightarrow p$. It defines reflexivity. Reflexivity is of course first-order definable by the *sentença* $\forall x Q(x, x)$. But it is also definable by the monadic second-order *sentença*

$$\forall X \forall x (\forall y (Q(x, y) \rightarrow X(y)) \rightarrow X(x)).$$

This means, of course, that the two sentences are equivalent. Here's how you might convince yourself of this directly: First suppose the second-order sentence is true in a *estrutura* $\mathfrak{M}$. Since x and X are universally quantified, the remainder must hold for any $x \in W$ and set $X \subseteq W$, p.ex., the set $\{z : Rxz\}$ where $R = Q^{\mathfrak{M}}$. So, for any s with $s(x) \in W$ and $s(X) = \{z : Rxz\}$ we have $\mathfrak{M} \models \forall y (Q(x, y) \rightarrow X(y)) \rightarrow X(x)$. But by the way we've picked $s(X)$ that means $\mathfrak{M}, s \models \forall y (Q(x, y) \rightarrow Q(x, y)) \rightarrow Q(x, x)$, which is equivalent to $Q(x, x)$ since the antecedent is valid. Since $s(x)$ is arbitrary, we have $\mathfrak{M} \models \forall x Q(x, x)$.

Now suppose that $\mathfrak{M} \models \forall x Q(x, x)$ and show that $\mathfrak{M} \models \forall X \forall x (\forall y (Q(x, y) \rightarrow X(y)) \rightarrow X(x))$. Pick any assignment s , and assume $\mathfrak{M}, s \models \forall y (Q(x, y) \rightarrow X(y))$. Let s' be the y -variant of s with $s'(y) = s(x)$; we have $\mathfrak{M}, s' \models Q(x, y) \rightarrow X(y)$, O.s., $\mathfrak{M}, s \models Q(x, x) \rightarrow X(x)$. Since $\mathfrak{M} \models \forall x Q(x, x)$, the antecedent is true, and we have $\mathfrak{M}, s \models X(x)$, which is what we needed to show.

Since some definable classes of frames are not first-order definable, not every monadic second-order *sentença* of the form φ' is equivalent to a first-

order *sentença*. There is no effective method to decide which ones are.

Capítulo 51

Axiomatic Derivações

-NoValue-

`content/normal-modal-logic/axioms-systems/introduction.tex`

51.1 Introduction

nml:axs:int:
sec

We have a semantics for the basic modal language in terms of modal models, and a notion of a *fórmula* being valid—true at all worlds in all models—or valid with respect to some class of models or frames—true at all worlds in all models in the class, or based on the frame. Logic usually connects such semantic characterizations of validity with a proof-theoretic notion of *derivabilidade*. The aim is to define a notion of *derivabilidade* in some system such that a *fórmula* is *derivável* iff it is valid.

The simplest and historically oldest *derivação* systems are so-called Hilbert-type or axiomatic *derivação* systems. Hilbert-type *derivação* systems for many modal logics are relatively easy to construct: they are simple as objects of metatheoretical study (p.ex., to prove soundness and completeness). However, they are much harder to use to prove *fórmulas* in than, say, natural deduction systems.

In Hilbert-type *derivação* systems, a *derivação* of a *fórmula* is a sequence of *fórmulas* leading from certain axioms, via a handful of inference rules, to the *fórmula* in question. Since we want the *derivação* system to match the semantics, we have to guarantee that the set of *derivável* formulas are true in all models (or true in all models in which all axioms are true). We'll first isolate some properties of modal logics that are necessary for this to work: the “normal” modal logics. For normal modal logics, there are only two inference rules that need to be assumed: modus ponens and necessitation. As axioms we take all (substitution instances) of tautologies, and, depending on the modal logic we deal with, a number of modal axioms. Even if we are just interested in the class of all models, we must also count all substitution instances of K and Dual as axioms. This alone generates the minimal normal modal logic **K**.

Definição 51.1. The rule of *modus ponens* is the inference schema

$$\frac{\varphi \quad \varphi \rightarrow \psi}{\psi} \text{ MP}$$

We say a *fórmula* ψ follows from *fórmulas* φ, χ by modus ponens iff $\chi \equiv \varphi \rightarrow \psi$.

Definição 51.2. The rule of *necessitation* is the inference schema

$$\frac{\varphi}{\Box \varphi} \text{ NEC}$$

We say the *fórmula* ψ follows from the *fórmulas* φ by necessitation iff $\psi \equiv \Box \varphi$.

Definição 51.3. A *derivação* from a set of axioms Σ is a sequence of *fórmulas* $\psi_1, \psi_2, \dots, \psi_n$, where each ψ_i is either

1. a substitution instance of a tautology, or
2. a substitution instance of a *fórmula* in Σ , or
3. follows from two *fórmulas* ψ_j, ψ_k with $j, k < i$ by modus ponens, or
4. follows from a *fórmula* ψ_j with $j < i$ by necessitation.

If there is such a *derivação* with $\psi_n \equiv \varphi$, we say that φ is *derivável* from Σ , in symbols $\Sigma \vdash \varphi$.

With this definition, it will turn out that the set of *derivável* formulas forms a normal modal logic, and that any *derivável fórmula* is true in every model in which every axiom is true. This property of *derivações* is called *soundness*. The converse, *completeness*, is harder to prove.

-NoValue-

content/normal-modal-logic/axioms-systems/normal-logics.tex

51.2 Normal Modal Logics

Not every set of modal *fórmulas* can easily be characterized as those *fórmulas* derivable from a set of axioms. We want modal logics to be well-behaved. First of all, everything we can *deriva* in classical propositional logic should still be *derivável*, of course taking into account that the *fórmulas* may now contain also $\Box$ and $\Diamond$. To this end, we require that a modal logic contain all tautological instances and be closed under modus ponens.

Definição 51.4. A *modal logic* is a set Σ of modal *fórmulas* which

1. contains all tautologies, and

51.2. NORMAL MODAL LOGICS

2. is closed under substitution, O.s., if $\varphi \in \Sigma$, and $\theta_1, \dots, \theta_n$ are *fórmulas*, then

$$\varphi[\theta_1/p_1, \dots, \theta_n/p_n] \in \Sigma,$$

3. is closed under *modus ponens*, O.s., if φ and $\varphi \rightarrow \psi \in \Sigma$, then $\psi \in \Sigma$.

In order to use the relational semantics for modal logics, we also have to require that all *fórmulas* valid in all modal models are included. It turns out that this requirement is met as soon as all instances of K and DUAL are *derivável*, and whenever a *fórmula* φ is *derivável*, so is $\Box\varphi$. A modal logic that satisfies these conditions is called *normal*. (Of course, there are also non-normal modal logics, but the usual relational models are not adequate for them.)

Definição 51.5. A modal logic Σ is *normal* if it contains

$$\Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q), \quad (\text{K})$$

$$\Diamond p \leftrightarrow \neg \Box \neg p \quad (\text{DUAL})$$

and is closed under *necessitation*, O.s., if $\varphi \in \Sigma$, then $\Box\varphi \in \Sigma$.

Observe that while tautological implication is “fine-grained” enough to preserve *truth at a world*, the rule NEC only preserves *truth in a model* (and hence also validity in a frame or in a class of frames).

nml:prf:nor: **Proposição 51.6.** Every normal modal logic is closed under rule RK,
prop:rk

$$\frac{\varphi_1 \rightarrow (\varphi_2 \rightarrow \dots (\varphi_{n-1} \rightarrow \varphi_n) \dots)}{\Box\varphi_1 \rightarrow (\Box\varphi_2 \rightarrow \dots (\Box\varphi_{n-1} \rightarrow \Box\varphi_n) \dots)} \text{ RK}$$

Prova. By induction on n : If $n = 1$, then the rule is just NEC, and every normal modal logic is closed under NEC.

Now suppose the result holds for $n - 1$; we show it holds for n .

Assume

$$\varphi_1 \rightarrow (\varphi_2 \rightarrow \dots (\varphi_{n-1} \rightarrow \varphi_n) \dots) \in \Sigma$$

By the induction hypothesis, we have

$$\Box\varphi_1 \rightarrow (\Box\varphi_2 \rightarrow \dots \Box(\varphi_{n-1} \rightarrow \varphi_n) \dots) \in \Sigma$$

Since Σ is a normal modal logic, it contains all instances of K, in particular

$$\Box(\varphi_{n-1} \rightarrow \varphi_n) \rightarrow (\Box\varphi_{n-1} \rightarrow \Box\varphi_n) \in \Sigma$$

Using modus ponens and suitable tautological instances we get

$$\Box\varphi_1 \rightarrow (\Box\varphi_2 \rightarrow \dots (\Box\varphi_{n-1} \rightarrow \Box\varphi_n) \dots) \in \Sigma. \quad \square$$

Proposição 51.7. Every normal modal logic Σ contains $\neg\Diamond\perp$.

*nml:prf:nor:
prop:notDiamondBot*

Problema 51.1. Prove **Proposição 51.7**.

Proposição 51.8. Let $\varphi_1, \dots, \varphi_n$ be *fórmulas*. Then there is a smallest modal logic Σ containing all instances of $\varphi_1, \dots, \varphi_n$.

Prova. Given $\varphi_1, \dots, \varphi_n$, define Σ as the intersection of all normal modal logics containing all instances of $\varphi_1, \dots, \varphi_n$. The intersection is non-empty as $\text{Frm}(\mathcal{L})$, the set of all *fórmulas*, is such a modal logic. $\square$

Definição 51.9. The smallest normal modal logic containing $\varphi_1, \dots, \varphi_n$ is called a *modal system* and denoted by $\mathbf{K}\varphi_1 \dots \varphi_n$. The smallest normal modal logic is denoted by $\mathbf{K}$.

-NoValue-

`content/normal-modal-logic/axioms-systems/logics-proofs.tex`

51.3 Derivações e Modal Systems

We first define what a *derivação* is for normal modal logics. Roughly, a *derivação* is a sequence of *fórmulas* in which every *membro* is either (a substitution instance of) one of a number of *axioms*, or follows from previous *membros* by one of a few inference rules. For normal modal logics, all instances of tautologies, K, and DUAL count as axioms. This results in the modal system $\mathbf{K}$, the smallest normal modal logic. We may wish to add additional axioms to obtain other systems, however. The rules are always modus ponens MP and necessitation NEC.

*nml:prf:prf:
sec*

Definição 51.10. Given a modal system $\mathbf{K}\varphi_1 \dots \varphi_n$ and a *fórmula* ψ we say that ψ is *derivável* in $\mathbf{K}\varphi_1 \dots \varphi_n$, written $\mathbf{K}\varphi_1 \dots \varphi_n \vdash \psi$, if and only if there are *fórmulas* $\chi_1, \dots, \chi_k$ such that $\chi_k = \psi$ and each χ_i is either a tautological instance, or an instance of one of K, DUAL, $\varphi_1, \dots, \varphi_n$, or it follows from previous *fórmulas* by means of the rules MP or NEC.

The following proposition allows us to show that $\psi \in \Sigma$ by exhibiting a Σ -*derivação* of ψ .

Proposição 51.11. $\mathbf{K}\varphi_1 \dots \varphi_n = \{\psi : \mathbf{K}\varphi_1 \dots \varphi_n \vdash \psi\}$.

Prova. We use induction on the length of *derivações* to show that $\{\psi : \mathbf{K}\varphi_1 \dots \varphi_n \vdash \psi\} \subseteq \mathbf{K}\varphi_1 \dots \varphi_n$.

If the *derivação* of ψ has length 1, it contains a single *fórmula*. That *fórmula* cannot follow from previous formulas by MP or NEC, so must be a tautological instance, an instance of K, DUAL, or an instance of one of $\varphi_1, \dots, \varphi_n$. But $\mathbf{K}\varphi_1 \dots \varphi_n$ contains these as well, so $\psi \in \mathbf{K}\varphi_1 \dots \varphi_n$.

51.4. PROOFS IN K

If the *derivação* of ψ has length > 1 , then ψ may in addition be obtained by MP or NEC from *fórmulas* not occurring as the last line in the *derivação*. If ψ follows from χ and $\chi \rightarrow \psi$ (by MP), then χ and $\chi \rightarrow \psi \in \mathbf{K}\varphi_1 \dots \varphi_n$ by induction hypothesis. But every modal logic is closed under modus ponens, so $\psi \in \mathbf{K}\varphi_1 \dots \varphi_n$. If $\psi \equiv \Box\chi$ follows from χ by NEC, then $\chi \in \mathbf{K}\varphi_1 \dots \varphi_n$ by induction hypothesis. But every normal modal logic is closed under NEC, so $\psi \in \mathbf{K}\varphi_1 \dots \varphi_n$.

The converse inclusion follows by showing that $\Sigma = \{\psi : \mathbf{K}\varphi_1 \dots \varphi_n \vdash \psi\}$ is a normal modal logic containing all the instances of $\varphi_1, \dots, \varphi_n$, and the observation that $\mathbf{K}\varphi_1 \dots \varphi_n$ is, by definition, the smallest such logic.

1. Every tautology ψ is a tautological instance, so $\mathbf{K}\varphi_1 \dots \varphi_n \vdash \psi$, so Σ contains all tautologies.
2. If $\mathbf{K}\varphi_1 \dots \varphi_n \vdash \chi$ and $\mathbf{K}\varphi_1 \dots \varphi_n \vdash \chi \rightarrow \psi$, then $\mathbf{K}\varphi_1 \dots \varphi_n \vdash \psi$: Combine the *derivação* of χ with that of $\chi \rightarrow \psi$, and add the line ψ . The last line is justified by MP. So Σ is closed under modus ponens.
3. If ψ has a *derivação*, then every substitution instance of ψ also has a *derivação*: apply the substitution to every *fórmula* in the *derivação*. (Exercise: prove by induction on the length of *derivações* that the result is also a correct *derivação*). So Σ is closed under uniform substitution. (We have now established that Σ satisfies all conditions of a modal logic.)
4. We have $\mathbf{K}\varphi_1 \dots \varphi_n \vdash K$, so $K \in \Sigma$.
5. We have $\mathbf{K}\varphi_1 \dots \varphi_n \vdash \text{DUAL}$, so $\text{DUAL} \in \Sigma$.
6. If $\mathbf{K}\varphi_1 \dots \varphi_n \vdash \chi$, the additional line $\Box\chi$ is justified by NEC. Consequently, Σ is closed under NEC. Thus, Σ is normal. $\square$

-NoValue-

content/normal-modal-logic/axioms-systems/proofs-in-K.tex

51.4 Proofs in K

nml:prf:prk:
sec In order to practice proofs in the smallest modal system, we show the valid *fórmulas* on the left-hand side of **Tabela 49.1** can all be given **K**-proofs.

Proposição 51.12. $\mathbf{K} \vdash \Box\varphi \rightarrow \Box(\psi \rightarrow \varphi)$

Prova.

1.	$\varphi \rightarrow (\psi \rightarrow \varphi)$	TAUT	
2.	$\Box(\varphi \rightarrow (\psi \rightarrow \varphi))$	NEC, 1	
3.	$\Box(\varphi \rightarrow (\psi \rightarrow \varphi)) \rightarrow (\Box\varphi \rightarrow \Box(\psi \rightarrow \varphi))$	K	
4.	$\Box\varphi \rightarrow \Box(\psi \rightarrow \varphi)$	MP, 2, 3	$\square$

Proposição 51.13. $\mathbf{K} \vdash \Box(\varphi \wedge \psi) \rightarrow (\Box\varphi \wedge \Box\psi)$

Prova.

1.	$(\varphi \wedge \psi) \rightarrow \varphi$	TAUT
2.	$\Box((\varphi \wedge \psi) \rightarrow \varphi)$	NEC
3.	$\Box((\varphi \wedge \psi) \rightarrow \varphi) \rightarrow (\Box(\varphi \wedge \psi) \rightarrow \Box\varphi)$	K
4.	$\Box(\varphi \wedge \psi) \rightarrow \Box\varphi$	MP, 2, 3
5.	$(\varphi \wedge \psi) \rightarrow \psi$	TAUT
6.	$\Box((\varphi \wedge \psi) \rightarrow \psi)$	NEC
7.	$\Box((\varphi \wedge \psi) \rightarrow \psi) \rightarrow (\Box(\varphi \wedge \psi) \rightarrow \Box\psi)$	K
8.	$\Box(\varphi \wedge \psi) \rightarrow \Box\psi$	MP, 6, 7
9.	$(\Box(\varphi \wedge \psi) \rightarrow \Box\varphi) \rightarrow$ $((\Box(\varphi \wedge \psi) \rightarrow \Box\psi) \rightarrow$ $(\Box(\varphi \wedge \psi) \rightarrow (\Box\varphi \wedge \Box\psi)))$	TAUT
10.	$(\Box(\varphi \wedge \psi) \rightarrow \Box\psi) \rightarrow$ $(\Box(\varphi \wedge \psi) \rightarrow (\Box\varphi \wedge \Box\psi))$	MP, 4, 9
11.	$\Box(\varphi \wedge \psi) \rightarrow (\Box\varphi \wedge \Box\psi)$	MP, 8, 10.

Note that the fórmula on line 9 is an instance of the tautology

$$(p \rightarrow q) \rightarrow ((p \rightarrow r) \rightarrow (p \rightarrow (q \wedge r))).$$

□

Proposição 51.14. $\mathbf{K} \vdash (\Box\varphi \wedge \Box\psi) \rightarrow \Box(\varphi \wedge \psi)$

Prova.

1.	$\varphi \rightarrow (\psi \rightarrow (\varphi \wedge \psi))$	TAUT
2.	$\Box(\varphi \rightarrow (\psi \rightarrow (\varphi \wedge \psi)))$	NEC, 1
3.	$\Box(\varphi \rightarrow (\psi \rightarrow (\varphi \wedge \psi))) \rightarrow (\Box\varphi \rightarrow \Box(\psi \rightarrow (\varphi \wedge \psi)))$	K
4.	$\Box\varphi \rightarrow \Box(\psi \rightarrow (\varphi \wedge \psi))$	MP, 2, 3
5.	$\Box(\psi \rightarrow (\varphi \wedge \psi)) \rightarrow (\Box\psi \rightarrow \Box(\varphi \wedge \psi))$	K
6.	$(\Box\varphi \rightarrow \Box(\psi \rightarrow (\varphi \wedge \psi))) \rightarrow$ $(\Box(\psi \rightarrow (\varphi \wedge \psi)) \rightarrow (\Box\psi \rightarrow \Box(\varphi \wedge \psi))) \rightarrow$ $(\Box\varphi \rightarrow (\Box\psi \rightarrow \Box(\varphi \wedge \psi)))$	TAUT
7.	$(\Box(\psi \rightarrow (\varphi \wedge \psi)) \rightarrow (\Box\psi \rightarrow \Box(\varphi \wedge \psi))) \rightarrow$ $(\Box\varphi \rightarrow (\Box\psi \rightarrow \Box(\varphi \wedge \psi)))$	MP, 4, 6
8.	$\Box\varphi \rightarrow (\Box\psi \rightarrow \Box(\varphi \wedge \psi))$	MP, 5, 7
9.	$(\Box\varphi \rightarrow (\Box\psi \rightarrow \Box(\varphi \wedge \psi))) \rightarrow$ $((\Box\varphi \wedge \Box\psi) \rightarrow \Box(\varphi \wedge \psi))$	TAUT
10.	$(\Box\varphi \wedge \Box\psi) \rightarrow \Box(\varphi \wedge \psi)$	MP, 8, 9

The fórmulas on lines 6 and 9 are instances of the tautologies

$$(p \rightarrow q) \rightarrow ((q \rightarrow r) \rightarrow (p \rightarrow r))$$

$$(p \rightarrow (q \rightarrow r)) \rightarrow ((p \wedge q) \rightarrow r)$$

□

51.5. DERIVED RULES

Proposição 51.15. $\mathbf{K} \vdash \neg \Box p \rightarrow \Diamond \neg p$

Prova.

1.	$\Diamond \neg p \leftrightarrow \neg \Box \neg \neg p$	DUAL
2.	$(\Diamond \neg p \leftrightarrow \neg \Box \neg \neg p) \rightarrow$ $(\neg \Box \neg \neg p \rightarrow \Diamond \neg p)$	TAUT
3.	$\neg \Box \neg \neg p \rightarrow \Diamond \neg p$	MP, 1, 2
4.	$\neg \neg p \rightarrow p$	TAUT
5.	$\Box(\neg \neg p \rightarrow p)$	NEC, 4
6.	$\Box(\neg \neg p \rightarrow p) \rightarrow (\Box \neg \neg p \rightarrow \Box p)$	K
7.	$(\Box \neg \neg p \rightarrow \Box p)$	MP, 5, 6
8.	$(\Box \neg \neg p \rightarrow \Box p) \rightarrow (\neg \Box p \rightarrow \neg \Box \neg \neg p)$	TAUT
9.	$\neg \Box p \rightarrow \neg \Box \neg \neg p$	MP, 7, 8
10.	$(\neg \Box p \rightarrow \neg \Box \neg \neg p) \rightarrow$ $((\neg \Box \neg \neg p \rightarrow \Diamond \neg p) \rightarrow (\neg \Box p \rightarrow \Diamond \neg p))$	TAUT
11.	$(\neg \Box \neg \neg p \rightarrow \Diamond \neg p) \rightarrow (\neg \Box p \rightarrow \Diamond \neg p)$	MP, 9, 10
12.	$\neg \Box p \rightarrow \Diamond \neg p$	MP, 3, 11

The **fórmulas** on lines 8 and 10 are instances of the tautologies

$$(p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p)$$

$$(p \rightarrow q) \rightarrow ((q \rightarrow r) \rightarrow (p \rightarrow r)).$$

□

Problema 51.2. Find **derivações** in **K** for the following **fórmulas**:

1. $\Box \neg p \rightarrow \Box(p \rightarrow q)$
2. $(\Box p \vee \Box q) \rightarrow \Box(p \vee q)$
3. $\Diamond p \rightarrow \Diamond(p \vee q)$

-NoValue-

content/normal-modal-logic/axioms-systems/derived-rules.tex

51.5 Derived Rules

nml:prf:der:
sec

Finding and writing **derivações** is obviously difficult, cumbersome, and repetitive. For instance, very often we want to pass from $\varphi \rightarrow \psi$ to $\Box \varphi \rightarrow \Box \psi$, O.s., apply rule RK. That requires an application of NEC, then recording the proper instance of K, then applying MP. Passing from $\varphi \rightarrow \psi$ and $\psi \rightarrow \chi$ to $\varphi \rightarrow \chi$ requires recording the (long) tautological instance

$$(\varphi \rightarrow \psi) \rightarrow ((\psi \rightarrow \chi) \rightarrow (\varphi \rightarrow \chi))$$

and applying MP twice. Often we want to replace a sub-**fórmula** by a formula we know to be equivalent, p.ex., $\Diamond \varphi$ by $\neg \Box \neg \varphi$, or $\neg \neg \varphi$ by φ . So rather than

write out the actual **derivação**, it is more convenient to simply record why the intermediate steps are **derivável**. For this purpose, let us collect some facts about **derivabilidade**.

Proposição 51.16. *If $\mathbf{K} \vdash \varphi_1, \dots, \mathbf{K} \vdash \varphi_n$, and ψ follows from $\varphi_1, \dots, \varphi_n$ by propositional logic, then $\mathbf{K} \vdash \psi$.*

Prova. If ψ follows from $\varphi_1, \dots, \varphi_n$ by propositional logic, then

$$\varphi_1 \rightarrow (\varphi_2 \rightarrow \dots (\varphi_n \rightarrow \psi) \dots)$$

is a tautological instance. Applying MP n times gives a **derivação** of ψ . □

We will indicate use of this proposition by PL.

Proposição 51.17. *If $\mathbf{K} \vdash \varphi_1 \rightarrow (\varphi_2 \rightarrow \dots (\varphi_{n-1} \rightarrow \varphi_n) \dots)$ then $\mathbf{K} \vdash \Box \varphi_1 \rightarrow (\Box \varphi_2 \rightarrow \dots (\Box \varphi_{n-1} \rightarrow \Box \varphi_n) \dots)$.*

Prova. By induction on n , just as in the proof of **Proposição 51.6**. □

We will indicate use of this proposition by RK. Let's illustrate how these results help establishing **derivabilidade** results more easily.

Proposição 51.18. $\mathbf{K} \vdash (\Box \varphi \wedge \Box \psi) \rightarrow \Box(\varphi \wedge \psi)$

Prova.

1.	$\mathbf{K} \vdash \varphi \rightarrow (\psi \rightarrow (\varphi \wedge \psi))$	TAUT	
2.	$\mathbf{K} \vdash \Box \varphi \rightarrow (\Box \psi \rightarrow \Box(\varphi \wedge \psi))$	RK, 1	
3.	$\mathbf{K} \vdash (\Box \varphi \wedge \Box \psi) \rightarrow \Box(\varphi \wedge \psi)$	PL, 2	□

Proposição 51.19. *If $\mathbf{K} \vdash \varphi \leftrightarrow \psi$ and $\mathbf{K} \vdash \chi[\varphi/q]$ then $\mathbf{K} \vdash \chi[B/q]$*

*nml:prf:der:
prop:rewriting*

Prova. Exercise. □

Problema 51.3. Prove **Proposição 51.19** by proving, by induction on the complexity of χ , that if $\mathbf{K} \vdash \varphi \leftrightarrow \psi$ then $\mathbf{K} \vdash \chi[\varphi/q] \leftrightarrow \chi[\psi/q]$.

This proposition comes in handy especially when we want to convert $\Diamond$ into $\Box$ (or vice versa), or remove double negations inside a **fórmula**. In what follows, we will mark applications of **Proposição 51.19** by “ φ for ψ ” whenever we re-write a **fórmula** $\chi(\psi)$ for $\chi(\varphi)$. In other words, “ φ for ψ ” abbreviates:

$$\begin{array}{ll} \vdash \chi(\varphi) & \\ \vdash \varphi \leftrightarrow \psi & \\ \vdash \chi(\psi) & \text{by } \textbf{Proposição 51.19} \end{array}$$

For instance:

Proposição 51.20. $\mathbf{K} \vdash \neg \Box p \rightarrow \Diamond \neg p$

51.6. MORE PROOFS IN $\mathbf{K}$

Prova.

1. $\mathbf{K} \vdash \Diamond \neg p \leftrightarrow \neg \Box \neg \neg p$ DUAL
2. $\mathbf{K} \vdash \neg \Box \neg \neg p \rightarrow \Diamond \neg p$ PL, 1
3. $\mathbf{K} \vdash \neg \Box p \rightarrow \Diamond \neg p$ p for $\neg \neg p$ □

In the above *derivação*, the final step “ p for $\neg \neg p$ ” is short for

$$\begin{array}{ll} \mathbf{K} \vdash \neg \Box \neg \neg p \rightarrow \Diamond \neg p & \\ \mathbf{K} \vdash \neg \neg p \leftrightarrow p & \text{TAUT} \\ \mathbf{K} \vdash \neg \Box p \rightarrow \Diamond \neg p & \text{by Proposição 51.19} \end{array}$$

The roles of $\chi(q)$, φ , and ψ in *Proposição 51.19* are played here, respectively, by $\neg \Box q \rightarrow \Diamond \neg p$, $\neg \neg p$, and p .

When a *fórmula* contains a sub-*fórmula* $\neg \Diamond \varphi$, we can replace it by $\Box \neg \varphi$ using *Proposição 51.19*, since $\mathbf{K} \vdash \neg \Diamond \varphi \leftrightarrow \Box \neg \varphi$. We’ll indicate this and similar replacements simply by “ $\Box \neg$ for $\neg \Diamond$.”

The following proposition justifies that we can establish *derivabilidade* results schematically. P.ex., the previous proposition does not just establish that $\mathbf{K} \vdash \neg \Box p \rightarrow \Diamond \neg p$, but $\mathbf{K} \vdash \neg \Box \varphi \rightarrow \Diamond \neg \varphi$ for arbitrary φ .

Proposição 51.21. *If φ is a substitution instance of ψ and $\mathbf{K} \vdash \psi$, then $\mathbf{K} \vdash \varphi$.*

Prova. It is tedious but routine to verify (by induction on the length of the *derivação* of ψ) that applying a substitution to an entire *derivação* also results in a correct *derivação*. Specifically, substitution instances of tautological instances are themselves tautological instances, substitution instances of instances of DUAL and K are themselves instances of DUAL and K, and applications of MP and NEC remain correct when substituting *fórmulas* for *variáveis proposicionais* in both premise(s) and conclusion. □

-NoValue-

content/normal-modal-logic/axioms-systems/more-proofs-in-K.tex

51.6 More Proofs in $\mathbf{K}$

nml:prf:mpr:sec Let’s see some more examples of *derivabilidade* in $\mathbf{K}$, now using the simplified method introduced in *Seção 51.5*.

Proposição 51.22. $\mathbf{K} \vdash \Box(\varphi \rightarrow \psi) \rightarrow (\Diamond \varphi \rightarrow \Diamond \psi)$

Prova.

1. $\mathbf{K} \vdash (\varphi \rightarrow \psi) \rightarrow (\neg \psi \rightarrow \neg \varphi)$ PL
2. $\mathbf{K} \vdash \Box(\varphi \rightarrow \psi) \rightarrow (\Box \neg \psi \rightarrow \Box \neg \varphi)$ RK, 1
3. $\mathbf{K} \vdash (\Box \neg \psi \rightarrow \Box \neg \varphi) \rightarrow (\neg \Box \neg \varphi \rightarrow \neg \Box \neg \psi)$ TAUT
4. $\mathbf{K} \vdash \Box(\varphi \rightarrow \psi) \rightarrow (\neg \Box \neg \varphi \rightarrow \neg \Box \neg \psi)$ PL, 2, 3
5. $\mathbf{K} \vdash \Box(\varphi \rightarrow \psi) \rightarrow (\Diamond \varphi \rightarrow \Diamond \psi)$ $\Diamond$ for $\neg \Box \neg$. □

Proposição 51.23. $\mathbf{K} \vdash \Box\varphi \rightarrow (\Diamond(\varphi \rightarrow \psi) \rightarrow \Diamond\psi)$

Prova.

1. $\mathbf{K} \vdash \varphi \rightarrow (\neg\psi \rightarrow \neg(\varphi \rightarrow \psi))$ TAUT
2. $\mathbf{K} \vdash \Box\varphi \rightarrow (\Box\neg\psi \rightarrow \Box\neg(\varphi \rightarrow \psi))$ RK, 1
3. $\mathbf{K} \vdash \Box\varphi \rightarrow (\neg\Box\neg(\varphi \rightarrow \psi) \rightarrow \neg\Box\neg\psi)$ PL, 2
4. $\mathbf{K} \vdash \Box\varphi \rightarrow (\Diamond(\varphi \rightarrow \psi) \rightarrow \Diamond\psi)$ $\Diamond$ for $\neg\Box\neg$. □

Proposição 51.24. $\mathbf{K} \vdash (\Diamond\varphi \vee \Diamond\psi) \rightarrow \Diamond(\varphi \vee \psi)$

Prova.

1. $\mathbf{K} \vdash \neg(\varphi \vee \psi) \rightarrow \neg\varphi$ TAUT
2. $\mathbf{K} \vdash \Box\neg(\varphi \vee \psi) \rightarrow \Box\neg\varphi$ RK, 1
3. $\mathbf{K} \vdash \neg\Box\neg\varphi \rightarrow \neg\Box\neg(\varphi \vee \psi)$ PL, 2
4. $\mathbf{K} \vdash \Diamond\varphi \rightarrow \Diamond(\varphi \vee \psi)$ $\Diamond$ for $\neg\Box\neg$
5. $\mathbf{K} \vdash \Diamond\psi \rightarrow \Diamond(\varphi \vee \psi)$ similarly
6. $\mathbf{K} \vdash (\Diamond\varphi \vee \Diamond\psi) \rightarrow \Diamond(\varphi \vee \psi)$ PL, 4, 5. □

Proposição 51.25. $\mathbf{K} \vdash \Diamond(\varphi \vee \psi) \rightarrow (\Diamond\varphi \vee \Diamond\psi)$

Prova.

1. $\mathbf{K} \vdash \neg\varphi \rightarrow (\neg\psi \rightarrow \neg(\varphi \vee \psi))$ TAUT
2. $\mathbf{K} \vdash \Box\neg\varphi \rightarrow (\Box\neg\psi \rightarrow \Box\neg(\varphi \vee \psi))$ RK
3. $\mathbf{K} \vdash \Box\neg\varphi \rightarrow (\neg\Box\neg(\varphi \vee \psi) \rightarrow \neg\Box\neg\psi)$ PL, 2
4. $\mathbf{K} \vdash \neg\Box\neg(\varphi \vee \psi) \rightarrow (\Box\neg\varphi \rightarrow \neg\Box\neg\psi)$ PL, 3
5. $\mathbf{K} \vdash \neg\Box\neg(\varphi \vee \psi) \rightarrow (\neg\neg\Box\neg\psi \rightarrow \neg\Box\neg\varphi)$ PL, 4
6. $\mathbf{K} \vdash \Diamond(\varphi \vee \psi) \rightarrow (\neg\Diamond\psi \rightarrow \Diamond\varphi)$ $\Diamond$ for $\neg\Box\neg$
7. $\mathbf{K} \vdash \Diamond(\varphi \vee \psi) \rightarrow (\Diamond\psi \vee \Diamond\varphi)$ PL, 6. □

Problema 51.4. Show that the following *derivabilidade* claims hold:

1. $\mathbf{K} \vdash \Diamond\neg\perp \rightarrow (\Box\varphi \rightarrow \Diamond\varphi);$
2. $\mathbf{K} \vdash \Box(\varphi \vee \psi) \rightarrow (\Diamond\varphi \vee \Box\psi);$
3. $\mathbf{K} \vdash (\Diamond\varphi \rightarrow \Box\psi) \rightarrow \Box(\varphi \rightarrow \psi).$

-NoValue-

51.7. DUAL FÓRMULAS

51.7 Dual Fórmulas

nml:prf:dua:
nml:prf:dua:
def:duals

Definição 51.26. Each of the fórmulas T, B, 4, and 5 has a *dual*, denoted by a subscripted diamond, as follows:

$$\begin{aligned} p &\rightarrow \Diamond p & (T_{\Diamond}) \\ \Diamond \Box p &\rightarrow p & (B_{\Diamond}) \\ \Diamond \Diamond p &\rightarrow \Diamond p & (4_{\Diamond}) \\ \Diamond \Box p &\rightarrow \Box p & (5_{\Diamond}) \end{aligned}$$

Each of the above dual fórmulas is obtained from the corresponding fórmula by substituting $\neg p$ for p , contraposing, replacing $\neg \Box \neg$ by $\Diamond$, and replacing $\neg \Diamond \neg$ by $\Box$. D, O.s., $\Box \varphi \rightarrow \Diamond \varphi$ is its own dual in that sense.

nml:prf:dua:
prop:dualsys

Proposição 51.27. For each fórmula φ in Definição 51.26: $\mathbf{K}\varphi = \mathbf{K}\varphi_{\Diamond}$.

Prova. Exercise. □

Problema 51.5. Prove Proposição 51.27.

-NoValue-

content/normal-modal-logic/axioms-systems/proofs-modal-systems.tex

51.8 Proofs in Modal Systems

nml:prf:prs:
sec

We now come to proofs in systems of modal logic other than **K**.

nml:prf:prs:
prop:S5facts

Proposição 51.28. The following provability results obtain:

1. **KT5** $\vdash$ B;
2. **KT5** $\vdash$ 4;
3. **KDB4** $\vdash$ T;
4. **KB4** $\vdash$ 5;
5. **KB5** $\vdash$ 4;
6. **KT** $\vdash$ D.

nml:prf:prs:
prop:S5facts-KT-D

Prova. We exhibit proofs for each.

1. **KT5** $\vdash$ B:
 1. **KT5** $\vdash \Diamond \varphi \rightarrow \Box \Diamond \varphi$ 5
 2. **KT5** $\vdash \varphi \rightarrow \Diamond \varphi$ $T_{\Diamond}$
 3. **KT5** $\vdash \varphi \rightarrow \Box \Diamond \varphi$ PL, 2, 1.

2. **KT5** $\vdash$ 4:

1. **KT5** $\vdash \Diamond \Box \varphi \rightarrow \Box \Diamond \Box \varphi$ 5 with $\Box \varphi$ for p
2. **KT5** $\vdash \Box \varphi \rightarrow \Diamond \Box \varphi$ $T_{\Diamond}$ with $\Box \varphi$ for p
3. **KT5** $\vdash \Box \varphi \rightarrow \Box \Diamond \Box \varphi$ PL, 2, 1
4. **KT5** $\vdash \Diamond \Box \varphi \rightarrow \Box \varphi$ $5_{\Diamond}$
5. **KT5** $\vdash \Box \Diamond \Box \varphi \rightarrow \Box \Box \varphi$ RK, 4
6. **KT5** $\vdash \Box \varphi \rightarrow \Box \Box \varphi$ PL, 3, 5.

3. **KDB4** $\vdash$ T:

1. **KDB4** $\vdash \Diamond \Box \varphi \rightarrow \varphi$ $B_{\Diamond}$
2. **KDB4** $\vdash \Box \Box \varphi \rightarrow \Diamond \Box \varphi$ D with $\Box \varphi$ for p
3. **KDB4** $\vdash \Box \Box \varphi \rightarrow \varphi$ PL, 1, 2
4. **KDB4** $\vdash \Box \varphi \rightarrow \Box \Box \varphi$ 4
5. **KDB4** $\vdash \Box \varphi \rightarrow \varphi$ PL, 4, 3.

4. **KB4** $\vdash$ 5:

1. **KB4** $\vdash \Diamond \varphi \rightarrow \Box \Diamond \Diamond \varphi$ B with $\Diamond \varphi$ for p
2. **KB4** $\vdash \Diamond \Diamond \varphi \rightarrow \Diamond \varphi$ $4_{\Diamond}$
3. **KB4** $\vdash \Box \Diamond \Diamond \varphi \rightarrow \Box \Diamond \varphi$ RK, 2
4. **KB4** $\vdash \Diamond \varphi \rightarrow \Box \Diamond \varphi$ PL, 1, 3.

5. **KB5** $\vdash$ 4:

1. **KB5** $\vdash \Box \varphi \rightarrow \Box \Diamond \Box \varphi$ B with $\Box \varphi$ for p
2. **KB5** $\vdash \Diamond \Box \varphi \rightarrow \Box \varphi$ $5_{\Diamond}$
3. **KB5** $\vdash \Box \Diamond \Box \varphi \rightarrow \Box \Box \varphi$ RK, 2
4. **KB5** $\vdash \Box \varphi \rightarrow \Box \Box \varphi$ PL, 1, 3.

6. **KT** $\vdash$ D:

1. **KT** $\vdash \Box \varphi \rightarrow \varphi$ T
2. **KT** $\vdash \varphi \rightarrow \Diamond \varphi$ $T_{\Diamond}$
3. **KT** $\vdash \Box \varphi \rightarrow \Diamond \varphi$ PL, 1, 2

□

Definição 51.29. Following tradition, we define **S4** to be the system **KT4**, and **S5** the system **KT5**.

The following proposition shows that the classical system **S5** has several equivalent axiomatizations. This should not surprise, as the various combinations of axioms all characterize equivalence relations (see [Proposição 50.12](#)).

Proposição 51.30. **KT5** = **KDB4** = **KDB5**.

*nml:prf:prs:
prop:S5*

Prova. Exercise.

□

51.9. SOUNDNESS

Problema 51.6. Prove [Proposição 51.30](#).

-NoValue-

[content/normal-modal-logic/axioms-systems/soundness.tex](#)

51.9 Soundness

nml:prf:snd:
sec

A [derivação](#) system is called sound if everything that can be [derivad](#) is valid. When considering modal systems, O.s., [derivações](#) where in addition to K we can use instances of some [fórmulas](#) $\varphi_1, \dots, \varphi_n$, we want every [derivável](#) formula to be true in any model in which $\varphi_1, \dots, \varphi_n$ are true.

nml:prf:snd:
thm:soundness

Teorema 51.31 (Soundness Theorem). *If every instance of $\varphi_1, \dots, \varphi_n$ is valid in the classes of models $\mathcal{C}_1, \dots, \mathcal{C}_n$, respectively, then $\mathbf{K}\varphi_1 \dots \varphi_n \vdash \psi$ implies that ψ is valid in the class of models $\mathcal{C}_1 \cap \dots \cap \mathcal{C}_n$.*

Prova. By induction on length of proofs. For brevity, put $\mathcal{C} = \mathcal{C}_1 \cap \dots \cap \mathcal{C}_n$.

1. Induction Basis: If ψ has a proof of length 1, then it is either a tautological instance, an instance of K, or of DUAL, or an instance of one of $\varphi_1, \dots, \varphi_n$. In the first case, ψ is valid in $\mathcal{C}$, since tautological instance are valid in *any* class of models, by [Proposição 49.15](#). Similarly in the second case, by [Proposição 49.18](#) and [Proposição 49.19](#). Finally in the third case, since ψ is valid in $\mathcal{C}_i$ and $\mathcal{C} \subseteq \mathcal{C}_i$, we have that ψ is valid in $\mathcal{C}$ as well by [Proposição 49.11](#).
2. Inductive step: Suppose ψ has a proof of length $k > 1$. If ψ is a tautological instance or an instance of one of $\varphi_1, \dots, \varphi_n$, we proceed as in the previous step. So suppose ψ is obtained by MP from previous [fórmulas](#) $\chi \rightarrow \psi$ and χ . Then $\chi \rightarrow \psi$ and χ have proofs of length $< k$, and by inductive hypothesis they are valid in $\mathcal{C}$. By [Proposição 49.20](#), ψ is valid in $\mathcal{C}$ as well. Finally suppose ψ is obtained by NEC from χ (so that $\psi = \Box\chi$). By inductive hypothesis, χ is valid in $\mathcal{C}$, and by [Proposição 49.12](#) so is ψ .
□

-NoValue-

[content/normal-modal-logic/axioms-systems/systems-distinct.tex](#)

51.10 Showing Systems are Distinct

nml:prf:dis:
sec

In [Seção 51.8](#) we saw how to prove that two systems of modal logic are in fact the same system. [Teorema 51.31](#) allows us to show that two modal systems Σ and Σ' are distinct, by finding a [fórmula](#) φ such that $\Sigma' \vdash \varphi$ that fails in a model of Σ .

Proposição 51.32. $\mathbf{KD} \subsetneq \mathbf{KT}$

Prova. This is the syntactic counterpart to the semantic fact that all reflexive relations are serial. To show $\mathbf{KD} \subseteq \mathbf{KT}$ we need to see that $\mathbf{KD} \vdash \psi$ implies $\mathbf{KT} \vdash \psi$, which follows from $\mathbf{KT} \vdash \mathbf{D}$, as shown in [Proposição 51.28\(6\)](#). To show that the inclusion is proper, by Soundness ([Teorema 51.31](#)), it suffices to exhibit a model of $\mathbf{KD}$ where T, O.s., $\Box p \rightarrow p$, fails (an easy task left as an exercise), for then by Soundness $\mathbf{KD} \not\vdash \Box p \rightarrow p$. $\square$

Proposição 51.33. $\mathbf{KB} \neq \mathbf{K4}$.

Prova. We construct a symmetric model where some instance of 4 fails; since obviously the instance is [derivável](#) for $\mathbf{K4}$ but not in $\mathbf{KB}$, it will follow $\mathbf{K4} \not\subseteq \mathbf{KB}$. Consider the symmetric model $\mathfrak{M}$ of [Figura 51.1](#). Since the model is symmetric, K and B are true in $\mathfrak{M}$ (by [Proposição 49.18](#) and [Teorema 50.1](#), respectively). However, $\mathfrak{M}, w_1 \not\models \Box p \rightarrow \Box \Box p$. $\square$

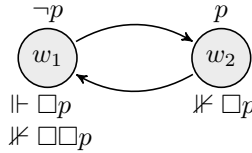
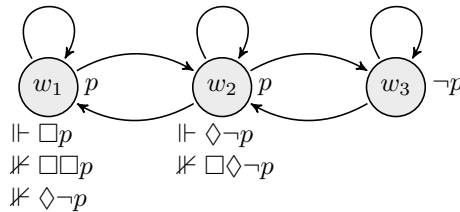


Figura 51.1: A symmetric model falsifying an instance of 4.

Teorema 51.34. $\mathbf{KTB} \not\vdash 4$ and $\mathbf{KTB} \not\vdash 5$.

nml:prf:dis:
fig:KTBnot45
thm:KTBnot45

Prova. By [Teorema 50.1](#) we know that all instances of T and B are true in every reflexive symmetric model (respectively). So by soundness, it suffices to find a reflexive symmetric model containing a world at which some instance of 4 fails, and similarly for 5. We use the same model for both claims. Consider the symmetric, reflexive model in [Figura 51.2](#). Then $\mathfrak{M}, w_1 \not\models \Box p \rightarrow \Box \Box p$, so 4 fails at w_1 . Similarly, $\mathfrak{M}, w_2 \not\models \Diamond \neg p \rightarrow \Box \Diamond \neg p$, so the instance of 5 with $\varphi = \neg p$ fails at w_2 . $\square$


 Figura 51.2: The model for [Teorema 51.34](#).

Teorema 51.35. $\mathbf{KD5} \neq \mathbf{KT4} = \mathbf{S4}$.

nml:prf:dis:
fig:KTBnot45
nml:prf:dis:
thm:KD5not4

51.11. DERIVABILIDADE FROM A SET OF FÓRMULAS

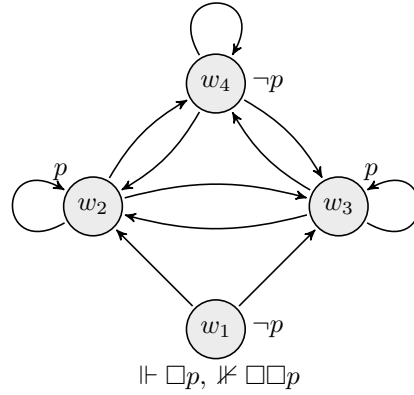


Figura 51.3: The model for Teorema 51.35.

nml:prf:dis:
fig:KD5not4

Prova. By Teorema 50.1 we know that all instances of D and 5 are true in all serial euclidean models. So it suffices to find a serial euclidean model containing a world at which some instance of 4 fails. Consider the model of Figura 51.3, and notice that $\mathfrak{M}, w_1 \nVdash \Box p \rightarrow \Box \Box p$. $\square$

Problema 51.7. Give an alternative proof of Teorema 51.35 using a model with 3 worlds.

Problema 51.8. Provide a single reflexive transitive model showing that both $\mathbf{KT4} \nVdash \mathbf{B}$ and $\mathbf{KT4} \nVdash 5$.

-NoValue-

content/normal-modal-logic/axioms-systems/provability-from-set.tex

51.11 Derivabilidade from a Set of Fórmulas

nml:prf:prg:
sec In Seção 51.8 we defined a notion of provability of a fórmula in a system Σ . We now extend this notion to provability in Σ from fórmulas in a set Γ .

nml:prf:prg:
defn:Gamma proves **Definição 51.36.** A fórmula φ is derivável in a system Σ from a set of fórmulas Γ , written $\Gamma \vdash_{\Sigma} \varphi$ if and only if there are $\psi_1, \dots, \psi_n \in \Gamma$ such that $\Sigma \vdash \psi_1 \rightarrow (\psi_2 \rightarrow \dots (\psi_n \rightarrow \varphi) \dots)$.

-NoValue-

content/normal-modal-logic/axioms-systems/provability-properties.tex

51.12 Properties of Derivabilidade

Proposição 51.37. *Let Σ be a modal system and Γ a set of modal fórmulas. The following properties hold:*

nml:prf:prp:
sec
nml:prf:prp:
prop:derivabilityfacts

1. Monotonicity: *If $\Gamma \vdash_{\Sigma} \varphi$ and $\Gamma \subseteq \Delta$ then $\Delta \vdash_{\Sigma} \varphi$;*
2. Reflexivity: *If $\varphi \in \Gamma$ then $\Gamma \vdash_{\Sigma} \varphi$;*
3. Cut: *If $\Gamma \vdash_{\Sigma} \varphi$ and $\Delta \cup \{\varphi\} \vdash_{\Sigma} \psi$ then $\Gamma \cup \Delta \vdash_{\Sigma} \psi$;*
4. Deduction theorem: *$\Gamma \cup \{\psi\} \vdash_{\Sigma} \varphi$ if and only if $\Gamma \vdash_{\Sigma} \psi \rightarrow \varphi$;*
5. *$\Gamma \vdash_{\Sigma} \varphi_1$ and ... and $\Gamma \vdash_{\Sigma} \varphi_n$ and $\varphi_1 \rightarrow (\varphi_2 \rightarrow \dots (\varphi_n \rightarrow \psi) \dots)$ is a tautological instance, then $\Gamma \vdash_{\Sigma} \psi$.*

nml:prf:prp:
prop:derivabilityfacts-monotonicity
nml:prf:prp:
prop:derivabilityfacts-reflexivity
nml:prf:prp:
prop:derivabilityfacts-cut
nml:prf:prp:
prop:derivabilityfacts-deduction
nml:prf:prp:
prop:derivabilityfacts-ruleT

The proof is an easy exercise. Part (5) of **Proposição 51.37** gives us that, for instance, if $\Gamma \vdash_{\Sigma} \varphi \vee \psi$ and $\Gamma \vdash_{\Sigma} \neg\varphi$, then $\Gamma \vdash_{\Sigma} \psi$. Also, in what follows, we write $\Gamma, \varphi \vdash_{\Sigma} \psi$ instead of $\Gamma \cup \{\varphi\} \vdash_{\Sigma} \psi$.

Definição 51.38. A set Γ is *deductively closed* relatively to a system Σ if and only if $\Gamma \vdash_{\Sigma} \varphi$ implies $\varphi \in \Gamma$.

-NoValue-

content/normal-modal-logic/axioms-systems/consistency.tex

51.13 Consistency

Consistency is an important property of sets of fórmulas. A set of fórmulas is inconsistent if a contradiction, such as $\perp$, is derivável from it; and otherwise consistent. If a set is inconsistent, its fórmulas cannot all be true in a model at a world. For the completeness theorem we prove the converse: every consistent set is true at a world in a model, namely in the “canonical model.”

nml:prf:con:
sec

Definição 51.39. A set Γ is *consistent* relatively to a system Σ or, as we will say, Σ -consistent, if and only if $\Gamma \not\vdash_{\Sigma} \perp$.

So for instance, the set $\{\Box(p \rightarrow q), \Box p, \neg\Box q\}$ is consistent relatively to propositional logic, but not **K**-consistent. Similarly, the set $\{\Diamond p, \Box\Diamond p \rightarrow q, \neg q\}$ is not **K5**-consistent.

Proposição 51.40. *Let Γ be a set of fórmulas. Then:*

nml:prf:con:
prop:consistencyfacts

1. *Γ is Σ -consistent if and only if there is some fórmula φ such that $\Gamma \not\vdash_{\Sigma} \varphi$.*
2. *$\Gamma \vdash_{\Sigma} \varphi$ if and only if $\Gamma \cup \{\neg\varphi\}$ is not Σ -consistent.*

nml:prf:con:
prop:consistencyfacts-b

3. If Γ is Σ -consistent, then for any *fórmula* φ , either $\Gamma \cup \{\varphi\}$ is Σ -consistent or $\Gamma \cup \{\neg\varphi\}$ is Σ -consistent.

Prova. These facts follow easily using classical propositional logic. We give the argument for (3). Proceed contrapositively and suppose neither $\Gamma \cup \{\varphi\}$ nor $\Gamma \cup \{\neg\varphi\}$ is Σ -consistent. Then by (2), both $\Gamma, \varphi \vdash_{\Sigma} \perp$ and $\Gamma, \neg\varphi \vdash_{\Sigma} \perp$. By the deduction theorem $\Gamma \vdash_{\Sigma} \varphi \rightarrow \perp$ and $\Gamma \vdash_{\Sigma} \neg\varphi \rightarrow \perp$. But $(\varphi \rightarrow \perp) \rightarrow ((\neg\varphi \rightarrow \perp) \rightarrow \perp)$ is a tautological instance, hence by **Proposição 51.37(5)**, $\Gamma \vdash_{\Sigma} \perp$. $\square$

Capítulo 52

Completeness and Canonical Models

-NoValue-

content/normal-modal-logic/completeness/introduction.tex

52.1 Introduction

nml:com:int:
sec

If Σ is a modal system, then the soundness theorem establishes that if $\Sigma \vdash \varphi$, then φ is valid in any class $\mathcal{C}$ of models in which all instances of all *fórmulas* in Σ are valid. In particular that means that if $\mathbf{K} \vdash \varphi$ then φ is true in all models; if $\mathbf{KT} \vdash \varphi$ then φ is true in all reflexive models; if $\mathbf{KD} \vdash \varphi$ then φ is true in all serial models, etc.

Completeness is the converse of soundness: that $\mathbf{K}$ is complete means that if a *fórmula* φ is valid, $\vdash \varphi$, for instance. Proving completeness is a lot harder to do than proving soundness. It is useful, first, to consider the contrapositive: $\mathbf{K}$ is complete iff whenever $\not\vdash \varphi$, there is a countermodel, O.s., a model $\mathfrak{M}$ such that $\mathfrak{M} \not\models \varphi$. Equivalently (negating φ), we could prove that whenever $\not\vdash \neg\varphi$, there is a model of φ . In the construction of such a model, we can use information contained in φ . When we find models for specific *fórmulas* we often do the same: p.ex., if we want to find a countermodel to $p \rightarrow \Box q$, we know that it has to contain a world where p is true and $\Box q$ is false. And a world where $\Box q$ is false means there has to be a world accessible from it where

q is false. And that's all we need to know: which worlds make the **variáveis proposicionais** true, and which worlds are accessible from which worlds.

In the case of proving completeness, however, we don't have a specific **fórmula** φ for which we are constructing a model. We want to establish that a model exists for every φ such that $\not\vdash_{\Sigma} \neg\varphi$. This is a minimal requirement, since if $\vdash_{\Sigma} \neg\varphi$, by soundness, there is no model for φ (in which Σ is true). Now note that $\not\vdash_{\Sigma} \neg\varphi$ iff φ is Σ -consistent. (Recall that $\Sigma \not\vdash_{\Sigma} \neg\varphi$ and $\varphi \not\vdash_{\Sigma} \perp$ are equivalent.) So our task is to construct a model for every Σ -consistent **fórmula**.

The trick we'll use is to find a Σ -consistent set of **fórmulas** that contains φ , but also other formulas which tell us what the world that makes φ true has to look like. Such sets are *complete* Σ -consistent sets. It's not enough to construct a model with a single world to make φ true, it will have to contain multiple worlds and an accessibility relation. The complete Σ -consistent set containing φ will also contain other **fórmulas** of the form $\Box\psi$ and $\Diamond\chi$. In all accessible worlds, ψ has to be true; in at least one, χ has to be true. In order to accomplish this, we'll simply take *all* possible complete Σ -consistent sets as the basis for the set of worlds. A tricky part will be to figure out when a complete Σ -consistent set should count as being accessible from another in our model.

We'll show that in the model so defined, φ is true at a world—which is also a complete Σ -consistent set—iff φ is **um membro** of that set. If φ is Σ -consistent, it will be **um membro** of at least one complete Σ -consistent set (a fact we'll prove), and so there will be a world where φ is true. So we will have a single model where every Σ -consistent **fórmula** φ is true at some world. This single model is the *canonical* model for Σ .

-NoValue-

`content/normal-modal-logic/completeness/complete-consistent-sets.tex`

52.2 Complete Σ -Consistent Sets

Suppose Σ is a set of modal **fórmulas**—think of them as the axioms or defining principles of a normal modal logic. A set Γ is Σ -consistent iff $\Gamma \not\vdash_{\Sigma} \perp$, O.s., if there is no **derivação** of $\varphi_1 \rightarrow (\varphi_2 \rightarrow \dots (\varphi_n \rightarrow \perp) \dots)$ from Σ , where each $\varphi_i \in \Gamma$. We will construct a “canonical” model in which each world is taken to be a special kind of Σ -consistent set: one which is not just Σ -consistent, but maximally so, in the sense that it settles the truth value of every modal **fórmula**: for every φ , either $\varphi \in \Gamma$ or $\neg\varphi \in \Gamma$:

[nml:com:ccs:sec](#)

Definição 52.1. A set Γ is *complete Σ -consistent* if and only if it is Σ -consistent and for every φ , either $\varphi \in \Gamma$ or $\neg\varphi \in \Gamma$.

Complete Σ -consistent sets Γ have a number of useful properties. For one, they are deductively closed, O.s., if $\Gamma \vdash_{\Sigma} \varphi$ then $\varphi \in \Gamma$. This means in particular that every instance of a **fórmula** $\varphi \in \Sigma$ is also $\in \Gamma$. Moreover,

52.2. COMPLETE Σ -CONSISTENT SETS

membership in Γ mirrors the truth conditions for the propositional connectives. This will be important when we define the “canonical model.”

Proposição 52.2. *Suppose Γ is complete Σ -consistent. Then:*

*nml:com:ccs:
prop:ccs-properties*

1. Γ is deductively closed in Σ .

*nml:com:ccs:
prop:ccs-closed*

2. $\Sigma \subseteq \Gamma$.

*nml:com:ccs:
prop:ccs-sigma*

3. $\perp \notin \Gamma$

*nml:com:ccs:
prop:ccs-lfalse*

4. $\top \in \Gamma$

*nml:com:ccs:
prop:ccs-ltrue*

5. $\neg\varphi \in \Gamma$ if and only if $\varphi \notin \Gamma$.

*nml:com:ccs:
prop:ccs-lnot*

6. $\varphi \wedge \psi \in \Gamma$ iff $\varphi \in \Gamma$ and $\psi \in \Gamma$

*nml:com:ccs:
prop:ccs-land*

7. $\varphi \vee \psi \in \Gamma$ iff $\varphi \in \Gamma$ or $\psi \in \Gamma$

*nml:com:ccs:
prop:ccs-lor*

8. $\varphi \rightarrow \psi \in \Gamma$ iff $\varphi \notin \Gamma$ or $\psi \in \Gamma$

*nml:com:ccs:
prop:ccs-lif*

9. $\varphi \leftrightarrow \psi \in \Gamma$ iff either $\varphi \in \Gamma$ and $\psi \in \Gamma$, or $\varphi \notin \Gamma$ and $\psi \notin \Gamma$

*nml:com:ccs:
prop:ccs-liff*

Prova. 1. Suppose $\Gamma \vdash_{\Sigma} \varphi$ but $\varphi \notin \Gamma$. Then since Γ is complete Σ -consistent, $\neg\varphi \in \Gamma$. This would make Γ inconsistent, since $\varphi, \neg\varphi \vdash_{\Sigma} \perp$.

2. If $\varphi \in \Sigma$ then $\Gamma \vdash_{\Sigma} \varphi$, and $\varphi \in \Gamma$ by deductive closure, O.s., case (1).

3. If $\perp \in \Gamma$, then $\Gamma \vdash_{\Sigma} \perp$, so Γ would be Σ -inconsistent.

4. $\Gamma \vdash_{\Sigma} \top$, so $\top \in \Gamma$ by deductive closure, O.s., case (1).

5. If $\neg\varphi \in \Gamma$, then by consistency $\varphi \notin \Gamma$; and if $\varphi \notin \Gamma$ then $\varphi \in \Gamma$ since Γ is complete Σ -consistent.

6. Suppose $\varphi \wedge \psi \in \Gamma$. Since $(\varphi \wedge \psi) \rightarrow \varphi$ is a tautological instance, $\varphi \in \Gamma$ by deductive closure, O.s., case (1). Similarly for $\psi \in \Gamma$. On the other hand, suppose both $\varphi \in \Gamma$ and $\psi \in \Gamma$. Then deductive closure implies $(\varphi \wedge \psi) \in \Gamma$, since $\varphi \rightarrow (\psi \rightarrow (\varphi \wedge \psi))$ is a tautological instance.

7. Suppose $\varphi \vee \psi \in \Gamma$, and $\varphi \notin \Gamma$ and $\psi \notin \Gamma$. Since Γ is complete Σ -consistent, $\neg\varphi \in \Gamma$ and $\neg\psi \in \Gamma$. Then $\neg(\varphi \vee \psi) \in \Gamma$ since $\neg\varphi \rightarrow (\neg\psi \rightarrow \neg(\varphi \vee \psi))$ is a tautological instance. This would mean that Γ is Σ -inconsistent, a contradiction.

8. Suppose $\varphi \rightarrow \psi \in \Gamma$ and $\varphi \in \Gamma$; then $\Gamma \vdash_{\Sigma} \psi$, whence $\psi \in \Gamma$ by deductive closure. Conversely, if $\varphi \rightarrow \psi \notin \Gamma$ then since Γ is complete Σ -consistent, $\neg(\varphi \rightarrow \psi) \in \Gamma$. Since $\neg(\varphi \rightarrow \psi) \rightarrow \varphi$ is a tautological instance, $\varphi \in \Gamma$ by deductive closure. Since $\neg(\varphi \rightarrow \psi) \rightarrow \neg\psi$ is a tautological instance, $\neg\psi \in \Gamma$. Then $\psi \notin \Gamma$ since Γ is Σ -consistent.

9. Suppose $\varphi \leftrightarrow \psi \in \Gamma$. If $\varphi \in \Gamma$, then $\psi \in \Gamma$, since $(\varphi \leftrightarrow \psi) \rightarrow (\varphi \rightarrow \psi)$ is a tautological instance. Similarly, if $\psi \in \Gamma$, then $\varphi \in \Gamma$. So either both $\varphi \in \Gamma$ and $\psi \in \Gamma$, or neither $\varphi \in \Gamma$ nor $\psi \in \Gamma$.

Conversely, suppose $\varphi \rightarrow \psi \notin \Gamma$. Since Γ is complete Σ -consistent, $\neg(\varphi \leftrightarrow \psi) \in \Gamma$. Since $\neg(\varphi \leftrightarrow \psi) \rightarrow (\varphi \rightarrow \neg\psi)$ is a tautological instance, if $\varphi \in \Gamma$ then $\neg\psi \in \Gamma$, and since Γ is Σ -consistent, $\psi \notin \Gamma$. Similarly, if $\psi \in \Gamma$ then $\varphi \notin \Gamma$. So neither $\varphi \in \Gamma$ and $\psi \in \Gamma$, nor $\varphi \notin \Gamma$ and $\psi \notin \Gamma$. $\square$

Problema 52.1. Complete the proof of [Proposição 52.2](#).

-NoValue-

[content/normal-modal-logic/completeness/lindenbaums-lemma.tex](#)

52.3 Lindenbaum's Lemma

Lindenbaum's Lemma establishes that every Σ -consistent set of [fórmulas](#) is contained in at least one *complete* Σ -consistent set. Our construction of the canonical model will show that for each complete Σ -consistent set Δ , there is a world in the canonical model where all and only the [fórmulas](#) in Δ are true. So Lindenbaum's Lemma guarantees that every Σ -consistent set is true at some world in the canonical model.

[nml:com:lin:sec](#)

Teorema 52.3 (Lindenbaum's Lemma). *If Γ is Σ -consistent then there is a complete Σ -consistent set Δ extending Γ .*

[nml:com:lin:thm:lindenbaum](#)

Prova. Let $\varphi_0, \varphi_1, \dots$ be an exhaustive listing of all formulas of the language (repetitions are allowed). For instance, start by listing p_0 , and at each stage $n \geq 1$ list the finitely many formulas of length n using only variables among $p_0, \dots, p_n$. We define sets of [fórmulas](#) Δ_n by induction on n , and we then set $\Delta = \bigcup_n \Delta_n$. We first put $\Delta_0 = \Gamma$. Supposing that Δ_n has been defined, we define Δ_{n+1} by:

$$\Delta_{n+1} = \begin{cases} \Delta_n \cup \{\varphi_n\}, & \text{if } \Delta_n \cup \{\varphi_n\} \text{ is } \Sigma\text{-consistent;} \\ \Delta_n \cup \{\neg\varphi_n\}, & \text{otherwise.} \end{cases}$$

Now let $\Delta = \bigcup_{n=0}^{\infty} \Delta_n$.

We have to show that this definition actually yields a set Δ with the required properties, O.s., $\Gamma \subseteq \Delta$ and Δ is complete Σ -consistent.

It's obvious that $\Gamma \subseteq \Delta$, since $\Delta_0 \subseteq \Delta$ by construction, and $\Delta_0 = \Gamma$. In fact, $\Delta_n \subseteq \Delta$ for all n , since Δ is the union of all Δ_n . (Since in each step of the construction, we add a [fórmula](#) to the set already constructed, $\Delta_n \subseteq \Delta_{n+1}$, so since $\subseteq$ is transitive, $\Delta_n \subseteq \Delta_m$ whenever $n \leq m$.) At each stage of the construction, we either add φ_n or $\neg\varphi_n$, and every [fórmula](#) appears (at least once) in the list of all φ_n . So, for every φ either $\varphi \in \Delta$ or $\neg\varphi \in \Delta$, so Δ is complete by definition.

52.4. MODALITIES AND COMPLETE CONSISTENT SETS

Finally, we have to show, that Δ is Σ -consistent. To do this, we show that (a) if Δ were Σ -inconsistent, then some Δ_n would be Σ -inconsistent, and (b) all Δ_n are Σ -consistent.

So suppose Δ were Σ -inconsistent. Then $\Delta \vdash_{\Sigma} \perp$, O.s., there are $\varphi_1, \dots, \varphi_k \in \Delta$ such that $\Sigma \vdash \varphi_1 \rightarrow (\varphi_2 \rightarrow \dots (\varphi_k \rightarrow \perp) \dots)$. Since $\Delta = \bigcup_{n=0}^{\infty} \Delta_n$, each $\varphi_i \in \Delta_{n_i}$ for some n_i . Let n be the largest of these. Since $n_i \leq n$, $\Delta_{n_i} \subseteq \Delta_n$. So, all φ_i are in some Δ_n . This would mean $\Delta_n \vdash_{\Sigma} \perp$, O.s., Δ_n is Σ -inconsistent.

To show that each Δ_n is Σ -consistent, we use a simple induction on n . $\Delta_0 = \Gamma$, and we assumed Γ was Σ -consistent. So the claim holds for $n = 0$. Now suppose it holds for n , O.s., Δ_n is Σ -consistent. Δ_{n+1} is either $\Delta_n \cup \{\varphi_n\}$ if that is Σ -consistent, otherwise it is $\Delta_n \cup \{\neg\varphi_n\}$. In the first case, Δ_{n+1} is clearly Σ -consistent. However, by [Proposição 51.40\(3\)](#), either $\Delta_n \cup \{\varphi_n\}$ or $\Delta_n \cup \{\neg\varphi_n\}$ is consistent, so Δ_{n+1} is consistent in the other case as well. $\square$

Corolário 52.4. $\Gamma \vdash_{\Sigma} \varphi$ if and only if $\varphi \in \Delta$ for each complete Σ -consistent set Δ extending Γ (including when $\Gamma = \emptyset$, in which case we get another characterization of the modal system Σ .)

Prova. Suppose $\Gamma \vdash_{\Sigma} \varphi$, and let Δ be any complete Σ -consistent set extending Γ . If $\varphi \notin \Delta$ then by maximality $\neg\varphi \in \Delta$ and so $\Delta \vdash_{\Sigma} \varphi$ (by monotonicity) and $\Delta \vdash_{\Sigma} \neg\varphi$ (by reflexivity), and so Δ is inconsistent. Conversely if $\Gamma \not\vdash_{\Sigma} \varphi$, then $\Gamma \cup \{\neg\varphi\}$ is Σ -consistent, and by Lindenbaum's Lemma there is a complete consistent set Δ extending $\Gamma \cup \{\neg\varphi\}$. By consistency, $\varphi \notin \Delta$. $\square$

-NoValue-

content/normal-modal-logic/completeness/modalities-ccs.tex

52.4 Modalities and Complete Consistent Sets

When we construct a model $\mathfrak{M}^{\Sigma}$ whose set of worlds is given by the complete Σ -consistent sets Δ in some normal modal logic Σ , we will also need to define an accessibility relation R^{Σ} between such “worlds.” We want it to be the case that the accessibility relation (and the assignment V^{Σ}) are defined in such a way that $\mathfrak{M}^{\Sigma}, \Delta \Vdash \varphi$ iff $\varphi \in \Delta$. How should we do this?

Once the accessibility relation is defined, the definition of truth at a world ensures that $\mathfrak{M}^{\Sigma}, \Delta \Vdash \Box\varphi$ iff $\mathfrak{M}^{\Sigma}, \Delta' \Vdash \varphi$ for all Δ' such that $R^{\Sigma}\Delta\Delta'$. The proof that $\mathfrak{M}^{\Sigma}, \Delta \Vdash \varphi$ iff $\varphi \in \Delta$ requires that this is true in particular for [fórmulas](#) starting with a modal operator, O.s., $\mathfrak{M}^{\Sigma}, \Delta \Vdash \Box\varphi$ iff $\Box\varphi \in \Delta$. Combining this requirement with the definition of truth at a world for $\Box\varphi$ yields:

$$\Box\varphi \in \Delta \text{ iff } \varphi \in \Delta' \text{ for all } \Delta' \text{ with } R^{\Sigma}\Delta\Delta'$$

Consider the left-to-right direction: it says that if $\Box\varphi \in \Delta$, then $\varphi \in \Delta'$ for any φ and any Δ' with $R^{\Sigma}\Delta\Delta'$. If we stipulate that $R^{\Sigma}\Delta\Delta'$ iff $\varphi \in \Delta'$ for all

$\Box\varphi \in \Delta$, then this holds. We can write the condition on the right of the “iff” more compactly as: $\{\varphi : \Box\varphi \in \Delta\} \subseteq \Delta'$.

So the question is: does this definition of R^Σ in fact guarantee that $\Box\varphi \in \Delta$ iff $\mathfrak{M}^\Sigma, \Delta \Vdash \Box\varphi$? Does it also guarantee that $\Diamond\varphi \in \Delta$ iff $\mathfrak{M}^\Sigma, \Delta \Vdash \Diamond\varphi$? The next few results will establish this.

Definição 52.5. If Γ is a set of *fórmulas*, let

$$\begin{aligned}\Box\Gamma &= \{\Box\psi : \psi \in \Gamma\} \\ \Diamond\Gamma &= \{\Diamond\psi : \psi \in \Gamma\}\end{aligned}$$

and

$$\begin{aligned}\Box^{-1}\Gamma &= \{\psi : \Box\psi \in \Gamma\} \\ \Diamond^{-1}\Gamma &= \{\psi : \Diamond\psi \in \Gamma\}\end{aligned}$$

In other words, $\Box\Gamma$ is Γ with $\Box$ in front of every *fórmula* in Γ ; $\Box^{-1}\Gamma$ is all the $\Box$ 'ed *fórmulas* of Γ with the initial $\Box$'s removed. This definition is not terribly important on its own, but will simplify the notation considerably.

Note that $\Box\Box^{-1}\Gamma \subseteq \Gamma$:

$$\Box\Box^{-1}\Gamma = \{\Box\psi : \Box\psi \in \Gamma\}$$

O.s., it's just the set of all those *fórmulas* of Γ that start with $\Box$.

Lema 52.6. *If $\Gamma \vdash_\Sigma \varphi$ then $\Box\Gamma \vdash_\Sigma \Box\varphi$.*

*nml:com:mod:
lem:box1*

Prova. If $\Gamma \vdash_\Sigma \varphi$ then there are $\psi_1, \dots, \psi_k \in \Gamma$ such that $\Sigma \vdash \psi_1 \rightarrow (\psi_2 \rightarrow \dots (\psi_n \rightarrow \varphi) \dots)$. Since Σ is normal, by rule RK, $\Sigma \vdash \Box\psi_1 \rightarrow (\Box\psi_2 \rightarrow \dots (\Box\psi_n \rightarrow \Box\varphi) \dots)$, where obviously $\Box\psi_1, \dots, \Box\psi_k \in \Box\Gamma$. Hence, by definition, $\Box\Gamma \vdash_\Sigma \Box\varphi$. $\square$

Lema 52.7. *If $\Box^{-1}\Gamma \vdash_\Sigma \varphi$ then $\Gamma \vdash_\Sigma \Box\varphi$.*

*nml:com:mod:
lem:box2*

Prova. Suppose $\Box^{-1}\Gamma \vdash_\Sigma \varphi$; then by **Lema 52.6**, $\Box\Box^{-1}\Gamma \vdash \Box\varphi$. But since $\Box\Box^{-1}\Gamma \subseteq \Gamma$, also $\Gamma \vdash_\Sigma \Box\varphi$ by monotonicity. $\square$

Proposição 52.8. *If Γ is complete Σ -consistent, then $\Box\varphi \in \Gamma$ if and only if for every complete Σ -consistent Δ such that $\Box^{-1}\Gamma \subseteq \Delta$, it holds that $\varphi \in \Delta$.*

*nml:com:mod:
prop:box*

Prova. Suppose Γ is complete Σ -consistent. The “only if” direction is easy: Suppose $\Box\varphi \in \Gamma$ and that $\Box^{-1}\Gamma \subseteq \Delta$. Since $\Box\varphi \in \Gamma$, $\varphi \in \Box^{-1}\Gamma \subseteq \Delta$, so $\varphi \in \Delta$.

For the “if” direction, we prove the contrapositive: Suppose $\Box\varphi \notin \Gamma$. Since Γ is complete Σ -consistent, it is deductively closed, and hence $\Gamma \not\vdash_\Sigma \Box\varphi$. By **Lema 52.7**, $\Box^{-1}\Gamma \not\vdash_\Sigma \varphi$. By **Proposição 51.40(2)**, $\Box^{-1}\Gamma \cup \{\neg\varphi\}$ is Σ -consistent. By Lindenbaum's Lemma, there is a complete Σ -consistent set Δ such that $\Box^{-1}\Gamma \cup \{\neg\varphi\} \subseteq \Delta$. By consistency, $\varphi \notin \Delta$. $\square$

52.5. CANONICAL MODELS

nml:com:mod:
lem:box-iff-diamond

Lema 52.9. *Suppose Γ and Δ are complete Σ -consistent. Then $\Box^{-1}\Gamma \subseteq \Delta$ if and only if $\Diamond\Delta \subseteq \Gamma$.*

Prova. “Only if” direction: Assume $\Box^{-1}\Gamma \subseteq \Delta$ and suppose $\Diamond\varphi \in \Diamond\Delta$ (O.s., $\varphi \in \Delta$). In order to show $\Diamond\varphi \in \Gamma$, it suffices to show $\Box\neg\varphi \notin \Gamma$, for then by maximality, $\neg\Box\neg\varphi \in \Gamma$. Now, if $\Box\neg\varphi \in \Gamma$ then by hypothesis $\neg\varphi \in \Delta$, against the consistency of Δ (since $\varphi \in \Delta$). Hence $\Box\neg\varphi \notin \Gamma$, as required.

“If” direction: Assume $\Diamond\Delta \subseteq \Gamma$. We argue contrapositively: suppose $\varphi \notin \Delta$ in order to show $\Box\varphi \notin \Gamma$. If $\varphi \notin \Delta$ then by maximality $\neg\varphi \in \Delta$ and so by hypothesis $\Diamond\neg\varphi \in \Gamma$. But in a normal modal logic $\Diamond\neg\varphi$ is equivalent to $\neg\Box\varphi$, and if the latter is in Γ , by consistency $\Box\varphi \notin \Gamma$, as required. $\square$

nml:com:mod:
prop:diamond

Proposição 52.10. *If Γ is complete Σ -consistent, then $\Diamond\varphi \in \Gamma$ if and only if for some complete Σ -consistent Δ such that $\Diamond\Delta \subseteq \Gamma$, it holds that $\varphi \in \Delta$.*

Prova. Suppose Γ is complete Σ -consistent. $\Diamond\varphi \in \Gamma$ iff $\neg\Box\neg\varphi \in \Gamma$ by DUAL and closure. $\neg\Box\neg\varphi \in \Gamma$ iff $\Box\neg\varphi \notin \Gamma$ by Proposição 52.2(5) since Γ is complete Σ -consistent. By Proposição 52.8, $\Box\neg\varphi \notin \Gamma$ iff, for some complete Σ -consistent Δ with $\Box^{-1}\Gamma \subseteq \Delta$, $\neg\varphi \notin \Delta$. Now consider any such Δ . By Lema 52.9, $\Box^{-1}\Gamma \subseteq \Delta$ iff $\Diamond\Delta \subseteq \Gamma$. Also, $\neg\varphi \notin \Delta$ iff $\varphi \in \Delta$ by Proposição 52.2(5). So $\Diamond\varphi \in \Gamma$ iff, for some complete Σ -consistent Δ with $\Diamond\Delta \subseteq \Gamma$, $\varphi \in \Delta$. $\square$

Problema 52.2. Show that if Γ is complete Σ -consistent, then $\Diamond\varphi \in \Gamma$ if and only if there is a complete Σ -consistent Δ such that $\Box^{-1}\Gamma \subseteq \Delta$ and $\varphi \in \Delta$. Do this without using Lema 52.9.

-NoValue-

content/normal-modal-logic/completeness/canonical-models.tex

52.5 Canonical Models

nml:com:cmd:
sec

The *canonical model* for a modal system Σ is a specific model $\mathfrak{M}^\Sigma$ in which the worlds are all complete Σ -consistent sets. Its accessibility relation R^Σ and valuation V^Σ are defined so as to guarantee that the fórmulas true at a world Δ are exactly the fórmulas making up Δ .

Definição 52.11. Let Σ be a normal modal logic. The *canonical model* for Σ is $\mathfrak{M}^\Sigma = \langle W^\Sigma, R^\Sigma, V^\Sigma \rangle$, where:

1. $W^\Sigma = \{\Delta : \Delta \text{ is complete } \Sigma\text{-consistent}\}$.
2. $R^\Sigma\Delta\Delta'$ holds if and only if $\Box^{-1}\Delta \subseteq \Delta'$.
3. $V^\Sigma(p) = \{\Delta : p \in \Delta\}$.

-NoValue-

content/normal-modal-logic/completeness/truth-lemma.tex

52.6 The Truth Lemma

The canonical model $\mathfrak{M}^\Sigma$ is defined in such a way that $\mathfrak{M}^\Sigma, \Delta \Vdash \varphi$ iff $\varphi \in \Delta$. For **variáveis proposicionais**, the definition of V^Σ yields this directly. We have to verify that the equivalence holds for all **fórmulas**, however. We do this by induction. The inductive step involves proving the equivalence for **fórmulas** involving propositional operators (where we have to use **Proposição 52.2**) and the modal operators (where we invoke the results of **Seção 52.4**).

nml:com:tru:
sec

Proposição 52.12 (Truth Lemma). *For every **fórmula** φ , $\mathfrak{M}^\Sigma, \Delta \Vdash \varphi$ if and only if $\varphi \in \Delta$.*

nml:com:tru:
prop:truthlemma

Prova. By induction on φ .

1. $\varphi \equiv \perp$: $\mathfrak{M}^\Sigma, \Delta \nVdash \perp$ by **Definição 49.6**, and $\perp \notin \Delta$ by **Proposição 52.2(3)**.
2. $\varphi \equiv \top$: $\mathfrak{M}^\Sigma, \Delta \Vdash \top$ by **Definição 49.6**, and $\top \in \Delta$ by **Proposição 52.2(4)**.
3. $\varphi \equiv p$: $\mathfrak{M}^\Sigma, \Delta \Vdash p$ iff $\Delta \in V^\Sigma(p)$ by **Definição 49.6**. Also, $\Delta \in V^\Sigma(p)$ iff $p \in \Delta$ by definition of V^Σ .
4. $\varphi \equiv \neg\psi$: $\mathfrak{M}^\Sigma, \Delta \Vdash \neg\psi$ iff $\mathfrak{M}^\Sigma, \Delta \nVdash \psi$ (**Definição 49.6**) iff $\psi \notin \Delta$ (by inductive hypothesis) iff $\neg\psi \in \Delta$ (by **Proposição 52.2(5)**).
5. $\varphi \equiv \psi \wedge \chi$: $\mathfrak{M}^\Sigma, \Delta \Vdash \psi \wedge \chi$ iff $\mathfrak{M}^\Sigma, \Delta \Vdash \psi$ and $\mathfrak{M}^\Sigma, \Delta \Vdash \chi$ (by **Definição 49.6**) iff $\psi \in \Delta$ and $\chi \in \Delta$ (by inductive hypothesis) iff $\psi \wedge \chi \in \Delta$ (by **Proposição 52.2(6)**).
6. $\varphi \equiv \psi \vee \chi$: $\mathfrak{M}^\Sigma, \Delta \Vdash \psi \vee \chi$ iff $\mathfrak{M}^\Sigma, \Delta \Vdash \psi$ or $\mathfrak{M}^\Sigma, \Delta \Vdash \chi$ (by **Definição 49.6**) iff $\psi \in \Delta$ or $\chi \in \Delta$ (by inductive hypothesis) iff $\psi \vee \chi \in \Delta$ (by **Proposição 52.2(7)**).
7. $\varphi \equiv \psi \rightarrow \chi$: $\mathfrak{M}^\Sigma, \Delta \Vdash \psi \rightarrow \chi$ iff $\mathfrak{M}^\Sigma, \Delta \nVdash \psi$ or $\mathfrak{M}^\Sigma, \Delta \Vdash \chi$ (by **Definição 49.6**) iff $\psi \notin \Delta$ or $\chi \in \Delta$ (by inductive hypothesis) iff $\psi \rightarrow \chi \in \Delta$ (by **Proposição 52.2(8)**).
8. $\varphi \equiv \psi \leftrightarrow \chi$: $\mathfrak{M}^\Sigma, \Delta \Vdash \psi \leftrightarrow \chi$ iff either $\mathfrak{M}^\Sigma, \Delta \Vdash \psi$ and $\mathfrak{M}^\Sigma, \Delta \Vdash \chi$ or $\mathfrak{M}^\Sigma, \Delta \nVdash \psi$ and $\mathfrak{M}^\Sigma, \Delta \nVdash \chi$ (by **Definição 49.6**) iff either $\psi \in \Delta$ and $\chi \in \Delta$ or $\psi \notin \Delta$ and $\chi \notin \Delta$ (by inductive hypothesis) iff $\psi \leftrightarrow \chi \in \Delta$ (by **Proposição 52.2(9)**).

9. $\varphi \equiv \Box\psi$: First suppose that $\mathfrak{M}^\Sigma, \Delta \Vdash \Box\psi$. By **Definição 49.6**, for every Δ' such that $R^\Sigma \Delta \Delta'$, $\mathfrak{M}^\Sigma, \Delta' \Vdash \psi$. By inductive hypothesis, for every Δ' such that $R^\Sigma \Delta \Delta'$, $\psi \in \Delta'$. By definition of R^Σ , for every Δ' such that $\Box^{-1}\Delta \subseteq \Delta'$, $\psi \in \Delta'$. By **Proposição 52.8**, $\Box\psi \in \Delta$.

Now assume $\Box\psi \in \Delta$. Let $\Delta' \in W^\Sigma$ be such that $R^\Sigma \Delta \Delta'$, O.s., $\Box^{-1}\Delta \subseteq \Delta'$. Since $\Box\psi \in \Delta$, $\psi \in \Box^{-1}\Delta$. Consequently, $\psi \in \Delta'$. By inductive hypothesis, $\mathfrak{M}^\Sigma, \Delta' \Vdash \psi$. Since Δ' is arbitrary with $R^\Sigma \Delta \Delta'$, for all $\Delta' \in W^\Sigma$ such that $R^\Sigma \Delta \Delta'$, $\mathfrak{M}^\Sigma, \Delta' \Vdash \psi$. By **Definição 49.6**, $\mathfrak{M}^\Sigma, \Delta \Vdash \Box\psi$.

52.7. DETERMINATION AND COMPLETENESS FOR K

10. $\varphi \equiv \Diamond\psi$: First suppose that $\mathfrak{M}^\Sigma, \Delta \Vdash \Diamond\psi$. By [Definição 49.6](#), for some Δ' such that $R^\Sigma \Delta \Delta'$, $\mathfrak{M}^\Sigma, \Delta' \Vdash \psi$. By inductive hypothesis, for some Δ' such that $R^\Sigma \Delta \Delta'$, $\psi \in \Delta'$. By definition of R^Σ , for some Δ' such that $\Box^{-1}\Delta \subseteq \Delta'$, $\psi \in \Delta'$. By [Proposição 52.10](#), for some Δ' such that $\Diamond\Delta' \subseteq \Delta$, $\psi \in \Delta'$. Since $\psi \in \Delta'$, $\Diamond\psi \in \Diamond\Delta'$, so $\Diamond\psi \in \Delta$.

Now assume $\Diamond\psi \in \Delta$. By [Proposição 52.10](#), there is a complete Σ -consistent $\Delta' \in W^\Sigma$ such that $\Diamond\Delta' \subseteq \Delta$ and $\psi \in \Delta'$. By [Lema 52.9](#), there is a $\Delta' \in W^\Sigma$ such that $\Box^{-1}\Delta \subseteq \Delta'$, and $\psi \in \Delta'$. By definition of R^Σ , $R^\Sigma \Delta \Delta'$, so there is a $\Delta' \in W^\Sigma$ such that $R^\Sigma \Delta \Delta'$ and $\psi \in \Delta'$. By [Definição 49.6](#), $\mathfrak{M}^\Sigma, \Delta \Vdash \Diamond\psi$. $\square$

Problema 52.3. Complete the proof of [Proposição 52.12](#).

-NoValue-

[content/normal-modal-logic/completeness/completeness-K.tex](#)

52.7 Determination and Completeness for K

[nml:com:cmk:](#)
[sec](#)

We are now prepared to use the canonical model to establish completeness. Completeness follows from the fact that the [fórmulas](#) true in the canonical model for Σ are exactly the Σ -[derivável](#) ones. Models with this property are said to *determine* Σ .

Definição 52.13. A model $\mathfrak{M}$ *determines* a normal modal logic Σ precisely when $\mathfrak{M} \Vdash \varphi$ if and only if $\Sigma \vdash \varphi$, for all [fórmulas](#) φ .

[nml:com:cmk:](#)
[thm:determination](#)

Teorema 52.14 (Determination). $\mathfrak{M}^\Sigma \Vdash \varphi$ if and only if $\Sigma \vdash \varphi$.

Prova. If $\mathfrak{M}^\Sigma \Vdash \varphi$, then for every complete Σ -consistent Δ , we have $\mathfrak{M}^\Sigma, \Delta \Vdash \varphi$. Hence, by the Truth Lemma, $\varphi \in \Delta$ for every complete Σ -consistent Δ , whence by [Corolário 52.4](#) (with $\Gamma = \emptyset$), $\Sigma \vdash \varphi$.

Conversely, if $\Sigma \vdash \varphi$ then by [Proposição 52.2\(1\)](#), every complete Σ -consistent Δ contains φ , and hence by the Truth Lemma, $\mathfrak{M}^\Sigma, \Delta \Vdash \varphi$ for every $\Delta \in W^\Sigma$, O.s., $\mathfrak{M}^\Sigma \Vdash \varphi$. $\square$

Since the canonical model for **K** determines **K**, we immediately have completeness of **K** as a corollary:

[nml:com:cmk:](#)
[cor:Kcomplete](#)

Corolário 52.15. *The basic modal logic **K** is complete with respect to the class of all models, O.s., if $\models \varphi$ then $\mathbf{K} \vdash \varphi$.*

Prova. Contrapositively, if $\mathbf{K} \not\vdash \varphi$ then by Determination $\mathfrak{M}^\mathbf{K} \not\vdash \varphi$ and hence φ is not valid. $\square$

For the general case of completeness of a system Σ with respect to a class of models, p.ex., of **KTB4** with respect to the class of reflexive, symmetric, transitive models, determination alone is not enough. We must also show that the canonical model for the system Σ is a member of the class, which does not follow obviously from the canonical model construction—nor is it always true!

-NoValue-

[content/normal-modal-logic/completeness/frame-completeness.tex](#)

52.8 Frame Completeness

The completeness theorem for **K** can be extended to other modal systems, once we show that the canonical model for a given logic has the corresponding frame property.

[nml:com:fra:](#)
[sec](#)

Teorema 52.16. *If a normal modal logic Σ contains one of the **fórmulas** on the left-hand side of [Tabela 52.1](#), then the canonical model for Σ has the corresponding property on the right-hand side.*

[nml:com:fra:](#)
[thm:completeframeprops](#)

<i>If Σ contains ...</i>	<i>... the canonical model for Σ is:</i>
D: $\Box\varphi \rightarrow \Diamond\varphi$	serial;
T: $\Box\varphi \rightarrow \varphi$	reflexive;
B: $\varphi \rightarrow \Box\Diamond\varphi$	symmetric;
4: $\Box\varphi \rightarrow \Box\Box\varphi$	transitive;
5: $\Diamond\varphi \rightarrow \Box\Diamond\varphi$	euclidean.

Tabela 52.1: Basic correspondence facts.

Prova. We take each of these up in turn.

[nml:com:fra:](#)
[tab:correspondencetable](#)

Suppose Σ contains D, and let $\Delta \in W^\Sigma$; we need to show that there is a Δ' such that $R^\Sigma \Delta \Delta'$. It suffices to show that $\Box^{-1}\Delta$ is Σ -consistent, for then by Lindenbaum's Lemma, there is a complete Σ -consistent set $\Delta' \supseteq \Box^{-1}\Delta$, and by definition of R^Σ we have $R^\Sigma \Delta \Delta'$. So, suppose for contradiction that $\Box^{-1}\Delta$ is *not* Σ -consistent, O.s., $\Box^{-1}\Delta \vdash_\Sigma \perp$. By [Lema 52.7](#), $\Delta \vdash_\Sigma \Box \perp$, and since Σ contains D, also $\Delta \vdash_\Sigma \Diamond \perp$. But Σ is normal, so $\Sigma \vdash \neg \Diamond \perp$ ([Proposição 51.7](#)), whence also $\Delta \vdash_\Sigma \neg \Diamond \perp$, against the consistency of Δ .

Now suppose Σ contains T, and let $\Delta \in W^\Sigma$. We want to show $R^\Sigma \Delta \Delta$, O.s., $\Box^{-1}\Delta \subseteq \Delta$. But if $\Box\varphi \in \Delta$ then by T also $\varphi \in \Delta$, as desired.

Now suppose Σ contains B, and suppose $R^\Sigma \Delta \Delta'$ for $\Delta, \Delta' \in W^\Sigma$. We need to show that $R^\Sigma \Delta' \Delta$, O.s., $\Box^{-1}\Delta' \subseteq \Delta$. By [Lema 52.9](#), this is equivalent to $\Diamond\Delta \subseteq \Delta'$. So suppose $\varphi \in \Delta$. By B, also $\Box\Diamond\varphi \in \Delta$. By the hypothesis that $R^\Sigma \Delta \Delta'$, we have that $\Box^{-1}\Delta \subseteq \Delta'$, and hence $\Diamond\varphi \in \Delta'$, as required.

Now suppose Σ contains 4, and suppose $R^\Sigma \Delta_1 \Delta_2$ and $R^\Sigma \Delta_2 \Delta_3$. We need to show $R^\Sigma \Delta_1 \Delta_3$. From the hypothesis we have both $\Box^{-1}\Delta_1 \subseteq \Delta_2$ and $\Box^{-1}\Delta_2 \subseteq \Delta_3$. In order to show $R^\Sigma \Delta_1 \Delta_3$ it suffices to show $\Box^{-1}\Delta_1 \subseteq \Delta_3$. So

52.8. FRAME COMPLETENESS

let $\psi \in \Box^{-1}\Delta_1$, O.s., $\Box\psi \in \Delta_1$. By 4, also $\Box\Box\psi \in \Delta_1$ and by hypothesis we get, first, that $\Box\psi \in \Delta_2$ and, second, that $\psi \in \Delta_3$, as desired.

Now suppose Σ contains 5, suppose $R^\Sigma\Delta_1\Delta_2$ and $R^\Sigma\Delta_1\Delta_3$. We need to show $R^\Sigma\Delta_2\Delta_3$. The first hypothesis gives $\Box^{-1}\Delta_1 \subseteq \Delta_2$, and the second hypothesis is equivalent to $\Diamond\Delta_3 \subseteq \Delta_1$, by [Lema 52.9](#). To show $R^\Sigma\Delta_2\Delta_3$, by [Lema 52.9](#), it suffices to show $\Diamond\Delta_3 \subseteq \Delta_2$. So let $\Diamond\varphi \in \Diamond\Delta_3$, O.s., $\varphi \in \Delta_3$. By the second hypothesis $\Diamond\varphi \in \Delta_1$ and by 5, $\Box\Diamond\varphi \in \Delta_1$ as well. But now the first hypothesis gives $\Diamond\varphi \in \Delta_2$, as desired. $\square$

As a corollary we obtain completeness results for a number of systems. For instance, we know that **S5** = **KT5** = **KTB4** is complete with respect to the class of all reflexive euclidean models, which is the same as the class of all reflexive, symmetric and transitive models.

*nml:com:fra:
thm:generaldet*

Teorema 52.17. *Let $\mathcal{C}_D$, $\mathcal{C}_T$, $\mathcal{C}_B$, $\mathcal{C}_4$, and $\mathcal{C}_5$ be the class of all serial, reflexive, symmetric, transitive, and euclidean models (respectively). Then for any schemas $\varphi_1, \dots, \varphi_n$ among D, T, B, 4, and 5, the system $\mathbf{K}\varphi_1 \dots \varphi_n$ is determined by the class of models $\mathcal{C} = \mathcal{C}_{\varphi_1} \cap \dots \cap \mathcal{C}_{\varphi_n}$.*

Proposição 52.18. *Let Σ be a normal modal logic; then:*

*nml:com:fra:
prop:anotherfive-a*

1. *If Σ contains the schema $\Diamond\varphi \rightarrow \Box\varphi$ then the canonical model for Σ is partially functional.*
2. *If Σ contains the schema $\Diamond\varphi \leftrightarrow \Box\varphi$ then the canonical model for Σ is functional.*
3. *If Σ contains the schema $\Box\Box\varphi \rightarrow \Box\varphi$ then the canonical model for Σ is weakly dense.*

(see [Tabela 50.2](#) for definitions of these frame properties).

Prova. 1. Suppose that Σ contains the schema $\Diamond\varphi \rightarrow \Box\varphi$, to show that R^Σ is partially functional we need to prove that for any $\Delta_1, \Delta_2, \Delta_3 \in W^\Sigma$, if $R^\Sigma\Delta_1\Delta_2$ and $R^\Sigma\Delta_1\Delta_3$ then $\Delta_2 = \Delta_3$. Since $R^\Sigma\Delta_1\Delta_2$ we have $\Box^{-1}\Delta_1 \subseteq \Delta_2$ and since $R^\Sigma\Delta_1\Delta_3$ also $\Box^{-1}\Delta_1 \subseteq \Delta_3$. The identity $\Delta_2 = \Delta_3$ will follow if we can establish the two inclusions $\Delta_2 \subseteq \Delta_3$ and $\Delta_3 \subseteq \Delta_2$. For the first inclusion, let $\varphi \in \Delta_2$; then $\Diamond\varphi \in \Delta_1$, and by the schema and deductive closure of Δ_1 also $\Box\varphi \in \Delta_1$, whence by the hypothesis that $R^\Sigma\Delta_1\Delta_3$, $\varphi \in \Delta_3$. The second inclusion is similar.

2. This follows immediately from part (1) and the seriality proof in [Teorema 52.16](#).
3. Suppose Σ contains the schema $\Box\Box\varphi \rightarrow \Box\varphi$ and to show that R^Σ is weakly dense, let $R^\Sigma\Delta_1\Delta_2$. We need to show that there is a complete Σ -consistent set Δ_3 such that $R^\Sigma\Delta_1\Delta_3$ and $R^\Sigma\Delta_3\Delta_2$. Let:

$$\Gamma = \Box^{-1}\Delta_1 \cup \Diamond\Delta_2.$$

It suffices to show that Γ is Σ -consistent, for then by Lindenbaum's Lemma it can be extended to a complete Σ -consistent set Δ_3 such that $\Box^{-1}\Delta_1 \subseteq \Delta_3$ and $\Diamond\Delta_2 \subseteq \Delta_3$, O.s., $R^\Sigma\Delta_1\Delta_3$ and $R^\Sigma\Delta_3\Delta_2$ (by [Lema 52.9](#)).

Suppose for contradiction that Γ is not consistent. Then there are [fórmulas](#) $\Box\varphi_1, \dots, \Box\varphi_n \in \Delta_1$ and $\psi_1, \dots, \psi_m \in \Delta_2$ such that

$$\varphi_1, \dots, \varphi_n, \Diamond\psi_1, \dots, \Diamond\psi_m \vdash_\Sigma \perp.$$

Since $\Diamond(\psi_1 \wedge \dots \wedge \psi_m) \rightarrow (\Diamond\psi_1 \wedge \dots \wedge \Diamond\psi_m)$ is [derivável](#) in every normal modal logic, we argue as follows, contradicting the consistency of Δ_2 :

$$\begin{aligned} & \varphi_1, \dots, \varphi_n, \Diamond\psi_1, \dots, \Diamond\psi_m \vdash_\Sigma \perp \\ & \varphi_1, \dots, \varphi_n \vdash_\Sigma (\Diamond\psi_1 \wedge \dots \wedge \Diamond\psi_m) \rightarrow \perp \\ & \quad \text{by the deduction theorem} \\ & \quad \text{Proposição 51.37(4), and TAUT} \\ & \varphi_1, \dots, \varphi_n \vdash_\Sigma \Diamond(\psi_1 \wedge \dots \wedge \psi_m) \rightarrow \perp \\ & \quad \text{since } \Sigma \text{ is normal} \\ & \varphi_1, \dots, \varphi_n \vdash_\Sigma \neg\Diamond(\psi_1 \wedge \dots \wedge \psi_m) \\ & \quad \text{by PL} \\ & \varphi_1, \dots, \varphi_n \vdash_\Sigma \Box\neg(\psi_1 \wedge \dots \wedge \psi_m) \\ & \quad \Box\neg \text{ for } \neg\Diamond \\ & \Box\varphi_1, \dots, \Box\varphi_n \vdash_\Sigma \Box\Box\neg(\psi_1 \wedge \dots \wedge \psi_m) \\ & \quad \text{by Lema 52.6} \\ & \Box\varphi_1, \dots, \Box\varphi_n \vdash_\Sigma \Box\neg(\psi_1 \wedge \dots \wedge \psi_m) \\ & \quad \text{by schema } \Box\Box\varphi \rightarrow \Box\varphi \\ & \Delta_1 \vdash_\Sigma \Box\neg(\psi_1 \wedge \dots \wedge \psi_m) \\ & \quad \text{by monotonicity, Proposição 51.37(1)} \\ & \Box\neg(\psi_1 \wedge \dots \wedge \psi_m) \in \Delta_1 \\ & \quad \text{by deductive closure;} \\ & \neg(\psi_1 \wedge \dots \wedge \psi_m) \in \Delta_2 \\ & \quad \text{since } R^\Sigma\Delta_1\Delta_2. \quad \square \end{aligned}$$

On the strength of these examples, one might think that every system Σ of modal logic is *complete*, in the sense that it proves every formula which is valid in every frame in which every theorem of Σ is valid. Unfortunately, there are

many systems that are not complete in this sense.

Capítulo 53

Filtrations and Decidability

-NoValue-

`content/normal-modal-logic/filtrations/introduction.tex`

53.1 Introduction

nml:fil:int:
sec

One important question about a logic is always whether it is decidable, O.s., if there is an effective procedure which will answer the question “is this fórmula valid.” Propositional logic is decidable: we can effectively test if a fórmula is a tautology by constructing a truth table, and for a given fórmula, the truth table is finite. But we can’t obviously test if a modal formula is true in all models, for there are infinitely many of them. We can list all the finite models relevant to a given formula, since only the assignment of subsets of worlds to *variáveis proposicionais* which actually occur in the fórmula are relevant. If the accessibility relation is fixed, the possible different assignments $V(p)$ are just all the subsets of W , and if $|W| = n$ there are 2^n of those. If our fórmula φ contains m *variáveis proposicionais* there are then 2^{nm} different models with n worlds. For each one, we can test if φ is true at all worlds, simply by computing the truth value of φ in each. Of course, we also have to check all possible accessibility relations, but there are only finitely many relations on n worlds as well (specifically, the number of subsets of $W \times W$, O.s., 2^{n^2}).

If we are not interested in the logic **K**, but a logic defined by some class of models (p.ex., the reflexive transitive models), we also have to be able to test if the accessibility relation is of the right kind. We can do that whenever the frames we are interested in are definable by modal *fórmulas* (p.ex., by testing if T and 4 valid in the frame). So, the idea would be to run through all the finite frames, test each one if it is a frame in the class we’re interested in, then list all the possible models on that frame and test if φ is true in each. If not, stop: φ is not valid in the class of models of interest.

There is a problem with this idea: we don’t know when, if ever, we can stop looking. If the fórmula has a finite countermodel, our procedure will find it.

But if it has no finite countermodel, we won't get an answer. The *fórmula* may be valid (no countermodels at all), or it may have only an infinite countermodel, which we'll never look at. This problem can be overcome if we can show that every *fórmula* that has a countermodel has a finite countermodel. If this is the case we say the logic has the *finite model property*.

But how would we show that a logic has the finite model property? One way of doing this would be to find a way to turn an infinite (counter)model of φ into a finite one. If that can be done, then whenever there is a model in which φ is not true, then the resulting finite model also makes φ not true. That finite model will show up on our list of all finite models, and we will eventually determine, for every formula that is not valid, that it isn't. Our procedure won't terminate if the formula is valid. If we can show in addition that there is some maximum size that the finite model our procedure provides can have, and that this maximum size depends only on the *fórmula* φ , we will have a size up to which we have to test finite models in our search for countermodels. If we haven't found a countermodel by then, there are none. Then our procedure will, in fact, decide the question "is φ valid?" for any *fórmula* φ .

A strategy that often works for turning infinite structures into finite structures is that of "identifying" *membros* of the structure which behave the same way in relevant respects. If there are infinitely many worlds in $\mathfrak{M}$ that behave the same in relevant respects, then we may be able to collect *all* worlds in finitely many (possibly infinite) "classes" of such worlds. In other words, we should partition the set of worlds in the right way, O.s., in such a way that each partition contains infinitely many worlds, but there are only finitely many partitions. Then we define a new model $\mathfrak{M}^*$ where the worlds are the partitions. Finitely many partitions in the old model give us finitely many worlds in the new model, O.s., a finite model. Let's call the partition a world w is in $[w]$. We'll want it to be the case that $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}^*, [w] \Vdash \varphi$, since we want the new model to be a countermodel to φ if the old one was. This requires that we define the partition, as well as the accessibility relation of $\mathfrak{M}^*$ in the right way.

To see how this would go, first imagine we have no accessibility relation. $\mathfrak{M}, w \Vdash \Box\psi$ iff for some $v \in W$, $\mathfrak{M}, v \Vdash \Box\psi$, and the same for $\mathfrak{M}^*$, except with $[w]$ and $[v]$. As a first idea, let's say that two worlds u and v are equivalent (belong to the same partition) if they agree on all *variáveis proposicionais* in $\mathfrak{M}$, O.s., $\mathfrak{M}, u \Vdash p$ iff $\mathfrak{M}, v \Vdash p$. Let $V^*(p) = \{[w] : \mathfrak{M}, w \Vdash p\}$. Our aim is to show that $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}^*, [w] \Vdash \varphi$. Obviously, we'd prove this by induction: The base case would be $\varphi \equiv p$. First suppose $\mathfrak{M}, w \Vdash p$. Then $[w] \in V^*(p)$ by definition, so $\mathfrak{M}^*, [w] \Vdash p$. Now suppose that $\mathfrak{M}^*, [w] \Vdash p$. That means that $[w] \in V^*(p)$, O.s., for some v equivalent to w , $\mathfrak{M}, v \Vdash p$. But " w equivalent to v " means " w and v make all the same *variáveis proposicionais* true," so $\mathfrak{M}, w \Vdash p$. Now for the inductive step, p.ex., $\varphi \equiv \neg\psi$. Then $\mathfrak{M}, w \Vdash \neg\psi$ iff $\mathfrak{M}, w \not\Vdash \psi$ iff $\mathfrak{M}^*, [w] \not\Vdash \psi$ (by inductive hypothesis) iff $\mathfrak{M}^*, [w] \Vdash \neg\psi$. Similarly for the other non-modal operators. It also works for $\Box$: suppose $\mathfrak{M}^*, [w] \Vdash \Box\psi$. That means that for every $[u]$, $\mathfrak{M}^*, [u] \Vdash \psi$. By inductive hypothesis, for every u , $\mathfrak{M}, u \Vdash \psi$. Consequently, $\mathfrak{M}, w \Vdash \Box\psi$.

53.2. PRELIMINARIES

In the general case, where we have to also define the accessibility relation for $\mathfrak{M}^*$, things are more complicated. We'll call a model $\mathfrak{M}^*$ a *filtration* if its accessibility relation R^* satisfies the conditions required to make the inductive proof above go through. Then any filtration $\mathfrak{M}^*$ will make φ true at $[w]$ iff $\mathfrak{M}$ makes φ true at w . However, now we also have to show that there *are* filtrations, O.s., we can define R^* so that it satisfies the required conditions. In order for this to work, however, we have to require that worlds u, v count as equivalent not just when they agree on all *variáveis proposicionais*, but on all sub-fórmulas of φ . Since φ has only finitely many sub-fórmulas, this will still guarantee that the filtration is finite. There is not just one way to define a filtration, and in order to make sure that the accessibility relation of the filtration satisfies the required properties (p.ex., reflexive, transitive, etc.) we have to be inventive with the definition of R^* .

-NoValue-

content/normal-modal-logic/filtrations/preliminaries.tex

53.2 Preliminaries

nml:fil:pre:
sec Filtrations allow us to establish the decidability of our systems of modal logic by showing that they have the *finite model property*, O.s., that any *fórmula* that is true (false) in a model is also true (false) in a *finite* model. Filtrations are defined relative to sets of *fórmulas* which are closed under subformulas.

nml:fil:pre:
defn:modallyclosed **Definição 53.1.** A set Γ of *fórmulas* is *closed under subformulas* if it contains every subformula of a *fórmula* in Γ . Further, Γ is *modally closed* if it is closed under subformulas and moreover $\varphi \in \Gamma$ implies $\Box\varphi, \Diamond\varphi \in \Gamma$.

For instance, given a *fórmula* φ , the set of all its sub-fórmulas is closed under sub-fórmulas. When we're defining a filtration of a model through the set of sub-fórmulas of φ , it will have the property we're after: it makes φ true (false) iff the original model does.

The set of worlds of a filtration of $\mathfrak{M}$ through Γ is defined as the set of all equivalence classes of the following equivalence relation.

Definição 53.2. Let $\mathfrak{M} = \langle W, R, V \rangle$ and suppose Γ is closed under sub-fórmulas. Define a relation $\equiv$ on W to hold of any two worlds that make the same *fórmulas* from Γ true, O.s.:

$$u \equiv v \quad \text{if and only if} \quad \forall \varphi \in \Gamma : \mathfrak{M}, u \Vdash \varphi \Leftrightarrow \mathfrak{M}, v \Vdash \varphi.$$

The equivalence class $[w]_{\equiv}$ of a world w , or $[w]$ for short, is the set of all worlds $\equiv$ -equivalent to w :

$$[w] = \{v : v \equiv w\}.$$

Proposição 53.3. *Given $\mathfrak{M}$ and Γ , $\equiv$ as defined above is an equivalence relation, O.s., it is reflexive, symmetric, and transitive.*

Prova. The relation $\equiv$ is reflexive, since w makes exactly the same **fórmulas** from Γ true as itself. It is symmetric since if u makes the same **fórmulas** from Γ true as v , the same holds for v and u . It is also transitive, since if u makes the same **fórmulas** from Γ true as v , and v as w , then u makes the same **fórmulas** from Γ true as w . $\square$

The relation $\equiv$, like any equivalence relation, divides W into *partitions*, O.s., subsets of W which are pairwise disjoint, and together cover all of W . Every $w \in W$ is **um membro** of one of the partitions, namely of $[w]$, since $w \equiv w$. So the partitions $[w]$ cover all of W . They are pairwise disjoint, for if $u \in [w]$ and $u \in [v]$, then $u \equiv w$ and $u \equiv v$, and by symmetry and transitivity, $w \equiv v$, and so $[w] = [v]$.

-NoValue-

content/normal-modal-logic/filtrations/filtrations-def.tex

53.3 Filtrations

Rather than define “the” filtration of $\mathfrak{M}$ through Γ , we define when a model $\mathfrak{M}^*$ counts as a filtration of $\mathfrak{M}$. All filtrations have the same set of worlds W^* and the same valuation V^* . But different filtrations may have different accessibility relations R^* . To count as a filtration, R^* has to satisfy a number of conditions, however. These conditions are exactly what we’ll require to prove the main result, namely that $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}^*, [w] \Vdash \varphi$, provided $\varphi \in \Gamma$.

nml:fil:fil:
sec

Definição 53.4. Let Γ be closed under subformulas and $\mathfrak{M} = \langle W, R, V \rangle$. A *filtration of $\mathfrak{M}$ through Γ* is any model $\mathfrak{M}^* = \langle W^*, R^*, V^* \rangle$, where:

nml:fil:fil:
defn:filtration

1. $W^* = \{[w] : w \in W\}$;
2. For any $u, v \in W$:
 - a) If Ruv then $R^*[u][v]$;
 - b) If $R^*[u][v]$ then for any $\Box\varphi \in \Gamma$, if $\mathfrak{M}, u \Vdash \Box\varphi$ then $\mathfrak{M}, v \Vdash \varphi$;
 - c) If $R^*[u][v]$ then for any $\Diamond\varphi \in \Gamma$, if $\mathfrak{M}, v \Vdash \varphi$ then $\mathfrak{M}, u \Vdash \Diamond\varphi$.
3. $V^*(p) = \{[u] : u \in V(p)\}$.

nml:fil:fil:
defn:filtration-R
nml:fil:fil:
defn:filtration-R1
nml:fil:fil:
defn:filtration-R2
nml:fil:fil:
defn:filtration-R3

It’s worthwhile thinking about what $V^*(p)$ is: the set consisting of the equivalence classes $[w]$ of all worlds w where p is true in $\mathfrak{M}$. On the one hand, if $w \in V(p)$, then $[w] \in V^*(p)$ by that definition. However, it is not necessarily the case that if $[w] \in V^*(p)$, then $w \in V(p)$. If $[w] \in V^*(p)$ we are only guaranteed that $[w] = [u]$ for *some* $u \in V(p)$. Of course, $[w] = [u]$ means that $w \equiv u$. So, when $[w] \in V^*(p)$ we can (only) conclude that $w \equiv u$ for some $u \in V(p)$.

Teorema 53.5. If $\mathfrak{M}^*$ is a filtration of $\mathfrak{M}$ through Γ , then for every $\varphi \in \Gamma$ and $w \in W$, we have $\mathfrak{M}, w \Vdash \varphi$ if and only if $\mathfrak{M}^*, [w] \Vdash \varphi$.

nml:fil:fil:
thm:filtrations

53.3. FILTRATIONS

Prova. By induction on φ , using the fact that Γ is closed under subformulas. Since $\varphi \in \Gamma$ and Γ is closed under sub-fórmulas, all sub-fórmulas of φ are also $\in \Gamma$. Hence in each inductive step, the induction hypothesis applies to the sub-fórmulas of φ .

1. $\varphi \equiv \perp$: Neither $\mathfrak{M}, w \Vdash \varphi$ nor $\mathfrak{M}^*, [w] \Vdash \varphi$.
2. $\varphi \equiv \top$: Both $\mathfrak{M}, w \Vdash \varphi$ and $\mathfrak{M}^*, [w] \Vdash \varphi$.
3. $\varphi \equiv p$: The left-to-right direction is immediate, as $\mathfrak{M}, w \Vdash \varphi$ only if $w \in V(p)$, which implies $[w] \in V^*(p)$, O.s., $\mathfrak{M}^*, [w] \Vdash \varphi$. Conversely, suppose $\mathfrak{M}^*, [w] \Vdash \varphi$, O.s., $[w] \in V^*(p)$. Then for some $v \in V(p)$, $w \equiv v$. Of course then also $\mathfrak{M}, v \Vdash p$. Since $w \equiv v$, w and v make the same fórmulas from Γ true. Since by assumption $p \in \Gamma$ and $\mathfrak{M}, v \Vdash p$, $\mathfrak{M}, w \Vdash p$.
4. $\varphi \equiv \neg\psi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \nVdash \psi$. By induction hypothesis, $\mathfrak{M}, w \nVdash \psi$ iff $\mathfrak{M}^*, [w] \nVdash \psi$. Finally, $\mathfrak{M}^*, [w] \nVdash \psi$ iff $\mathfrak{M}^*, [w] \Vdash \varphi$.
5. $\varphi \equiv (\psi \wedge \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ and $\mathfrak{M}, w \Vdash \chi$. By induction hypothesis, $\mathfrak{M}, w \Vdash \psi$ iff $\mathfrak{M}^*, [w] \Vdash \psi$, and $\mathfrak{M}, w \Vdash \chi$ iff $\mathfrak{M}^*, [w] \Vdash \chi$. And $\mathfrak{M}^*, [w] \Vdash \varphi$ iff $\mathfrak{M}^*, [w] \Vdash \psi$ and $\mathfrak{M}^*, [w] \Vdash \chi$.
6. $\varphi \equiv (\psi \vee \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ or $\mathfrak{M}, w \Vdash \chi$. By induction hypothesis, $\mathfrak{M}, w \Vdash \psi$ iff $\mathfrak{M}^*, [w] \Vdash \psi$, and $\mathfrak{M}, w \Vdash \chi$ iff $\mathfrak{M}^*, [w] \Vdash \chi$. And $\mathfrak{M}^*, [w] \Vdash \varphi$ iff $\mathfrak{M}^*, [w] \Vdash \psi$ or $\mathfrak{M}^*, [w] \Vdash \chi$.
7. $\varphi \equiv (\psi \rightarrow \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \nVdash \psi$ or $\mathfrak{M}, w \Vdash \chi$. By induction hypothesis, $\mathfrak{M}, w \Vdash \psi$ iff $\mathfrak{M}^*, [w] \Vdash \psi$, and $\mathfrak{M}, w \Vdash \chi$ iff $\mathfrak{M}^*, [w] \Vdash \chi$. And $\mathfrak{M}^*, [w] \Vdash \varphi$ iff $\mathfrak{M}^*, [w] \nVdash \psi$ or $\mathfrak{M}^*, [w] \Vdash \chi$.
8. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ and $\mathfrak{M}, w \Vdash \chi$, or $\mathfrak{M}, w \nVdash \psi$ and $\mathfrak{M}, w \nVdash \chi$. By induction hypothesis, $\mathfrak{M}, w \Vdash \psi$ iff $\mathfrak{M}^*, [w] \Vdash \psi$, and $\mathfrak{M}, w \Vdash \chi$ iff $\mathfrak{M}^*, [w] \Vdash \chi$. And $\mathfrak{M}^*, [w] \Vdash \varphi$ iff $\mathfrak{M}^*, [w] \Vdash \psi$ and $\mathfrak{M}^*, [w] \Vdash \chi$, or $\mathfrak{M}^*, [w] \nVdash \psi$ and $\mathfrak{M}^*, [w] \nVdash \chi$.
9. $\varphi \equiv \Box\psi$: Suppose $\mathfrak{M}, w \Vdash \varphi$; to show that $\mathfrak{M}^*, [w] \Vdash \varphi$, let v be such that $R^*[w][v]$. From [Definição 53.4\(2b\)](#), we have that $\mathfrak{M}, v \Vdash \psi$, and by inductive hypothesis $\mathfrak{M}^*, [v] \Vdash \psi$. Since v was arbitrary, $\mathfrak{M}^*, [w] \Vdash \varphi$ follows.
Conversely, suppose $\mathfrak{M}^*, [w] \Vdash \varphi$ and let v be arbitrary such that Rwv . From [Definição 53.4\(2a\)](#), we have $R^*[w][v]$, so that $\mathfrak{M}^*, [v] \Vdash \psi$; by inductive hypothesis $\mathfrak{M}, v \Vdash \psi$, and since v was arbitrary, $\mathfrak{M}, w \Vdash \varphi$.
10. $\varphi \equiv \Diamond\psi$: Suppose $\mathfrak{M}, w \Vdash \varphi$. Then for some $v \in W$, Rwv and $\mathfrak{M}, v \Vdash \psi$. By inductive hypothesis $\mathfrak{M}^*, [v] \Vdash \psi$, and by [Definição 53.4\(2a\)](#), we have $R^*[w][v]$. Thus, $\mathfrak{M}^*, [w] \Vdash \varphi$.
Now suppose $\mathfrak{M}^*, [w] \Vdash \varphi$. Then for some $[v] \in W^*$ with $R^*[w][v]$, $\mathfrak{M}^*, [v] \Vdash \psi$. By inductive hypothesis $\mathfrak{M}, v \Vdash \psi$. By [Definição 53.4\(2c\)](#), we have that $\mathfrak{M}, w \Vdash \varphi$. $\square$

Problema 53.1. Complete the proof of [Teorema 53.5](#)

What holds for truth at worlds in a model also holds for truth in a model and validity in a class of models.

Corolário 53.6. *Let Γ be closed under subformulas. Then:*

1. *If $\mathfrak{M}^*$ is a filtration of $\mathfrak{M}$ through Γ then for any $\varphi \in \Gamma$: $\mathfrak{M} \Vdash \varphi$ if and only if $\mathfrak{M}^* \Vdash \varphi$.*
2. *If $\mathcal{C}$ is a class of models and $\Gamma(\mathcal{C})$ is the class of Γ -filtrations of models in $\mathcal{C}$, then any [fórmula](#) $\varphi \in \Gamma$ is valid in $\mathcal{C}$ if and only if it is valid in $\Gamma(\mathcal{C})$.*

-NoValue-

[content/normal-modal-logic/filtrations/examples-of-filtrations.tex](#)

53.4 Examples of Filtrations

We have not yet shown that there are any filtrations. But indeed, for any model $\mathfrak{M}$, there are many filtrations of $\mathfrak{M}$ through Γ . We identify two, in particular: the finest and coarsest filtrations. Filtrations of the same models will differ in their accessibility relation (as [Definição 53.4](#) stipulates directly what W^* and V^* should be). The finest filtration will have as few related worlds as possible, whereas the coarsest will have as many as possible. nml:fil:exf:
sec

Definição 53.7. Where Γ is closed under subformulas, the *finest* filtration $\mathfrak{M}^*$ of a model $\mathfrak{M}$ is defined by putting:

$$R^*[u][v] \quad \text{if and only if} \quad \exists u' \in [u] \exists v' \in [v] : Ru'v'.$$

Proposição 53.8. *The finest filtration $\mathfrak{M}^*$ is indeed a filtration.* nml:fil:exf:
prop:finest

Prova. We need to check that R^* , so defined, satisfies [Definição 53.4\(2\)](#). We check the three conditions in turn.

If Ruv then since $u \in [u]$ and $v \in [v]$, also $R^*[u][v]$, so [\(2a\)](#) is satisfied.

For [\(2b\)](#), suppose $\Box\varphi \in \Gamma$, $R^*[u][v]$, and $\mathfrak{M}, u \Vdash \Box\varphi$. By definition of R^* , there are $u' \equiv u$ and $v' \equiv v$ such that $Ru'v'$. Since u and u' agree on Γ , also $\mathfrak{M}, u' \Vdash \Box\varphi$, so that $\mathfrak{M}, v' \Vdash \varphi$. By closure of Γ under sub-[fórmulas](#), v and v' agree on φ , so $\mathfrak{M}, v \Vdash \varphi$, as desired.

To verify [\(2c\)](#), suppose $\Diamond\varphi \in \Gamma$, $R^*[u][v]$, and $\mathfrak{M}, v \Vdash \varphi$. By definition of R^* , there are $u' \equiv u$ and $v' \equiv v$ such that $Ru'v'$. Since v and v' agree on Γ , and Γ is closed under sub-[fórmulas](#), also $\mathfrak{M}, v' \Vdash \varphi$, so that $\mathfrak{M}, u' \Vdash \Diamond\varphi$. Since u and u' also agree on Γ , $\mathfrak{M}, u \Vdash \Diamond\varphi$. □

Problema 53.2. Complete the proof of [Proposição 53.8](#).

53.4. EXAMPLES OF FILTRATIONS

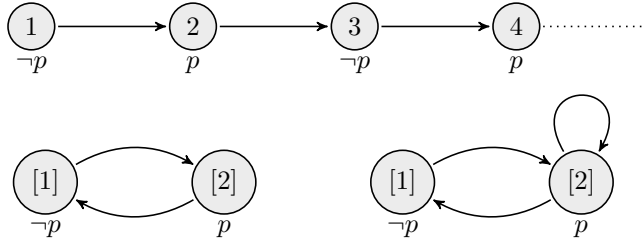


Figura 53.1: An infinite model and its filtrations.

nml:fil:exf:
fig:ex-filtration

Definição 53.9. Where Γ is closed under subformulas, the *coarsest* filtration $\mathfrak{M}^*$ of a model $\mathfrak{M}$ is defined by putting $R^*[u][v]$ if and only if *both* of the following conditions are met:

1. If $\Box\varphi \in \Gamma$ and $\mathfrak{M}, u \Vdash \Box\varphi$ then $\mathfrak{M}, v \Vdash \varphi$;
2. If $\Diamond\varphi \in \Gamma$ and $\mathfrak{M}, v \Vdash \varphi$ then $\mathfrak{M}, u \Vdash \Diamond\varphi$.

nml:fil:exf:
defn:coarsest-Box
nml:fil:exf:
defn:coarsest-Diamond

Proposição 53.10. *The coarsest filtration $\mathfrak{M}^*$ is indeed a filtration.*

Prova. Given the definition of R^* , the only condition that is left to verify is the implication from Ruv to $R^*[u][v]$. So assume Ruv . Suppose $\Box\varphi \in \Gamma$ and $\mathfrak{M}, u \Vdash \Box\varphi$; then obviously $\mathfrak{M}, v \Vdash \varphi$, and (1) is satisfied. Suppose $\Diamond\varphi \in \Gamma$ and $\mathfrak{M}, v \Vdash \varphi$. Then $\mathfrak{M}, u \Vdash \Diamond\varphi$ since Ruv , and (2) is satisfied. $\square$

Exemplo 53.11. Let $W = \mathbb{Z}^+$, Rnm iff $m = n + 1$, and $V(p) = \{2n : n \in \mathbb{N}\}$. The model $\mathfrak{M} = \langle W, R, V \rangle$ is depicted in [Figura 53.1](#). The worlds are 1, 2, etc.; each world can access exactly one other world—its successor—and p is true at all and only the even numbers.

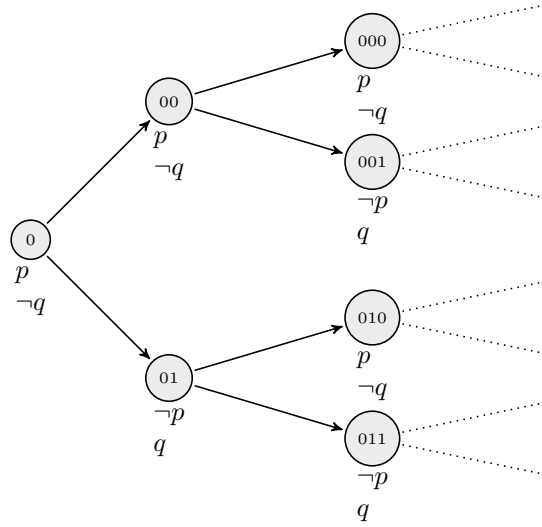
Now let Γ be the set of sub-fórmulas of $\Box p \rightarrow p$, O.s., $\{p, \Box p, \Box p \rightarrow p\}$. p is true at all and only the even numbers, $\Box p$ is true at all and only the odd numbers, so $\Box p \rightarrow p$ is true at all and only the even numbers. In other words, every odd number makes $\Box p$ true but p and $\Box p \rightarrow p$ false; every even number makes p and $\Box p \rightarrow p$ true, but $\Box p$ false. So $W^* = \{[1], [2]\}$, where $[1] = \{1, 3, 5, \dots\}$ and $[2] = \{2, 4, 6, \dots\}$. Since $2 \in V(p)$, $[2] \in V^*(p)$; since $1 \notin V(p)$, $[1] \notin V^*(p)$. So $V^*(p) = \{[2]\}$.

Any filtration based on W^* must have an accessibility relation that includes $\langle [1], [2] \rangle, \langle [2], [1] \rangle$: since $R12$, we must have $R^*[1][2]$ by [Definição 53.4\(2a\)](#), and since $R23$ we must have $R^*[2][3]$, and $[3] = [1]$. It cannot include $\langle [1], [1] \rangle$: if it did, we'd have $R^*[1][1]$, $\mathfrak{M}, 1 \Vdash \Box p$ but $\mathfrak{M}, 1 \not\Vdash p$, contradicting (2b). Nothing requires or rules out that $R^*[2][2]$. So, there are two possible filtrations of $\mathfrak{M}$, corresponding to the two accessibility relations

$$\{\langle [1], [2] \rangle, \langle [2], [1] \rangle\} \text{ and } \{\langle [1], [2] \rangle, \langle [2], [1] \rangle, \langle [2], [2] \rangle\}.$$

In either case, p and $\Box p \rightarrow p$ are false and $\Box p$ is true at $[1]$; p and $\Box p \rightarrow p$ are true and $\Box p$ is false at $[2]$.

Problema 53.3. Consider the following model $\mathfrak{M} = \langle W, R, V \rangle$ where $W = \{0\sigma : \sigma \in \mathbb{B}^*\}$, the set of sequences of 0s and 1s starting with 0, with $R\sigma\sigma'$ iff $\sigma' = \sigma 0$ or $\sigma' = \sigma 1$, and $V(p) = \{\sigma 0 : \sigma \in \mathbb{B}^*\}$ and $V(q) = \{\sigma 1 : \sigma \in \mathbb{B}^* \setminus \{1\}\}$. Here's a picture:



We have $\mathfrak{M}, w \not\models \Box(p \vee q) \rightarrow (\Box p \vee \Box q)$ for every w .

Let Γ be the set of sub-fórmulas of $\Box(p \vee q) \rightarrow (\Box p \vee \Box q)$. What are W^* and V^* ? What is the accessibility relation of the finest filtration of $\mathfrak{M}$? Of the coarsest?

-NoValue-

content/normal-modal-logic/filtrations/finite.tex

53.5 Filtrations are Finite

We've defined filtrations for any set Γ that is closed under sub-fórmulas. Nothing in the definition itself guarantees that filtrations are finite. In fact, when Γ is infinite (p.ex., is the set of all fórmulas), it may well be infinite. However, if Γ is finite (p.ex., when it is the set of sub-fórmulas of a given fórmula φ), so is any filtration through Γ .

nml:fil:fin:
sec

Proposição 53.12. *If Γ is finite then any filtration $\mathfrak{M}^*$ of a model $\mathfrak{M}$ through Γ is also finite.*

nml:fil:fin:
prop:fil-are-finite

Prova. The size of W^* is the number of different classes $[w]$ under the equivalence relation $\equiv$. Any two worlds u, v in such class—that is, any u and v such that $u \equiv v$ —agree on all fórmulas φ in Γ , $\varphi \in \Gamma$ either φ is true at both u and v , or at neither. So each class $[w]$ corresponds to subset of Γ , namely the set of

53.6. K AND S5 HAVE THE FINITE MODEL PROPERTY

all $\varphi \in \Gamma$ such that φ is true at the worlds in $[w]$. No two different classes $[u]$ and $[v]$ correspond to the same subset of Γ . For if the set of **fórmulas** true at u and that of **fórmulas** true at v are the same, then u and v agree on all formulas in Γ , O.s., $u \equiv v$. But then $[u] = [v]$. So, there is **uma injetiva** function from W^* to $\wp(\Gamma)$, and hence $|W^*| \leq |\wp(\Gamma)|$. Hence if Γ contains n sentences, the cardinality of W^* is no greater than 2^n . $\square$

-NoValue-

content/normal-modal-logic/filtrations/S5-fmp.tex

53.6 K and S5 have the Finite Model Property

nml:fil:fmp:
sec

Definição 53.13. A system Σ of modal logic is said to have the *finite model property* if whenever a **fórmula** φ is true at a world in a model of Σ then φ is true at a world in a *finite* model of Σ .

nml:fil:fmp:
prop:K-fmp

Proposição 53.14. **K** has the finite model property.

Prova. **K** is the set of valid **fórmulas**, O.s., any model is a model of **K**. By **Teorema 53.5**, if $\mathfrak{M}, w \Vdash \varphi$, then $\mathfrak{M}^*, w \Vdash \varphi$ for any filtration of $\mathfrak{M}$ through the set Γ of sub-**fórmulas** of φ . Any **fórmula** only has finitely many sub-**fórmulas**, so Γ is finite. By **Proposição 53.12**, $|W^*| \leq 2^n$, where n is the number of **fórmulas** in Γ . And since **K** imposes no restriction on models, $\mathfrak{M}^*$ is a **K**-model. $\square$

To show that a logic **L** has the finite model property via filtrations it is essential that the filtration of an **L**-model is itself a **L**-model. Often this requires a fair bit of work, and not any filtration yields a **L**-model. However, for universal models, this still holds.

nml:fil:fmp:
prop:univ-fin

Proposição 53.15. Let $\mathcal{U}$ be the class of universal models (see **Proposição 50.14**) and $\mathcal{U}_{\text{Fin}}$ the class of all finite universal models. Then any **fórmula** φ is valid in $\mathcal{U}$ if and only if it is valid in $\mathcal{U}_{\text{Fin}}$.

Prova. Finite universal models are universal models, so the left-to-right direction is trivial. For the right-to left direction, suppose that φ is false at some world w in a universal model $\mathfrak{M}$. Let Γ contain φ as well as all of its sub-formulas; clearly Γ is finite. Take a filtration $\mathfrak{M}^*$ of $\mathfrak{M}$; then $\mathfrak{M}^*$ is finite by **Proposição 53.12**, and by **Teorema 53.5**, φ is false at $[w]$ in $\mathfrak{M}^*$. It remains to observe that $\mathfrak{M}^*$ is also universal: given u and v , by hypothesis Ruv and by **Definição 53.4(2)**, also $R^*[u][v]$. $\square$

nml:fil:fmp:
cor:S5fmp

Corolário 53.16. **S5** has the finite model property.

Prova. By [Proposição 50.14](#), if φ is true at a world in some reflexive and euclidean model then it is true at a world in a universal model. By [Proposição 53.15](#), it is true at a world in a finite universal model (namely the filtration of the model through the set of sub-fórmulas of φ). Every universal model is also reflexive and euclidean; so φ is true at a world in a finite reflexive euclidean model. $\square$

Problema 53.4. Show that any filtration of a serial or reflexive model is also serial or reflexive (respectively).

Problema 53.5. Find a non-symmetric (non-transitive, non-euclidean) filtration of a symmetric (transitive, euclidean) model.

-NoValue-

[content/normal-modal-logic/filtrations/S5-decidable.tex](#)

53.7 S5 is Decidable

The finite model property gives us an easy way to show that systems of modal logic given by schemas are *decidable* (O.s., that there is a computable procedure to determine whether a fórmula is derivável in the system or not). nml:fil:dec:
sec

Teorema 53.17. *S5 is decidable.*

Prova. Let φ be given, and suppose the variáveis proposicionais occurring in φ are among $p_1, \dots, p_k$. Since for each n there are only finitely many models with n worlds assigning a value to $p_1, \dots, p_k$, we can enumerate, *in parallel*, all the theorems of **S5** by generating proofs in some systematic way; and all the models containing 1, 2, ... worlds and checking whether φ fails at a world in some such model. Eventually one of the two parallel processes will give an answer, as by [Teorema 52.17](#) and [Corolário 53.16](#), either φ is derivável or it fails in a finite universal model. $\square$

The above proof works for **S5** because filtrations of universal models are automatically universal. The same holds for reflexivity and seriality, but more work is needed for other properties.

-NoValue-

[content/normal-modal-logic/filtrations/more-filtrations.tex](#)

53.8 Filtrations and Properties of Accessibility

As noted, filtrations of universal, serial, and reflexive models are always also universal, serial, or reflexive. But not every filtration of a symmetric or transitive model is symmetric or transitive, respectively. In some cases, however, it nml:fil:acc:
sec

53.8. FILTRATIONS AND PROPERTIES OF ACCESSIBILITY

is possible to define filtrations so that this does hold. In order to do so, we proceed as in the definition of the coarsest filtration, but add additional conditions to the definition of R^* . Let Γ be closed under sub-fórmulas. Consider the relations $C_i(u, v)$ in [Tabela 53.1](#) between worlds u, v in a model $\mathfrak{M} = \langle W, R, V \rangle$. We can define $R^*[u][v]$ on the basis of combinations of these conditions. For instance, if we stipulate that $R^*[u][v]$ iff the condition $C_1(u, v)$ holds, we get exactly the coarsest filtration. If we stipulate $R^*[u][v]$ iff both $C_1(u, v)$ and $C_2(u, v)$ hold, we get a different filtration. It is “finer” than the coarsest since fewer pairs of worlds satisfy $C_1(u, v)$ and $C_2(u, v)$ than $C_1(u, v)$ alone.

$C_1(u, v)$:	if $\Box\varphi \in \Gamma$ and $\mathfrak{M}, u \Vdash \Box\varphi$ then $\mathfrak{M}, v \Vdash \varphi$; and if $\Diamond\varphi \in \Gamma$ and $\mathfrak{M}, v \Vdash \varphi$ then $\mathfrak{M}, u \Vdash \Diamond\varphi$;
$C_2(u, v)$:	if $\Box\varphi \in \Gamma$ and $\mathfrak{M}, v \Vdash \Box\varphi$ then $\mathfrak{M}, u \Vdash \varphi$; and if $\Diamond\varphi \in \Gamma$ and $\mathfrak{M}, u \Vdash \varphi$ then $\mathfrak{M}, v \Vdash \Diamond\varphi$;
$C_3(u, v)$:	if $\Box\varphi \in \Gamma$ and $\mathfrak{M}, u \Vdash \Box\varphi$ then $\mathfrak{M}, v \Vdash \Box\varphi$; and if $\Diamond\varphi \in \Gamma$ and $\mathfrak{M}, v \Vdash \Diamond\varphi$ then $\mathfrak{M}, u \Vdash \Diamond\varphi$;
$C_4(u, v)$:	if $\Box\varphi \in \Gamma$ and $\mathfrak{M}, v \Vdash \Box\varphi$ then $\mathfrak{M}, u \Vdash \Box\varphi$; and if $\Diamond\varphi \in \Gamma$ and $\mathfrak{M}, u \Vdash \Diamond\varphi$ then $\mathfrak{M}, v \Vdash \Diamond\varphi$;

Tabela 53.1: Conditions on possible worlds for defining filtrations.

nml:fil:acc:
tab:Cn-filtrations
nml:fil:acc:
thm:more-filtrations

Teorema 53.18. *Let $\mathfrak{M} = \langle W, R, V \rangle$ be a model, Γ closed under sub-fórmulas. Let W^* and V^* be defined as in [Definição 53.4](#). Then:*

1. *Suppose $R^*[u][v]$ if and only if $C_1(u, v) \wedge C_2(u, v)$. Then R^* is symmetric, and $\mathfrak{M}^* = \langle W^*, R^*, V^* \rangle$ is a filtration if $\mathfrak{M}$ is symmetric.*
2. *Suppose $R^*[u][v]$ if and only if $C_1(u, v) \wedge C_3(u, v)$. Then R^* is transitive, and $\mathfrak{M}^* = \langle W^*, R^*, V^* \rangle$ is a filtration if $\mathfrak{M}$ is transitive.*
3. *Suppose $R^*[u][v]$ if and only if $C_1(u, v) \wedge C_2(u, v) \wedge C_3(u, v) \wedge C_4(u, v)$. Then R^* is symmetric and transitive, and $\mathfrak{M}^* = \langle W^*, R^*, V^* \rangle$ is a filtration if $\mathfrak{M}$ is symmetric and transitive.*
4. *Suppose R^* is defined as $R^*[u][v]$ if and only if $C_1(u, v) \wedge C_3(u, v) \wedge C_4(u, v)$. Then R^* is transitive and euclidean, and $\mathfrak{M}^* = \langle W^*, R^*, V^* \rangle$ is a filtration if $\mathfrak{M}$ is transitive and euclidean.*

Prova. 1. It's immediate that R^* is symmetric, since $C_1(u, v) \Leftrightarrow C_2(v, u)$ and $C_2(u, v) \Leftrightarrow C_1(v, u)$. So it's left to show that if $\mathfrak{M}$ is symmetric then $\mathfrak{M}^*$ is a filtration through Γ . Condition $C_1(u, v)$ guarantees that [\(2b\)](#) and [\(2c\)](#) of [Definição 53.4](#) are satisfied. So we just have to verify [Definição 53.4\(2a\)](#), O.s., that Ruv implies $R^*[u][v]$.

So suppose Ruv . To show $R^*[u][v]$ we need to establish that $C_1(u, v)$ and $C_2(u, v)$. For C_1 : if $\Box\varphi \in \Gamma$ and $\mathfrak{M}, u \Vdash \Box\varphi$ then also $\mathfrak{M}, v \Vdash \varphi$ (since Ruv). Similarly, if $\Diamond\varphi \in \Gamma$ and $\mathfrak{M}, v \Vdash \varphi$ then $\mathfrak{M}, u \Vdash \Diamond\varphi$ since Ruv . For C_2 : if $\Box\varphi \in \Gamma$ and $\mathfrak{M}, v \Vdash \Box\varphi$ then Ruv implies Rvu by symmetry, so

that $\mathfrak{M}, u \Vdash \varphi$. Similarly, if $\Diamond\varphi \in \Gamma$ and $\mathfrak{M}, u \Vdash \varphi$ then $\mathfrak{M}, v \Vdash \Diamond\varphi$ (since Rvu by symmetry).

2. Exercise.

3. Exercise.

4. Exercise. □

Problema 53.6. Complete the proof of [Teorema 53.18](#).

-NoValue-

[content/normal-modal-logic/filtrations/euclidean-filtrations.tex](#)

53.9 Filtrations of Euclidean Models

The approach of [Seção 53.8](#) does not work in the case of models that are euclidean or serial and euclidean. Consider the model at the top of [Figura 53.2](#), which is both euclidean and serial. Let $\Gamma = \{p, \Box p\}$. When taking a filtration through Γ , then $[w_1] = [w_3]$ since w_1 and w_3 are the only worlds that agree on Γ . Any filtration will also have the arrow inherited from $\mathfrak{M}$, as depicted in [Figura 53.3](#). That model isn't euclidean. Moreover, we cannot add arrows to that model in order to make it euclidean. We would have to add double arrows between $[w_2]$ and $[w_4]$, and then also between w_2 and w_5 . But $\Box p$ is supposed to be true at w_2 , while p is false at w_5 .

[nml:fil:euc:](#)
[sec](#)

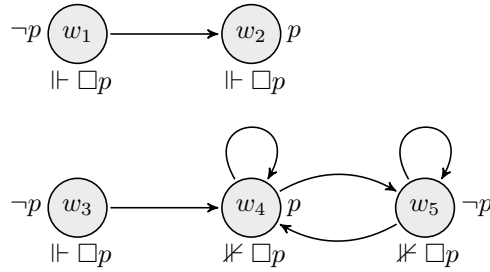


Figura 53.2: A serial and euclidean model.

In particular, to obtain a euclidean filtration it is not enough to consider filtrations through arbitrary Γ 's closed under sub-[fórmulas](#). Instead we need to consider sets Γ that are *modally closed* (see [Definição 53.1](#)). Such sets of sentences are infinite, and therefore do not immediately yield a finite model property or the decidability of the corresponding system.

[nml:fil:euc:](#)
[fig:ser-eucl](#)

Teorema 53.19. Let Γ be modally closed, $\mathfrak{M} = \langle W, R, V \rangle$, and $\mathfrak{M}^* = \langle W^*, R^*, V^* \rangle$ be a coarsest filtration of $\mathfrak{M}$.

[nml:fil:euc:](#)
[thm:modal-closed-filt](#)

1. If $\mathfrak{M}$ is symmetric, so is $\mathfrak{M}^*$.

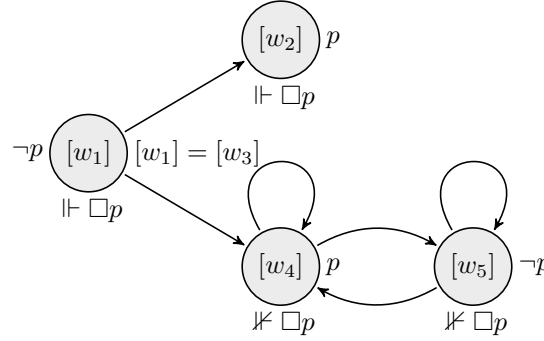


Figura 53.3: The filtration of the model in Figura 53.2.

nml:fl:euc:
fig:ser-eucl2

2. If $\mathfrak{M}$ is transitive, so is $\mathfrak{M}^*$.

3. If $\mathfrak{M}$ is euclidean, so is $\mathfrak{M}^*$.

Prova. 1. If $\mathfrak{M}^*$ is a coarsest filtration, then by definition $R^*[u][v]$ holds if and only if $C_1(u, v)$. For transitivity, suppose $C_1(u, v)$ and $C_1(v, w)$; we have to show $C_1(u, w)$. Suppose $\mathfrak{M}, u \Vdash \Box\varphi$; then $\mathfrak{M}, u \Vdash \Box\Box\varphi$ since 4 is valid in all transitive models; since $\Box\Box\varphi \in \Gamma$ by closure, also by $C_1(u, v)$, $\mathfrak{M}, v \Vdash \Box\varphi$ and by $C_1(v, w)$, also $\mathfrak{M}, w \Vdash \varphi$. Suppose $\mathfrak{M}, w \Vdash \varphi$; then $\mathfrak{M}, v \Vdash \Diamond\varphi$ by $C_1(v, w)$, since $\Diamond\varphi \in \Gamma$ by modal closure. By $C_1(u, v)$, we get $\mathfrak{M}, u \Vdash \Diamond\Diamond\varphi$ since $\Diamond\Diamond\varphi \in \Gamma$ by modal closure. Since $4_\Diamond$ is valid in all transitive models, $\mathfrak{M}, u \Vdash \Diamond\varphi$.

2. Exercise. Use the fact that both 5 and $5_\Diamond$ are valid in all euclidean models.

3. Exercise. Use the fact that B and $B_\Diamond$ are valid in all symmetric models.
□

Problema 53.7. Complete the proof of Teorema 53.19.

Capítulo 54

Modal Tableaux

Draft chapter on prefixed tableaux for modal logic. Needs more examples, completeness proofs, and discussion of how one can find countermodels from unsuccessful searches for closed tableaux.

-NoValue-

`content/normal-modal-logic/tableaux/introduction.tex`

54.1 Introduction

Tableaux are certain (downward-branching) trees of *fórmulas assinadas*, O.s., pairs consisting of a truth value sign ($\mathbb{T}$ or $\mathbb{F}$) and a *sentença*

nml:tab:int:
sec

$\mathbb{T} \varphi$ or $\mathbb{F} \varphi$.

A **tableau** begins with a number of *assumptions*. Each further *fórmula assinada* is generated by applying one of the inference rules. Some inference rules add one or more *fórmulas assinadas* to a tip of the tree; others add two new tips, resulting in two branches. Rules result in *fórmulas assinadas* where the *fórmula* is less complex than that of the *fórmula assinada* to which it was applied. When a branch contains both $\mathbb{T} \varphi$ and $\mathbb{F} \varphi$, we say the branch is *closed*. If every branch in a **tableau** is closed, the entire **tableau** is closed. A closed **tableau** constitutes a *derivação* that shows that the set of *fórmulas assinadas* which were used to begin the **tableau** are unsatisfiable. This can be used to define a $\vdash$ relation: $\Gamma \vdash \varphi$ iff there is some finite set $\Gamma_0 = \{\psi_1, \dots, \psi_n\} \subseteq \Gamma$ such that there is a closed **tableau** for the assumptions

$$\{\mathbb{F} \varphi, \mathbb{T} \psi_1, \dots, \mathbb{T} \psi_n\}.$$

For modal logics, we have to both extend the notion of *fórmula assinada* and add rules that cover $\Box$ and $\Diamond$. In addition to a sign($\mathbb{T}$ or $\mathbb{F}$), *fórmulas* in

54.2. RULES FOR K

$\frac{\sigma \mathbb{T} \neg \varphi}{\sigma \mathbb{F} \varphi} \neg \mathbb{T}$	$\frac{\sigma \mathbb{F} \neg \varphi}{\sigma \mathbb{T} \varphi} \neg \mathbb{F}$
$\frac{\sigma \mathbb{T} \varphi \wedge \psi}{\sigma \mathbb{T} \varphi \quad \sigma \mathbb{T} \psi} \wedge \mathbb{T}$	$\frac{\sigma \mathbb{F} \varphi \wedge \psi}{\sigma \mathbb{F} \varphi \quad \quad \sigma \mathbb{F} \psi} \wedge \mathbb{F}$
$\frac{\sigma \mathbb{T} \varphi \vee \psi}{\sigma \mathbb{T} \varphi \quad \quad \sigma \mathbb{T} \psi} \vee \mathbb{T}$	$\frac{\sigma \mathbb{F} \varphi \vee \psi}{\sigma \mathbb{F} \varphi \quad \sigma \mathbb{F} \psi} \vee \mathbb{F}$
$\frac{\sigma \mathbb{T} \varphi \rightarrow \psi}{\sigma \mathbb{F} \varphi \quad \quad \sigma \mathbb{T} \psi} \rightarrow \mathbb{T}$	$\frac{\sigma \mathbb{F} \varphi \rightarrow \psi}{\sigma \mathbb{T} \varphi \quad \sigma \mathbb{F} \psi} \rightarrow \mathbb{F}$

Tabela 54.1: Prefixed **tableau** rules for the propositional connectives

nml:tab:rul:
tab:prop-rules

modal **tableaux** also have *prefixes* σ . The prefixes are non-empty sequences of positive integers, O.s., $\sigma \in (\mathbb{Z}^+)^* \setminus \{\Lambda\}$. When we write such prefixes without the surrounding $\langle \rangle$, and separate the individual **membros** by .’s instead of ,’s. If σ is a prefix, then $\sigma.n$ is $\sigma \frown \langle n \rangle$; p.ex., if $\sigma = 1.2.1$, then $\sigma.3$ is 1.2.1.3. So for instance,

$$1.2 \mathbb{T} \Box \varphi \rightarrow \varphi$$

is a *prefixed fórmula assinada* (or just a *prefixed fórmula* for short).

Intuitively, the prefix names a world in a model that might satisfy the **fórmulas** on a branch of a **tableau**, and if σ names some world, then $\sigma.n$ names a world accessible from (the world named by) σ .

-NoValue-

content/normal-modal-logic/tableaux/rules-for-K.tex

54.2 Rules for K

nml:tab:rul:
sec

The rules for the regular propositional connectives are the same as for regular propositional signed **tableaux**, just with prefixes added. In each case, the rule applied to a signed **fórmula** $\sigma S \varphi$ produces new **fórmulas** that are also prefixed by σ . This should be intuitively clear: p.ex., if $\varphi \wedge \psi$ is true at (a world named by) σ , then φ and ψ are true at σ (and not at any other world). We collect the propositional rules in **Tabela 54.1**.

The closure condition is the same as for ordinary **tableaux**, although we require that not just the **fórmulas** but also the prefixes must match. So a

$\frac{\sigma \mathbb{T} \Box \varphi}{\sigma.n \mathbb{T} \varphi} \Box \mathbb{T}$ $\sigma.n$ is used	$\frac{\sigma \mathbb{F} \Box \varphi}{\sigma.n \mathbb{F} \varphi} \Box \mathbb{F}$ $\sigma.n$ is new
$\frac{\sigma \mathbb{T} \Diamond \varphi}{\sigma.n \mathbb{T} \varphi} \Diamond \mathbb{T}$ $\sigma.n$ is new	$\frac{\sigma \mathbb{F} \Diamond \varphi}{\sigma.n \mathbb{F} \varphi} \Diamond \mathbb{F}$ $\sigma.n$ is used

Tabela 54.2: The modal rules for K.

nml:tab:rul:
tab:rules-K

branch is closed if it contains both

$$\sigma \mathbb{T} \varphi \quad \text{and} \quad \sigma \mathbb{F} \varphi$$

for some prefix σ and **fórmula** φ .

The rules for setting up assumptions is also as for ordinary **tableaux**, except that for assumptions we always use the prefix 1. (It does not matter which prefix we use, as long as it's the same for all assumptions.) So, p.ex., we say that

$$\psi_1, \dots, \psi_n \vdash \varphi$$

iff there is a closed tableau for the assumptions

$$1 \mathbb{T} \psi_1, \dots, 1 \mathbb{T} \psi_n, 1 \mathbb{F} \varphi.$$

For the modal operators $\Box$ and $\Diamond$, the prefix of the conclusion of the rule applied to a **fórmula** with prefix σ is $\sigma.n$. However, which n is allowed depends on whether the sign is $\mathbb{T}$ or $\mathbb{F}$.

The $\Box \mathbb{T}$ rule extends a branch containing $\sigma \mathbb{T} \Box \varphi$ by $\sigma.n \mathbb{T} \varphi$. Similarly, the $\Diamond \mathbb{F}$ rule extends a branch containing $\sigma \mathbb{F} \Diamond \varphi$ by $\sigma.n \mathbb{F} \varphi$. They can only be applied for a prefix $\sigma.n$ which *already* occurs on the branch in which it is applied. Let's call such a prefix “used” (on the branch).

The $\Box \mathbb{F}$ rule extends a branch containing $\sigma \mathbb{F} \Box \varphi$ by $\sigma.n \mathbb{F} \varphi$. Similarly, the $\Diamond \mathbb{T}$ rule extends a branch containing $\sigma \mathbb{T} \Diamond \varphi$ by $\sigma.n \mathbb{T} \varphi$. These rules, however, can only be applied for a prefix $\sigma.n$ which *does not* already occur on the branch in which it is applied. We call such prefixes “new” (to the branch).

The rules are given in **Tabela 54.2**.

The requirement that the restriction that the prefix for $\Box \mathbb{T}$ must be used is necessary as otherwise we would count the following as a closed **tableau**:

54.3. TABLEAUX FOR K

1.	1 T $\Box\varphi$	Assumption
2.	1 F $\Diamond\varphi$	Assumption
3.	1.1 T φ	$\Box\text{T } 1$
4.	1.1 F φ	$\Diamond\text{F } 2$
	$\otimes$	

But $\Box\varphi \not\models \Diamond\varphi$, so our proof system would be unsound. Likewise, $\Diamond\varphi \not\models \Box\varphi$, but without the restriction that the prefix for $\Box\text{F}$ must be new, this would be a closed tableau:

1.	1 T $\Diamond\varphi$	Assumption
2.	1 F $\Box\varphi$	Assumption
3.	1.1 T φ	$\Diamond\text{T } 1$
4.	1.1 F φ	$\Box\text{F } 2$
	$\otimes$	

-NoValue-

content/normal-modal-logic/tableaux/proofs-in-K.tex

54.3 Tableaux for K

nml:tab:prk:
sec

Exemplo 54.1. We give a closed tableau that shows $\vdash (\Box\varphi \wedge \Box\psi) \rightarrow \Box(\varphi \wedge \psi)$.

1.	1 F $(\Box\varphi \wedge \Box\psi) \rightarrow \Box(\varphi \wedge \psi)$	Assumption
2.	1 T $\Box\varphi \wedge \Box\psi$	$\rightarrow\text{F } 1$
3.	1 F $\Box(\varphi \wedge \psi)$	$\rightarrow\text{F } 1$
4.	1 T $\Box\varphi$	$\wedge\text{T } 2$
5.	1 T $\Box\psi$	$\wedge\text{T } 2$
6.	1.1 F $\varphi \wedge \psi$	$\Box\text{F } 3$
	$\swarrow \quad \searrow$	
7.	1.1 F φ	$\wedge\text{F } 6$
8.	1.1 T φ	$\Box\text{T } 4; \Box\text{T } 5$
	$\otimes$	$\otimes$

Exemplo 54.2. We give a closed tableau that shows $\vdash \Diamond(\varphi \vee \psi) \rightarrow (\Diamond\varphi \vee \Diamond\psi)$:

1.	$1 \mathbb{F} \Diamond(\varphi \vee \psi) \rightarrow (\Diamond\varphi \vee \Diamond\psi)$	Assumption
2.	$1 \mathbb{T} \Diamond(\varphi \vee \psi)$	$\rightarrow \mathbb{F} 1$
3.	$1 \mathbb{F} \Diamond\varphi \vee \Diamond\psi$	$\rightarrow \mathbb{F} 1$
4.	$1 \mathbb{F} \Diamond\varphi$	$\vee \mathbb{F} 3$
5.	$1 \mathbb{F} \Diamond\psi$	$\vee \mathbb{F} 3$
6.	$1.1 \mathbb{T} \varphi \vee \psi$	$\Diamond \mathbb{T} 2$
7.	$1.1 \mathbb{T} \varphi$	$\vee \mathbb{T} 6$
8.	$1.1 \mathbb{F} \varphi$	$\Diamond \mathbb{F} 4; \Diamond \mathbb{F} 5$
	$\otimes$	$\otimes$

Problema 54.1. Find closed **tableaux** in **K** for the following **fórmulas**:

1. $\Box \neg p \rightarrow \Box(p \rightarrow q)$
2. $(\Box p \vee \Box q) \rightarrow \Box(p \vee q)$
3. $\Diamond p \rightarrow \Diamond(p \vee q)$
4. $\Box(p \wedge q) \rightarrow \Box p$

-NoValue-

[content/normal-modal-logic/tableaux/soundness.tex](#)

54.4 Soundness for **K**

[nml:tab:sou:](#)
[sec](#)

This soundness proof reuses the soundness proof for classical propositional logic, O.s., it proves everything from scratch. That's ok if you want a self-contained soundness proof. If you already have seen soundness for ordinary tableau this will be repetitive. It's planned to make it possible to switch between self-contained version and a version building on the non-modal case.

[explanation](#) In order to show that prefixed **tableaux** are sound, we have to show that if

$$1 \mathbb{T} \psi_1, \dots, 1 \mathbb{T} \psi_n, 1 \mathbb{F} \varphi$$

has a closed **tableau** then $\psi_1, \dots, \psi_n \models \varphi$. It is easier to prove the contrapositive: if for some $\mathfrak{M}$ and world w , $\mathfrak{M}, w \models \psi_i$ for all $i = 1, \dots, n$ but $\mathfrak{M}, w \not\models \varphi$, then no **tableau** can close. Such a countermodel shows that the initial assumptions of the **tableau** are satisfiable. The strategy of the proof is to show that whenever all the prefixed **fórmulas** on a **tableau** branch are satisfiable, any application of a rule results in at least one extended branch that

54.4. SOUNDNESS FOR K

is also satisfiable. Since closed branches are unsatisfiable, any **tableau** for a satisfiable set of prefixed **fórmulas** must have at least one open branch.

In order to apply this strategy in the modal case, we have to extend our definition of “satisfiable” to modal modals and prefixes. With that in hand, however, the proof is straightforward.

Definição 54.3. Let P be some set of prefixes, O.s., $P \subseteq (\mathbb{Z}^+)^* \setminus \{A\}$ and let $\mathfrak{M}$ be a model. A function $f: P \rightarrow W$ is an *interpretation of P in $\mathfrak{M}$* if, whenever σ and $\sigma.n$ are both in P , then $Rf(\sigma)f(\sigma.n)$.

Relative to an interpretation of prefixes P we can define:

1. $\mathfrak{M}$ satisfies $\sigma \mathbb{T} \varphi$ iff $\mathfrak{M}, f(\sigma) \Vdash \varphi$.
2. $\mathfrak{M}$ satisfies $\sigma \mathbb{F} \varphi$ iff $\mathfrak{M}, f(\sigma) \nVdash \varphi$.

Definição 54.4. Let Γ be a set of prefixed **fórmulas**, and let $P(\Gamma)$ be the set of prefixes that occur in it. If f is an interpretation of $P(\Gamma)$ in $\mathfrak{M}$, we say that $\mathfrak{M}$ satisfies Γ with respect to f , $\mathfrak{M}, f \Vdash \Gamma$, if $\mathfrak{M}$ satisfies every prefixed **fórmula** in Γ with respect to f . Γ is *satisfiable* iff there is a model $\mathfrak{M}$ and interpretation f of $P(\Gamma)$ such that $\mathfrak{M}, f \Vdash \Gamma$.

Proposição 54.5. If Γ contains both $\sigma \mathbb{T} \varphi$ and $\sigma \mathbb{F} \varphi$, for some **fórmula** φ and prefix σ , then Γ is unsatisfiable.

Prova. There cannot be a model $\mathfrak{M}$ and interpretation f of $P(\Gamma)$ such that both $\mathfrak{M}, f(\sigma) \Vdash \varphi$ and $\mathfrak{M}, f(\sigma) \nVdash \varphi$. $\square$

Teorema 54.6 (Soundness). If Γ has a closed **tableau**, Γ is unsatisfiable.

Prova. We call a branch of a **tableau** satisfiable iff the set of **fórmulas assinadas** on it is satisfiable, and let’s call a **tableau** satisfiable if it contains at least one satisfiable branch.

We show the following: Extending a satisfiable **tableau** by one of the rules of inference always results in a satisfiable **tableau**. This will prove the theorem: any closed **tableau** results by applying rules of inference to the **tableau** consisting only of assumptions from Γ . So if Γ were satisfiable, any **tableau** for it would be satisfiable. A closed **tableau**, however, is clearly not satisfiable, since all its branches are closed and closed branches are unsatisfiable.

Suppose we have a satisfiable **tableau**, O.s., a **tableau** with at least one satisfiable branch. Applying a rule of inference either adds **fórmulas assinadas** to a branch, or splits a branch in two. If the **tableau** has a satisfiable branch which is not extended by the rule application in question, it remains a satisfiable branch in the extended **tableau**, so the extended tableau is satisfiable. So we only have to consider the case where a rule is applied to a satisfiable branch.

Let Γ be the set of **fórmulas assinadas** on that branch, and let $\sigma S \varphi \in \Gamma$ be the **fórmula assinada** to which the rule is applied. If the rule does not result in a split branch, we have to show that the extended branch, O.s., Γ together with the conclusions of the rule, is still satisfiable. If the rule results in split branch,

we have to show that at least one of the two resulting branches is satisfiable. First, we consider the possible inferences with only one premise.

1. The branch is expanded by applying $\neg\mathbb{T}$ to $\sigma\mathbb{T}\neg\psi \in \Gamma$. Then the extended branch contains the *fórmulas assinadas* $\Gamma \cup \{\sigma\mathbb{F}\psi\}$. Suppose $\mathfrak{M}, f \Vdash \Gamma$. In particular, $\mathfrak{M}, f(\sigma) \Vdash \neg\psi$. Thus, $\mathfrak{M}, f(\sigma) \nVdash \psi$, O.s., $\mathfrak{M}$ satisfies $\sigma\mathbb{F}\psi$ with respect to f .
2. The branch is expanded by applying $\neg\mathbb{F}$ to $\sigma\mathbb{F}\neg\psi \in \Gamma$: Exercise.
3. The branch is expanded by applying $\wedge\mathbb{T}$ to $\sigma\mathbb{T}\psi \wedge \chi \in \Gamma$, which results in two new *fórmulas assinadas* on the branch: $\sigma\mathbb{T}\psi$ and $\sigma\mathbb{T}\chi$. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular $\mathfrak{M}, f(\sigma) \Vdash \psi \wedge \chi$. Then $\mathfrak{M}, f(\sigma) \Vdash \psi$ and $\mathfrak{M}, f(\sigma) \Vdash \chi$. This means that $\mathfrak{M}$ satisfies both $\sigma\mathbb{T}\psi$ and $\sigma\mathbb{T}\chi$ with respect to f .
4. The branch is expanded by applying $\vee\mathbb{F}$ to $\mathbb{F}\psi \vee \chi \in \Gamma$: Exercise.
5. The branch is expanded by applying $\rightarrow\mathbb{F}$ to $\sigma\mathbb{F}\psi \rightarrow \chi \in \Gamma$: This results in two new *fórmulas assinadas* on the branch: $\sigma\mathbb{T}\psi$ and $\sigma\mathbb{F}\chi$. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular $\mathfrak{M}, f(\sigma) \nVdash \psi \rightarrow \chi$. Then $\mathfrak{M}, f(\sigma) \Vdash \psi$ and $\mathfrak{M}, f(\sigma) \nVdash \chi$. This means that $\mathfrak{M}, f$ satisfies both $\sigma\mathbb{T}\psi$ and $\sigma\mathbb{F}\chi$.
6. The branch is expanded by applying $\Box\mathbb{T}$ to $\sigma\mathbb{T}\Box\psi \in \Gamma$: This results in a new *fórmula assinada* $\sigma.n\mathbb{T}\psi$ on the branch, for some $\sigma.n \in P(\Gamma)$ (since $\sigma.n$ must be used). Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma) \Vdash \Box\psi$. Since f is an interpretation of prefixes and both $\sigma, \sigma.n \in P(\Gamma)$, we know that $Rf(\sigma)f(\sigma.n)$. Hence, $\mathfrak{M}, f(\sigma.n) \Vdash \psi$, O.s., $\mathfrak{M}, f$ satisfies $\sigma.n\mathbb{T}\psi$.
7. The branch is expanded by applying $\Box\mathbb{F}$ to $\sigma\mathbb{F}\Box\psi \in \Gamma$: This results in a new *fórmula assinada* $\sigma.n\mathbb{F}\psi$, where $\sigma.n$ is a new prefix on the branch, O.s., $\sigma.n \notin P(\Gamma)$. Since Γ is satisfiable, there is a $\mathfrak{M}$ and interpretation f of $P(\Gamma)$ such that $\mathfrak{M}, f \models \Gamma$, in particular $\mathfrak{M}, f(\sigma) \nVdash \Box\psi$. We have to show that $\Gamma \cup \{\sigma.n\mathbb{F}\psi\}$ is satisfiable. To do this, we define an interpretation of $P(\Gamma) \cup \{\sigma.n\}$ as follows:

Since $\mathfrak{M}, f(\sigma) \nVdash \Box\psi$, there is a $w \in W$ such that $Rf(\sigma)w$ and $\mathfrak{M}, w \nVdash \psi$. Let f' be like f , except that $f'(\sigma.n) = w$. Since $f'(\sigma) = f(\sigma)$ and $Rf(\sigma)w$, we have $Rf'(\sigma)f'(\sigma.n)$, so f' is an interpretation of $P(\Gamma) \cup \{\sigma.n\}$. Obviously $\mathfrak{M}, f'(\sigma.n) \nVdash \psi$. Since $f(\sigma') = f'(\sigma')$ for all prefixes $\sigma' \in P(\Gamma)$, $\mathfrak{M}, f' \Vdash \Gamma$. So, $\mathfrak{M}, f'$ satisfies $\Gamma \cup \{\sigma.n\mathbb{F}\psi\}$.

Now let's consider the possible inferences with two premises.

1. The branch is expanded by applying $\wedge\mathbb{F}$ to $\sigma\mathbb{F}\psi \wedge \chi \in \Gamma$, which results in two branches, a left one continuing through $\sigma\mathbb{F}\psi$ and a right one through $\sigma\mathbb{F}\chi$. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular $\mathfrak{M}, f(\sigma) \nVdash \psi \wedge \chi$. Then $\mathfrak{M}, f(\sigma) \nVdash \psi$ or $\mathfrak{M}, f(\sigma) \nVdash \chi$. In the former case, $\mathfrak{M}, f$ satisfies $\sigma\mathbb{F}\psi$, O.s., the left branch is satisfiable. In the latter, $\mathfrak{M}, f$ satisfies $\sigma\mathbb{F}\chi$, O.s., the right branch is satisfiable.

54.5. RULES FOR OTHER ACCESSIBILITY RELATIONS

2. The branch is expanded by applying $\forall\mathbb{T}$ to $\sigma\mathbb{T}\psi \vee \chi \in \Gamma$: Exercise.
3. The branch is expanded by applying $\rightarrow\mathbb{T}$ to $\sigma\mathbb{T}\psi \rightarrow \chi \in \Gamma$: Exercise. $\square$

Problema 54.2. Complete the proof of **Teorema 54.6**.

*nml:tab:sou:
cor:entailment-soundness*

Corolário 54.7. *If $\Gamma \vdash \varphi$ then $\Gamma \models \varphi$.*

Prova. If $\Gamma \vdash \varphi$ then for some $\psi_1, \dots, \psi_n \in \Gamma$, $\Delta = \{1\mathbb{F}\varphi, 1\mathbb{T}\psi_1, \dots, 1\mathbb{T}\psi_n\}$ has a closed **tableau**. We want to show that $\Gamma \models \varphi$. Suppose not, so for some $\mathfrak{M}$ and w , $\mathfrak{M}, w \Vdash \psi_i$ for $i = 1, \dots, n$, but $\mathfrak{M}, w \not\models \varphi$. Let $f(1) = w$; then f is an interpretation of $P(\Delta)$ into $\mathfrak{M}$, and $\mathfrak{M}$ satisfies Δ with respect to f . But by **Teorema 54.6**, Δ is unsatisfiable since it has a closed **tableau**, a contradiction. So we must have $\Gamma \vdash \varphi$ after all. $\square$

*nml:tab:sou:
cor:weak-soundness*

Corolário 54.8. *If $\vdash \varphi$ then φ is true in all models.*

-NoValue-

content/normal-modal-logic/tableaux/more-rules.tex

54.5 Rules for Other Accessibility Relations

*nml:tab:mru:
sec*

In order to deal with logics determined by special accessibility relations, we consider the additional rules in **Tabela 54.3**.

Adding these rules results in systems that are sound and complete for the logics given in **Tabela 54.4**.

Exemplo 54.9. We give a closed tableau that shows **S5** $\vdash 5$, O.s., $\Box\varphi \rightarrow \Box\Diamond\varphi$.

1.	$1\mathbb{F} \Box\varphi \rightarrow \Box\Diamond\varphi$	Assumption
2.	$1\mathbb{T} \Box\varphi$	$\rightarrow\mathbb{F} 1$
3.	$1\mathbb{F} \Box\Diamond\varphi$	$\rightarrow\mathbb{F} 1$
4.	$1.1\mathbb{F} \Diamond\varphi$	$\Box\mathbb{F} 3$
5.	$1\mathbb{F} \Diamond\varphi$	$4r\Diamond 4$
6.	$1.1\mathbb{F} \varphi$	$\Diamond\mathbb{F} 5$
7.	$1.1\mathbb{T} \varphi$	$\Box\mathbb{T} 2$
	$\otimes$	

Problema 54.3. Give closed **tableaux** that show the following:

1. **KT5** $\vdash B$;
2. **KT5** $\vdash 4$;
3. **KDB4** $\vdash T$;

$\frac{\sigma \mathsf{T} \Box \varphi}{\sigma \mathsf{T} \varphi} \mathsf{T} \Box$	$\frac{\sigma \mathsf{F} \Diamond \varphi}{\sigma \mathsf{F} \varphi} \mathsf{T} \Diamond$
$\frac{\sigma \mathsf{T} \Box \varphi}{\sigma \mathsf{T} \Diamond \varphi} \mathsf{D} \Box$	$\frac{\sigma \mathsf{F} \Diamond \varphi}{\sigma \mathsf{F} \Box \varphi} \mathsf{D} \Diamond$
$\frac{\sigma.n \mathsf{T} \Box \varphi}{\sigma \mathsf{T} \varphi} \mathsf{B} \Box$	$\frac{\sigma.n \mathsf{F} \Diamond \varphi}{\sigma \mathsf{F} \varphi} \mathsf{B} \Diamond$
$\frac{\sigma \mathsf{T} \Box \varphi}{\sigma.n \mathsf{T} \Box \varphi} 4 \Box$ $\sigma.n$ is used	$\frac{\sigma \mathsf{F} \Diamond \varphi}{\sigma.n \mathsf{F} \Diamond \varphi} 4 \Diamond$ $\sigma.n$ is used
$\frac{\sigma.n \mathsf{T} \Box \varphi}{\sigma \mathsf{T} \Box \varphi} 4 \mathsf{r} \Box$	$\frac{\sigma.n \mathsf{F} \Diamond \varphi}{\sigma \mathsf{F} \Diamond \varphi} 4 \mathsf{r} \Diamond$

Tabela 54.3: More modal rules.

nml:tab:mru:
tab:more-rules

Logic	R is ...	Rules
T = KT	reflexive	$\mathsf{T} \Box, \mathsf{T} \Diamond$
D = KD	serial	$\mathsf{D} \Box, \mathsf{D} \Diamond$
K4	transitive	$4 \Box, 4 \Diamond$
B = KTB	reflexive, symmetric	$\mathsf{T} \Box, \mathsf{T} \Diamond$ $\mathsf{B} \Box, \mathsf{B} \Diamond$
S4 = KT4	reflexive, transitive	$\mathsf{T} \Box, \mathsf{T} \Diamond,$ $4 \Box, 4 \Diamond$
S5 = KT4B	reflexive, transitive, euclidean	$\mathsf{T} \Box, \mathsf{T} \Diamond,$ $4 \Box, 4 \Diamond,$ $4 \mathsf{r} \Box, 4 \mathsf{r} \Diamond$

Tabela 54.4: **Tableau** rules for various modal logics.

nml:tab:mru:
tab:logics-rules

54.6. SOUNDNESS FOR ADDITIONAL RULES

4. **KB4** $\vdash$ 5;

5. **KB5** $\vdash$ 4;

6. **KT** $\vdash$ D.

-NoValue-

content/normal-modal-logic/tableaux/more-soundness.tex

54.6 Soundness for Additional Rules

nml:tab:msn:sec We say a rule is sound for a class of models if, whenever a branch in a **tableau** is satisfiable in a model from that class, the branch resulting from applying the rule is also satisfiable in a model from that class.

nml:tab:msn:prop:soundness-T **Proposição 54.10.** $T\Box$ and $T\Diamond$ are sound for reflexive models.

Prova. 1. The branch is expanded by applying $T\Box$ to $\sigma T\Box\psi \in \Gamma$: This results in a new **fórmula assinada** $\sigma T\psi$ on the branch. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma) \Vdash \Box\psi$. Since R is reflexive, we know that $Rf(\sigma)f(\sigma)$. Hence, $\mathfrak{M}, f(\sigma) \Vdash \psi$, O.s., $\mathfrak{M}, f$ satisfies $\sigma T\psi$.

2. The branch is expanded by applying $T\Diamond$ to $\sigma F\Diamond\psi \in \Gamma$: This results in a new **fórmula assinada** $\sigma F\psi$ on the branch. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma) \nVdash \Diamond\psi$. Since R is reflexive, we know that $Rf(\sigma)f(\sigma)$. Hence, $\mathfrak{M}, f(\sigma) \nVdash \psi$, O.s., $\mathfrak{M}, f$ satisfies $\sigma F\psi$. $\square$

nml:tab:msn:prop:soundness-D **Proposição 54.11.** $D\Box$ and $D\Diamond$ are sound for serial models.

Prova. 1. The branch is expanded by applying $D\Box$ to $\sigma T\Box\psi \in \Gamma$: This results in a new **fórmula assinada** $\sigma T\Diamond\psi$ on the branch. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma) \Vdash \Box\psi$. Since R is serial, there is a $w \in W$ such that $Rf(\sigma)w$. Then $\mathfrak{M}, w \Vdash \psi$, and hence $\mathfrak{M}, f(\sigma) \Vdash \Diamond\psi$. So, $\mathfrak{M}, f$ satisfies $\sigma T\Diamond\psi$.

2. The branch is expanded by applying $D\Diamond$ to $\sigma F\Diamond\psi \in \Gamma$: This results in a new **fórmula assinada** $\sigma F\Box\psi$ on the branch. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma) \nVdash \Diamond\psi$. Since R is serial, there is a $w \in W$ such that $Rf(\sigma)w$. Then $\mathfrak{M}, w \nVdash \psi$, and hence $\mathfrak{M}, f(\sigma) \nVdash \Box\psi$. So, $\mathfrak{M}, f$ satisfies $\sigma F\Box\psi$. $\square$

nml:tab:msn:prop:soundness-B **Proposição 54.12.** $B\Box$ and $B\Diamond$ are sound for symmetric models.

- Prova.* 1. The branch is expanded by applying $B\Box$ to $\sigma.n\mathbb{T}\Box\psi \in \Gamma$: This results in a new **fórmula assinada** $\sigma\mathbb{T}\psi$ on the branch. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma.n) \Vdash \Box\psi$. Since f is an interpretation of prefixes on the branch into $\mathfrak{M}$, we know that $Rf(\sigma)f(\sigma.n)$. Since R is symmetric, $Rf(\sigma.n)f(\sigma)$. Since $\mathfrak{M}, f(\sigma.n) \Vdash \Box\psi$, $\mathfrak{M}, f(\sigma) \Vdash \psi$. Hence, $\mathfrak{M}, f$ satisfies $\sigma\mathbb{T}\psi$.
2. The branch is expanded by applying $B\Diamond$ to $\sigma.n\mathbb{F}\Diamond\psi \in \Gamma$: This results in a new **fórmula assinada** $\sigma\mathbb{F}\psi$ on the branch. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma.n) \nVdash \Diamond\psi$. Since f is an interpretation of prefixes on the branch into $\mathfrak{M}$, we know that $Rf(\sigma)f(\sigma.n)$. Since R is symmetric, $Rf(\sigma.n)f(\sigma)$. Since $\mathfrak{M}, f(\sigma.n) \nVdash \Diamond\psi$, $\mathfrak{M}, f(\sigma) \nVdash \psi$. Hence, $\mathfrak{M}, f$ satisfies $\sigma\mathbb{F}\psi$. $\square$

Proposição 54.13. $4\Box$ and $4\Diamond$ are sound for transitive models.

*nml:tab:msn:
prop:soundness-4*

- Prova.* 1. The branch is expanded by applying $4\Box$ to $\sigma\mathbb{T}\Box\psi \in \Gamma$: This results in a new **fórmula assinada** $\sigma.n\mathbb{T}\Box\psi$ on the branch. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma) \Vdash \Box\psi$. Since f is an interpretation of prefixes on the branch into $\mathfrak{M}$ and $\sigma.n$ must be used, we know that $Rf(\sigma)f(\sigma.n)$. Now let w be any world such that $Rf(\sigma.n)w$. Since R is transitive, $Rf(\sigma)w$. Since $\mathfrak{M}, f(\sigma) \Vdash \Box\psi$, $\mathfrak{M}, w \Vdash \psi$. Hence, $\mathfrak{M}, f(\sigma.n) \Vdash \Box\psi$, and $\mathfrak{M}, f$ satisfies $\sigma.n\mathbb{T}\Box\psi$.
2. The branch is expanded by applying $4\Diamond$ to $\sigma\mathbb{F}\Diamond\psi \in \Gamma$: This results in a new **fórmula assinada** $\sigma.n\mathbb{F}\Diamond\psi$ on the branch. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma) \nVdash \Diamond\psi$. Since f is an interpretation of prefixes on the branch into $\mathfrak{M}$ and $\sigma.n$ must be used, we know that $Rf(\sigma)f(\sigma.n)$. Now let w be any world such that $Rf(\sigma.n)w$. Since R is transitive, $Rf(\sigma)w$. Since $\mathfrak{M}, f(\sigma) \nVdash \Diamond\psi$, $\mathfrak{M}, w \nVdash \psi$. Hence, $\mathfrak{M}, f(\sigma.n) \nVdash \Diamond\psi$, and $\mathfrak{M}, f$ satisfies $\sigma.n\mathbb{F}\Diamond\psi$. $\square$

Proposição 54.14. $4r\Box$ and $4r\Diamond$ are sound for euclidean models.

*nml:tab:msn:
prop:soundness-4r*

- Prova.* 1. The branch is expanded by applying $4r\Box$ to $\sigma.n\mathbb{T}\Box\psi \in \Gamma$: This results in a new **fórmula assinada** $\sigma\mathbb{T}\Box\psi$ on the branch. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma.n) \Vdash \Box\psi$. Since f is an interpretation of prefixes on the branch into $\mathfrak{M}$, we know that $Rf(\sigma)f(\sigma.n)$. Now let w be any world such that $Rf(\sigma)w$. Since R is euclidean, $Rf(\sigma.n)w$. Since $\mathfrak{M}, f(\sigma.n) \Vdash \Box\psi$, $\mathfrak{M}, w \Vdash \psi$. Hence, $\mathfrak{M}, f(\sigma) \Vdash \Box\psi$, and $\mathfrak{M}, f$ satisfies $\sigma\mathbb{T}\Box\psi$.
2. The branch is expanded by applying $4r\Diamond$ to $\sigma.n\mathbb{F}\Diamond\psi \in \Gamma$: This results in a new **fórmula assinada** $\sigma\mathbb{T}\Box\psi$ on the branch. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular, $\mathfrak{M}, f(\sigma.n) \nVdash \Diamond\psi$. Since f is an interpretation of prefixes on the branch into $\mathfrak{M}$, we know that $Rf(\sigma)f(\sigma.n)$. Now let w be any world such

54.7. SIMPLE TABLEAUX FOR S5

$\frac{n \top \Box \varphi}{m \top \varphi} \Box \top$ m is used	$\frac{n \top \Box \varphi}{m \top \varphi} \Box \top$ m is new
$\frac{n \top \Diamond \varphi}{m \top \varphi} \Diamond \top$ m is new	$\frac{n \top \Diamond \varphi}{m \top \varphi} \Diamond \top$ m is used

Tabela 54.5: Simplified rules for **S5**.

nml:tab:s5:
tab:rules-S5

that $Rf(\sigma)w$. Since R is euclidean, $Rf(\sigma.n)w$. Since $\mathfrak{M}, f(\sigma).n \not\models \Diamond\psi$, $\mathfrak{M}, w \not\models \psi$. Hence, $\mathfrak{M}, f(\sigma) \not\models \Diamond\psi$, and $\mathfrak{M}, f$ satisfies $\sigma \models \Diamond\psi$. $\square$

nml:tab:msn:
cor:soundness-logics

Corolário 54.15. *The **tableau** systems given in **Tabela 54.4** are sound for the respective classes of models.*

-NoValue-

content/normal-modal-logic/tableaux/simple-S5.tex

54.7 Simple Tableaux for S5

nml:tab:s5:
sec

S5 is sound and complete with respect to the class of universal models, O.s., models where every world is accessible from every world. In universal models the accessibility relation doesn't matter: "there is a world w where $\mathfrak{M}, w \models \varphi$ " is true if and only if there is such a w that's accessible from u . So in **S5**, we can define models as simply a set of worlds and a valuation V . This suggests that we should be able to simplify the **tableau** rules as well. In the general case, we take as prefixes sequences of positive integers, so that we can keep track of which such prefixes name worlds which are accessible from others: $\sigma.n$ names a world accessible from σ . But in **S5** any world is accessible from any world, so there is no need to so keep track. Instead, we can use positive integers as prefixes. The simplified rules are given in **Tabela 54.5**.

Exemplo 54.16. We give a simplified closed tableau that shows **S5** $\vdash 5$, O.s., $\Diamond\varphi \rightarrow \Box\Diamond\varphi$.

1.	$1\mathbb{F} \Diamond\varphi \rightarrow \Box\Diamond\varphi$	Assumption
2.	$1\mathbb{T} \Diamond\varphi$	$\rightarrow\mathbb{F} 1$
3.	$1\mathbb{F} \Box\Diamond\varphi$	$\rightarrow\mathbb{F} 1$
4.	$2\mathbb{F} \Diamond\varphi$	$\Box\mathbb{F} 3$
5.	$3\mathbb{T} \varphi$	$\Diamond\mathbb{T} 2$
6.	$3\mathbb{F} \varphi$	$\Diamond\mathbb{F} 4$
	$\otimes$	

-NoValue-

content/normal-modal-logic/tableaux/completeness.tex

54.8 Completeness for K

explanation To show that the method of **tableaux** is complete, we have to show that whenever there is no closed **tableau** to show $\Gamma \vdash \varphi$, then $\Gamma \not\models \varphi$. O.s., there is a countermodel. But “there is no closed **tableau**” means that every way we could try to construct one has to fail to close. The trick is to see that if every such way fails to close, then a specific, *systematic and exhaustive* way also fails to close. And this systematic and exhaustive way would close if a closed **tableau** exists. The single tableau will contain, among its open branches, all the information required to define a countermodel. The countermodel given by an open branch in this tableau will contain the all the prefixes used on that branch as the worlds, and a **variável proposicional** p is true at σ iff $\sigma\mathbb{T}p$ occurs on the branch. nml:tab:cpl:sec

Definição 54.17. A branch in a **tableau** is called complete if, whenever it contains a prefixed **fórmula** $\sigma S\varphi$ to which a rule can be applied, it also contains

1. the prefixed **fórmulas** that are the corresponding conclusions of the rule, in the case of propositional stacking rules;
2. one of the corresponding conclusion **fórmulas** in the case of propositional branching rules;
3. at least one possible conclusion in the case of modal rules that require a new prefix;
4. the corresponding conclusion for every prefix occurring on the branch in the case of modal rules that require a used prefix.

explanation For instance, a complete branch contains $\sigma\mathbb{T}\psi$ and $\sigma\mathbb{T}\chi$ whenever it contains $\mathbb{T}\psi \wedge \chi$. If it contains $\sigma\mathbb{T}\psi \vee \chi$ it contains at least one of $\sigma\mathbb{F}\psi$ and $\sigma\mathbb{T}\chi$. If it contains $\sigma\mathbb{F}\Box$ it also contains $\sigma.n\mathbb{F}\Box$ for at least one n . And whenever it contains $\sigma\mathbb{T}\Box$ it also contains $\sigma.n\mathbb{T}\Box$ for every n such that $\sigma.n$ is used on the branch.

54.8. COMPLETENESS FOR K

nml:tab:cpl:
prop:complete-tableau

Proposição 54.18. *Every finite Γ has a tableau in which every branch is complete.*

Prova. Consider an open branch in a tableau for Γ . There are finitely many prefixed fórmulas in the branch to which a rule could be applied. In some fixed order (say, top to bottom), for each of these prefixed fórmulas for which the conditions (1)–(4) do not already hold, apply the rules that can be applied to it to extend the branch. In some cases this will result in branching; apply the rule at the tip of each resulting branch for all remaining prefixed fórmulas. Since the number of prefixed fórmulas is finite, and the number of used prefixes on the branch is finite, this procedure eventually results in (possibly many) branches extending the original branch. Apply the procedure to each, and repeat. But by construction, every branch is closed. $\square$

nml:tab:cpl:
thm:tableau-completeness

Teorema 54.19 (Completeness). *If Γ has no closed tableau, Γ is satisfiable.*

Prova. By the proposition, Γ has a tableau in which every branch is complete. Since it has no closed tableau, it has a tableau in which at least one branch is open and complete. Let Δ be the set of prefixed fórmulas on the branch, and $P(\Delta)$ the set of prefixes occurring in it.

We define a model $\mathfrak{M}(\Delta) = \langle P(\Delta), R, V \rangle$ where the worlds are the prefixes occurring in Δ , the accessibility relation is given by:

$$R\sigma\sigma' \text{ iff } \sigma' = \sigma.n \text{ for some } n$$

and

$$V(p) = \{\sigma : \sigma \mathbb{T} p \in \Delta\}.$$

We show by induction on φ that if $\sigma \mathbb{T} \varphi \in \Delta$ then $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$, and if $\sigma \mathbb{F} \varphi \in \Delta$ then $\mathfrak{M}(\Delta), \sigma \nVdash \varphi$.

1. $\varphi \equiv p$: If $\sigma \mathbb{T} \varphi \in \Delta$ then $\sigma \in V(p)$ (by definition of V) and so $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$.
If $\sigma \mathbb{F} \varphi \in \Delta$ then $\sigma \mathbb{T} \varphi \notin \Delta$, since the branch would otherwise be closed. So $\sigma \notin V(p)$ and thus $\mathfrak{M}(\Delta), \sigma \nVdash \varphi$.
2. $\varphi \equiv \neg\psi$: If $\sigma \mathbb{T} \varphi \in \Delta$, then $\sigma \mathbb{F} \psi \in \Delta$ since the branch is complete. By induction hypothesis, $\mathfrak{M}(\Delta), \sigma \nVdash \psi$ and thus $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$.
If $\sigma \mathbb{F} \varphi \in \Delta$, then $\sigma \mathbb{T} \psi \in \Delta$ since the branch is complete. By induction hypothesis, $\mathfrak{M}(\Delta), \sigma \Vdash \psi$ and thus $\mathfrak{M}(\Delta), \sigma \nVdash \varphi$.
3. $\varphi \equiv \psi \wedge \chi$: If $\sigma \mathbb{T} \varphi \in \Delta$, then both $\sigma \mathbb{T} \psi \in \Delta$ and $\sigma \mathbb{T} \chi \in \Delta$ since the branch is complete. By induction hypothesis, $\mathfrak{M}(\Delta), \sigma \Vdash \psi$ and $\mathfrak{M}(\Delta), \sigma \Vdash \chi$. Thus $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$.
If $\sigma \mathbb{F} \varphi \in \Delta$, then either $\sigma \mathbb{F} \psi \in \Delta$ or $\sigma \mathbb{F} \chi \in \Delta$ since the branch is complete. By induction hypothesis, either $\mathfrak{M}(\Delta), \sigma \nVdash \psi$ or $\mathfrak{M}(\Delta), \sigma \nVdash \chi$. Thus $\mathfrak{M}(\Delta), \sigma \nVdash \varphi$.

4. $\varphi \equiv \psi \vee \chi$: If $\sigma \mathbb{T} \varphi \in \Delta$, then either $\sigma \mathbb{T} \psi \in \Delta$ or $\sigma \mathbb{T} \chi \in \Delta$ since the branch is complete. By induction hypothesis, either $\mathfrak{M}(\Delta), \sigma \Vdash \psi$ or $\mathfrak{M}(\Delta), \sigma \Vdash \chi$. Thus $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$.

If $\sigma \mathbb{F} \varphi \in \Delta$, then both $\sigma \mathbb{F} \psi \in \Delta$ and $\sigma \mathbb{F} \chi \in \Delta$ since the branch is complete. By induction hypothesis, both $\mathfrak{M}(\Delta), \sigma \nVdash \psi$ and $\mathfrak{M}(\Delta), \sigma \nVdash \chi$. Thus $\mathfrak{M}(\Delta), \sigma \nVdash \varphi$.

5. $\varphi \equiv \psi \rightarrow \chi$: If $\sigma \mathbb{T} \varphi \in \Delta$, then either $\sigma \mathbb{F} \psi \in \Delta$ or $\sigma \mathbb{T} \chi \in \Delta$ since the branch is complete. By induction hypothesis, either $\mathfrak{M}(\Delta), \sigma \nVdash \psi$ or $\mathfrak{M}(\Delta), \sigma \Vdash \chi$. Thus $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$.

If $\sigma \mathbb{F} \varphi \in \Delta$, then both $\sigma \mathbb{T} \psi \in \Delta$ and $\sigma \mathbb{F} \chi \in \Delta$ since the branch is complete. By induction hypothesis, both $\mathfrak{M}(\Delta), \sigma \Vdash \psi$ and $\mathfrak{M}(\Delta), \sigma \nVdash \chi$. Thus $\mathfrak{M}(\Delta), \sigma \nVdash \varphi$.

6. $\varphi \equiv \Box \psi$: If $\sigma \mathbb{T} \varphi \in \Delta$, then, since the branch is complete, $\sigma.n \mathbb{T} \psi \in \Delta$ for every $\sigma.n$ used on the branch, O.s., for every $\sigma' \in P(\Delta)$ such that $R\sigma\sigma'$. By induction hypothesis, $\mathfrak{M}(\Delta), \sigma' \Vdash \psi$ for every σ' such that $R\sigma\sigma'$. Therefore, $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$.

If $\sigma \mathbb{F} \varphi \in \Delta$, then for some $\sigma.n$, $\sigma.n \mathbb{F} \psi \in \Delta$ since the branch is complete. By induction hypothesis, $\mathfrak{M}(\Delta), \sigma.n \nVdash \psi$. Since $R\sigma(\sigma.n)$, there is a σ' such that $\mathfrak{M}(\Delta), \sigma' \nVdash \psi$. Thus $\mathfrak{M}(\Delta), \sigma \nVdash \varphi$.

7. $\varphi \equiv \Diamond \psi$: If $\sigma \mathbb{T} \varphi \in \Delta$, then for some $\sigma.n$, $\sigma.n \mathbb{T} \psi \in \Delta$ since the branch is complete. By induction hypothesis, $\mathfrak{M}(\Delta), \sigma.n \Vdash \psi$. Since $R\sigma(\sigma.n)$, there is a σ' such that $\mathfrak{M}(\Delta), \sigma' \Vdash \psi$. Thus $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$.

If $\sigma \mathbb{F} \varphi \in \Delta$, then, since the branch is complete, $\sigma.n \mathbb{F} \psi \in \Delta$ for every $\sigma.n$ used on the branch, O.s., for every $\sigma' \in P(\Delta)$ such that $R\sigma\sigma'$. By induction hypothesis, $\mathfrak{M}(\Delta), \sigma' \nVdash \psi$ for every σ' such that $R\sigma\sigma'$. Therefore, $\mathfrak{M}(\Delta), \sigma \nVdash \varphi$.

Since $\Gamma \subseteq \Delta$, $\mathfrak{M}(\Delta) \Vdash \Gamma$. □

Problema 54.4. Complete the proof of [Teorema 54.19](#).

Corolário 54.20. If $\Gamma \models \varphi$ then $\Gamma \vdash \varphi$.

*nml:tab:cpl:
cor:entailment-completeness*

Corolário 54.21. If φ is true in all models, then $\vdash \varphi$.

*nml:tab:cpl:
cor:weak-completeness*

-NoValue-

54.9 Countermodels from Tableaux

nml:tab:cou:
sec

The proof of the completeness theorem doesn't just show that if $\models \varphi$ then $\vdash \varphi$, [explanation](#) it also gives us a method for constructing countermodels to φ if $\not\models A$. In the case of **K**, this method constitutes a *decision procedure*. For suppose $\not\models \varphi$. Then the proof of [Proposição 54.18](#) gives a method for constructing a complete [tableau](#). The method in fact always terminates. The propositional rules for **K** only add prefixed [fórmulas](#) of lower complexity, O.s., each propositional rule need only be applied once on a branch for any signed formula $\sigma S\varphi$. New prefixes are only generated by the $\Box F$ and $\Diamond T$ rules, and also only have to be applied once (and produce a single new prefix). $\Box T$ and $\Diamond F$ have to be applied potentially multiple times, but only once per prefix, and only finitely many new prefixes are generated. So the construction either results in a closed branch or a complete branch after finitely many stages.

Once a tableau with an open complete branch is constructed, the proof of [Teorema 54.19](#) gives us an explicit model that satisfies the original set of prefixed [fórmulas](#). So not only is it the case that if $\Gamma \models \varphi$, then a closed [tableau](#) exists and $\Gamma \vdash \varphi$, if we look for the closed [tableau](#) in the right way and end up with a “complete” [tableau](#), we'll not only know that $\Gamma \not\models \varphi$ but actually be able to construct a countermodel.

Exemplo 54.22. We know that $\not\models \Box(p \vee q) \rightarrow (\Box p \vee \Box q)$. The construction of a tableau begins with:

1.	$1 F \quad \Box(p \vee q) \rightarrow (\Box p \vee \Box q) \checkmark$	Assumption
2.	$1 T \quad \Box(p \vee q)$	$\rightarrow F 1$
3.	$1 F \quad \Box p \vee \Box q \checkmark$	$\rightarrow F 1$
4.	$1 F \quad \Box p \checkmark$	$\vee F 3$
5.	$1 F \quad \Box q \checkmark$	$\vee F 3$
6.	$1.1 F \quad p \checkmark$	$\Box F 4$
7.	$1.2 F \quad q \checkmark$	$\Box F 5$

The [tableau](#) is of course not finished yet. In the next step, we consider the only line without a checkmark: the prefixed [fórmula](#) $1 T \Box(p \vee q)$ on line 2. The construction of the closed tableau says to apply the $\Box T$ rule for every prefix used on the branch, O.s., for both 1.1 and 1.2:

1.	$1 F \quad \Box(p \vee q) \rightarrow (\Box p \vee \Box q) \checkmark$	Assumption
2.	$1 T \quad \Box(p \vee q)$	$\rightarrow F 1$
3.	$1 F \quad \Box p \vee \Box q \checkmark$	$\rightarrow F 1$
4.	$1 F \quad \Box p \checkmark$	$\vee F 3$
5.	$1 F \quad \Box q \checkmark$	$\vee F 3$
6.	$1.1 F \quad p \checkmark$	$\Box F 4$
7.	$1.2 F \quad q \checkmark$	$\Box F 5$
8.	$1.1 T \quad p \vee q$	$\Box T 2$
9.	$1.2 T \quad p \vee q$	$\Box T 2$

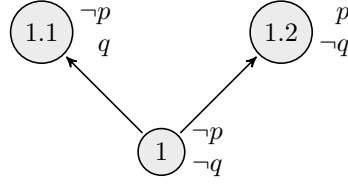


Figura 54.1: A countermodel to $\Box(p \vee q) \rightarrow (\Box p \vee \Box q)$.

nml:tab:cou:
fig:counter-Box

Now lines 2, 8, and 9, don't have checkmarks. But no new prefix has been added, so we apply $\vee\mathbb{T}$ to lines 8 and 9, on all resulting branches (as long as they don't close):

1.	$1 \mathbb{F} \Box(p \vee q) \rightarrow (\Box p \vee \Box q) \checkmark$	Assumption
2.	$1 \mathbb{T} \Box(p \vee q) \checkmark$	$\rightarrow \mathbb{F} 1$
3.	$1 \mathbb{F} \Box p \vee \Box q \checkmark$	$\rightarrow \mathbb{F} 1$
4.	$1 \mathbb{F} \Box p \checkmark$	$\vee \mathbb{F} 3$
5.	$1 \mathbb{F} \Box q \checkmark$	$\vee \mathbb{F} 3$
6.	$1.1 \mathbb{F} p \checkmark$	$\Box \mathbb{F} 4$
7.	$1.2 \mathbb{F} q \checkmark$	$\Box \mathbb{F} 5$
8.	$1.1 \mathbb{T} p \vee q \checkmark$	$\Box \mathbb{T} 2$
9.	$1.2 \mathbb{T} p \vee q \checkmark$	$\Box \mathbb{T} 2$
10.	$1.1 \mathbb{T} p \checkmark$ $1.1 \mathbb{T} q \checkmark$ $\otimes$	$\vee \mathbb{T} 8$
11.	$1.2 \mathbb{T} p \checkmark$ $1.2 \mathbb{T} q \checkmark$ $\otimes$	$\vee \mathbb{T} 9$

There is one remaining open branch, and it is complete. From it we define the model with worlds $W = \{1, 1.1, 1.2\}$ (the only prefixes appearing on the open branch), the accessibility relation $R = \{\langle 1, 1.1 \rangle, \langle 1, 1.2 \rangle\}$, and the assignment $V(p) = \{1.2\}$ (because line 11 contains $1.2 \mathbb{T} p$) and $V(q) = \{1.1\}$ (because line 10 contains $1.1 \mathbb{T} q$). The model is pictured in **Figura 54.1**, and you can verify that it is a countermodel to $\Box(p \vee q) \rightarrow (\Box p \vee \Box q)$.

Capítulo 55

Modal Sequent Calculus

Draft chapter on sequent calculi for modal logic. Needs more examples, soundness and completeness proofs.

-NoValue-

`content/normal-modal-logic/sequent-calculus/introduction.tex`

55.1 Introduction

nml:seq:int:
sec The sequent calculus for propositional logic can be extended by additional rules that deal with $\Box$ and $\Diamond$. For instance, for **K**, we have **LK** plus:

$$\frac{\Gamma \Rightarrow \Delta, \varphi}{\Box \Gamma \Rightarrow \Diamond \Delta, \Box \varphi} \Box \quad \frac{\varphi, \Gamma \Rightarrow \Delta}{\Diamond \varphi, \Box \Gamma \Rightarrow \Diamond \Delta} \Diamond$$

For extensions of **K**, additional rules have to be added as well.

Not every modal logic has such a sequent calculus. Even **S5**, which is semantically simple (it can be defined without using accessibility relations at all) is not known to have a sequent calculus that results from **LK** which is complete without the rule Cut. However, it has a cut-free complete *hypersequent* calculus.

-NoValue-

`content/normal-modal-logic/sequent-calculus/rules-for-K.tex`

55.2 Rules for K

nml:seq:rul:
sec The rules for the regular propositional connectives are the same as for regular sequent calculus **LK**. Axioms are also the same: any sequent of the form $\varphi \Rightarrow \varphi$ counts as an axiom.

For the modal operators $\Box$ and $\Diamond$, we have the following additional rules:

$$\frac{\Gamma \Rightarrow \Delta, \varphi}{\Box \Gamma \Rightarrow \Diamond \Delta, \Box \varphi} \Box \quad \frac{\varphi, \Gamma \Rightarrow \Delta}{\Diamond \varphi, \Box \Gamma \Rightarrow \Diamond \Delta} \Diamond$$

Here, $\Box \Gamma$ means the sequence of **fórmulas** resulting from Γ by putting $\Box$ in front of every **fórmula** in Γ and $\Diamond \Delta$ is the sequence of **fórmulas** resulting from Δ by putting $\Diamond$ in front of every **fórmula** in Δ . Γ and Δ may be empty; in that case the corresponding part $\Box \Gamma$ and $\Diamond \Delta$ of the conclusion sequent is empty as well.

The restriction of adding a $\Box$ on the right and $\Diamond$ on the left to a single **fórmula** φ is necessary. If we allowed to add $\Box$ to any number of **fórmulas** on the right or to add $\Diamond$ to any number of **fórmulas** on the left we would be able to **deriva**:

$$\frac{\frac{\varphi \Rightarrow \varphi}{\Rightarrow \varphi, \neg \varphi} \neg R \quad \frac{\Rightarrow \varphi, \Box \neg \varphi}{\Rightarrow \Box \varphi, \Box \neg \varphi} \Box *}{\Rightarrow \Box \varphi \vee \Box \neg \varphi} \vee R \quad \frac{\frac{\varphi \Rightarrow \varphi}{\neg \varphi, \varphi \Rightarrow} \neg L \quad \frac{\Diamond \neg \varphi, \Diamond \varphi \Rightarrow}{\Diamond \varphi \Rightarrow \neg \Diamond \neg \varphi} \neg R}{\Rightarrow \Diamond \varphi \rightarrow \neg \Diamond \neg \varphi} \rightarrow R$$

But $\Box \varphi \vee \Box \neg \varphi$ and $\Diamond \varphi \rightarrow \neg \Diamond \neg \varphi$ are not valid in **K**.

If we allowed side formulas in addition to φ in the premise, and allowed the $\Box$ rule to add $\Box$ to only φ on the right, or allowed the $\Diamond$ rule to add $\Diamond$ to only φ on the left (but do nothing to the side formulas) we would be able to **deriva**:

$$\frac{\frac{\varphi \Rightarrow \varphi}{\Rightarrow \varphi, \neg \varphi} \neg R \quad \frac{\Rightarrow \neg \varphi, \varphi}{\Rightarrow \neg \varphi, \Box \varphi} XR \quad \frac{\Rightarrow \neg \varphi, \Box \varphi}{\Rightarrow \neg \varphi \vee \Box \varphi} \vee R}{\Rightarrow \neg \varphi \vee \Box \varphi} \vee R \quad \frac{\frac{\varphi \Rightarrow \varphi}{\neg \varphi, \varphi \Rightarrow} \neg L \quad \frac{\Diamond \neg \varphi, \varphi \Rightarrow}{\varphi \Rightarrow \neg \Diamond \neg \varphi} \neg R}{\Rightarrow \varphi \rightarrow \neg \Diamond \neg \varphi} \rightarrow R$$

But $\neg \varphi \vee \Box \varphi$ (which is equivalent to $\varphi \rightarrow \Box \varphi$) and $\varphi \rightarrow \neg \Diamond \neg \varphi$ are not valid in **K**.

-NoValue-

content/normal-modal-logic/sequent-calculus/proofs-in-K.tex

55.3 Sequent Derivações for K

Exemplo 55.1. We give a sequent calculus **derivação** that shows $\vdash (\Box \varphi \wedge \Box \psi) \rightarrow \Box(\varphi \wedge \psi)$.

nml:seq:prk:
sec

55.3. SEQUENT DERIVAÇÕES FOR **K**

$$\begin{array}{c}
\frac{\varphi \Rightarrow \varphi}{\psi, \varphi \Rightarrow \varphi} \quad \frac{\psi \Rightarrow \psi}{\psi, \varphi \Rightarrow \psi} \quad \wedge R \\
\frac{\psi, \varphi \Rightarrow \varphi \wedge \psi}{\Box \psi, \Box \varphi \Rightarrow \Box(\varphi \wedge \psi)} \Box \\
\frac{\Box \psi, \Box \varphi \Rightarrow \Box(\varphi \wedge \psi)}{\Box \varphi \wedge \Box \psi, \Box \varphi \Rightarrow \Box(\varphi \wedge \psi)} \wedge L \\
\frac{\Box \varphi, \Box \varphi \wedge \Box \psi \Rightarrow \Box(\varphi \wedge \psi)}{\Box \varphi, \Box \varphi \wedge \Box \psi \Rightarrow \Box(\varphi \wedge \psi)} XL \\
\frac{\Box \varphi \wedge \Box \psi, \Box \varphi \wedge \Box \psi \Rightarrow \Box(\varphi \wedge \psi)}{\Box \varphi \wedge \Box \psi, \Box \varphi \wedge \Box \psi \Rightarrow \Box(\varphi \wedge \psi)} \wedge L \\
\frac{\Box \varphi \wedge \Box \psi \Rightarrow \Box(\varphi \wedge \psi)}{\Box \varphi \wedge \Box \psi \Rightarrow \Box(\varphi \wedge \psi)} CL \\
\frac{}{\Rightarrow (\Box \varphi \wedge \Box \psi) \rightarrow \Box(\varphi \wedge \psi)} \rightarrow R
\end{array}$$

Exemplo 55.2. We give a sequent calculus *derivação* that shows $\vdash \Diamond(\varphi \vee \psi) \rightarrow (\Diamond \varphi \vee \Diamond \psi)$.

$$\begin{array}{c}
\frac{\varphi \Rightarrow \varphi}{\varphi \Rightarrow \varphi, \psi} \quad \frac{\psi \Rightarrow \psi}{\psi \Rightarrow \varphi, \psi} \quad \vee L \\
\frac{\varphi \vee \psi \Rightarrow \varphi, \psi}{\Diamond(\varphi \vee \psi) \Rightarrow \Diamond \varphi, \Diamond \psi} \Diamond \\
\frac{\Diamond(\varphi \vee \psi) \Rightarrow \Diamond \varphi, \Diamond \psi}{\Diamond(\varphi \vee \psi) \Rightarrow \Diamond \varphi, \Diamond \varphi \vee \Diamond \psi} \vee R \\
\frac{\Diamond(\varphi \vee \psi) \Rightarrow \Diamond \varphi, \Diamond \varphi \vee \Diamond \psi}{\Diamond(\varphi \vee \psi) \Rightarrow \Diamond \varphi \vee \Diamond \psi, \Diamond \varphi} XR \\
\frac{\Diamond(\varphi \vee \psi) \Rightarrow \Diamond \varphi \vee \Diamond \psi, \Diamond \varphi \vee \Diamond \psi}{\Diamond(\varphi \vee \psi) \Rightarrow \Diamond \varphi \vee \Diamond \psi} \vee R \\
\frac{\Diamond(\varphi \vee \psi) \Rightarrow \Diamond \varphi \vee \Diamond \psi}{\Diamond(\varphi \vee \psi) \Rightarrow \Diamond \varphi \vee \Diamond \psi} CR \\
\frac{}{\Rightarrow \Diamond(\varphi \vee \psi) \rightarrow (\Diamond \varphi \vee \Diamond \psi)} \rightarrow R
\end{array}$$

Here is a *derivação* of DUAL.

$$\begin{array}{c}
\frac{\varphi \Rightarrow \varphi}{\neg \varphi, \varphi \Rightarrow} \neg R \\
\frac{\Diamond \neg \varphi, \Box \varphi \Rightarrow}{\Box \varphi \Rightarrow \neg \Diamond \neg \varphi} \Diamond \\
\frac{\Box \varphi \Rightarrow \neg \Diamond \neg \varphi}{\Rightarrow \Box \varphi \rightarrow \neg \Diamond \neg \varphi} \neg R \\
\frac{}{\Rightarrow \Box \varphi \leftrightarrow \neg \Diamond \neg \varphi} \rightarrow R \\
\frac{\varphi \Rightarrow \varphi}{\Rightarrow \varphi, \neg \varphi} \neg R \\
\frac{\Rightarrow \varphi, \neg \varphi}{\Rightarrow \neg \varphi, \varphi} XR \\
\frac{\Rightarrow \neg \varphi, \varphi}{\Rightarrow \Diamond \neg \varphi, \Box \varphi} \Box \\
\frac{\Rightarrow \Diamond \neg \varphi, \Box \varphi}{\Rightarrow \Box \varphi, \Diamond \neg \varphi} XR \\
\frac{\Rightarrow \Box \varphi, \Diamond \neg \varphi}{\neg \Diamond \neg \varphi \Rightarrow \Box \varphi} \neg R \\
\frac{\neg \Diamond \neg \varphi \Rightarrow \Box \varphi}{\Rightarrow \neg \Diamond \neg \varphi \rightarrow \Box \varphi} \rightarrow R \\
\frac{}{\Rightarrow \Box \varphi \leftrightarrow \neg \Diamond \neg \varphi} \wedge R
\end{array}$$

Problema 55.1. Find sequent calculus proofs in **K** for the following *fórmulas*:

1. $\Box \neg p \rightarrow \Box(p \rightarrow q)$
2. $(\Box p \vee \Box q) \rightarrow \Box(p \vee q)$
3. $\Diamond p \rightarrow \Diamond(p \vee q)$
4. $\Box(p \wedge q) \rightarrow \Box p$

-NoValue-

$\frac{\varphi, \Gamma \Rightarrow \Delta}{\Box \varphi, \Gamma \Rightarrow \Delta} \text{T}\Box$	$\frac{\Gamma \Rightarrow \Delta, \varphi}{\Gamma \Rightarrow \Delta, \Diamond \varphi} \text{T}\Diamond$
$\frac{\Gamma \Rightarrow \Delta}{\Box \Gamma \Rightarrow \Diamond \Delta} \text{D}$	
$\frac{\Gamma, \Diamond \Pi \Rightarrow \Box \Delta, \Lambda, \varphi}{\Box \Gamma, \Pi \Rightarrow \Delta, \Diamond \Lambda, \Box \varphi} \text{B}\Box$	$\frac{\varphi, \Diamond \Gamma, \Pi \Rightarrow \Box \Lambda, \Delta}{\Diamond \varphi, \Gamma, \Box \Pi \Rightarrow \Lambda, \Diamond \Delta} \text{B}\Diamond$
$\frac{\Box \Gamma \Rightarrow \Diamond \Delta, \varphi}{\Box \Gamma \Rightarrow \Diamond \Delta, \Box \varphi} 4\Box$	$\frac{\varphi, \Box \Gamma \Rightarrow \Diamond \Delta}{\Diamond \varphi, \Box \Gamma \Rightarrow \Diamond \Delta} 4\Diamond$
$\frac{\Box \Gamma, \Diamond \Pi \Rightarrow \Box \Delta, \Diamond \Lambda, \varphi}{\Box \Gamma, \Diamond \Pi \Rightarrow \Box \Delta, \Diamond \Lambda, \Box \varphi} 5\Box$	$\frac{\varphi, \Diamond \Gamma, \Box \Pi \Rightarrow \Diamond \Delta, \Box \Lambda}{\Diamond \varphi, \Diamond \Gamma, \Box \Pi \Rightarrow \Diamond \Delta, \Box \Lambda} 5\Diamond$

Tabela 55.1: More modal rules.

Logic	R is ...	Rules
T = KT	reflexive	$\Box, \text{T}\Box, \text{T}\Diamond$
D = KD	serial	$\Box, \text{D}$
K4	transitive	$\Box, 4\Box, 4\Diamond$
B = KTB	reflexive, symmetric	$\Box, \text{T}\Box, \text{T}\Diamond$ $\text{B}\Box, \text{B}\Diamond$
S4 = KT4	reflexive, transitive	$\Box, \text{T}\Box, \text{T}\Diamond$ $4\Box, 4\Diamond$
S5 = KT5	reflexive, transitive, euclidean	$\Box, \text{T}\Box, \text{T}\Diamond$ $5\Box, 5\Diamond$

nml:seq:mru:
tab:more-rules

Tabela 55.2: Sequent rules for various modal logics.

nml:seq:mru:
tab:logics-rules

55.4 Rules for Other Accessibility Relations

In order to deal with logics determined by special accessibility relations, we consider the additional rules in [Tabela 55.1](#).

nml:seq:mru:
sec

Adding these rules results in systems that are sound and complete for the logics given in [Tabela 55.2](#).

Exemplo 55.3. We give a sequent *derivação* that shows $\mathbf{K4} \vdash 4, \text{O.s.}, \Box \varphi \rightarrow \Box \Box \varphi$.

55.4. RULES FOR OTHER ACCESSIBILITY RELATIONS

$$\frac{\frac{\Box\varphi \Rightarrow \Box\varphi}{\Box\varphi \Rightarrow \Box\Box\varphi} 4\Box}{\Rightarrow \Box\varphi \rightarrow \Box\Box\varphi} \rightarrow R$$

Exemplo 55.4. We give a sequent *derivação* that shows **S5** $\vdash 5$, O.s., $\Diamond\varphi \rightarrow \Box\Diamond\varphi$.

$$\frac{\frac{\Diamond\varphi \Rightarrow \Diamond\varphi}{\Diamond\varphi \Rightarrow \Box\Diamond\varphi} 5\Box}{\Rightarrow \Diamond\varphi \rightarrow \Box\Diamond\varphi} \rightarrow R$$

Exemplo 55.5. The sequent calculus for **S5** is not complete without the Cut rule; p.ex., $\Diamond\Box\varphi \rightarrow \varphi$, which is valid in **S5**, has no proof without Cut. Here is a *derivação* using Cut:

$$\frac{\frac{\Box\varphi \Rightarrow \Box\varphi}{\Diamond\Box\varphi \Rightarrow \Box\varphi} 5\Diamond \quad \frac{\varphi \Rightarrow \varphi}{\Box\varphi \Rightarrow \varphi} T\Box}{\Diamond\Box\varphi \Rightarrow \varphi} \text{Cut}$$

$$\frac{\Diamond\Box\varphi \Rightarrow \varphi}{\Rightarrow \Diamond\Box\varphi \rightarrow \varphi} \rightarrow R$$

Problema 55.2. Give sequent *derivações* that show the following:

1. **KT5** $\vdash B$;
2. **KT5** $\vdash 4$;
3. **KDB4** $\vdash T$;
4. **KB4** $\vdash 5$;
5. **KB5** $\vdash 4$;
6. **KT** $\vdash D$.

Parte XII

Applied Modal Logic

This part contains experimental draft material on some applications of modal logic, such as temporal and epistemic logics.

Capítulo 56

Temporal Logics

This chapter covers temporal logics.

-NoValue-

`content/applied-modal-logic/temporal-logic/introduction.tex`

56.1 Introduction

Temporal logics deal with claims about things that will or have been the case. Arthur Prior is credited as the originator of temporal logic, which he called *tense logic*. Our treatment of temporal logic here will largely follow Prior’s original modal treatment of introducing temporal operators into the basic framework of propositional logic, which treats claims as generally lacking in tense.

aml:tl:int:
sec

For example, in propositional logic, I might talk about a dog, Bezie, who sometimes sits and sometimes doesn’t sit, as dogs are wont to do. It would be contradictory in classical logic to claim that Bezie is sitting and also that Bezie is not sitting. But obviously both can be true, just not at the same time; adding temporal operators to the language can allow us to express that claim relatively easily. The addition of temporal operators also allows us to account for the validity of inferences like the one from “Bezie will get a treat or a ball” to “Bezie will get a treat or Bezie will get a ball.”

56.2. SEMANTICS FOR TEMPORAL LOGIC

However, a lot of philosophical issues arise with temporal logic that might lead us to adopt one framework of temporal logic over another. For example, a future contingent is a statement about the future that is neither necessary nor impossible. If we say “Richard will go to the grocery store tomorrow,” we are expressing a claim about something that has not yet happened, and whose truth value is contestable. In fact, it is contestable whether that claim can even be *assigned* a truth value in the first place. If we are strict determinists, then perhaps we can be comfortable with the idea that this sentence is in fact true or false, even before the event in question is supposed to take place—it just may be that we do not know its truth value yet. In contrast, we might believe in a genuinely open future, in which the truth values of future contingents are undetermined.

As it turns out, a lot of these commitments about the structure and nature of time are built in to our choices of models and frameworks of temporal logics. For example, we might ask ourselves whether we should construct models in which time is linear, branching or even circular. We might have to make decisions about whether our temporal models will have beginning and end points, and whether time is to be represented using discrete instants or as a continuum.

-NoValue-

`content/applied-modal-logic/temporal-logic/temporal-logic-semantics.tex`

56.2 Semantics for Temporal Logic

aml:tl:sem:
sec

Definição 56.1. The basic language of temporal logic contains

1. The propositional constant for **falsidade** $\perp$.
2. The propositional constant for **verdade** $\top$.
3. A **infinitos enumeráveis** set of **variáveis proposicionais**: $p_0, p_1, p_2, \dots$
4. The propositional connectives: $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\rightarrow$ (**condicional**), $\leftrightarrow$ (**bicondicional**).
5. Past operators P and H.
6. Future operators F and G.

Later on, we will discuss the potential addition of other kinds of modal operators.

Definição 56.2. *Fórmulas* of the temporal language are inductively defined as follows:

1. $\perp$ is an atomic **fórmula**.

2. $\top$ is an atomic fórmula.
3. Every propositional variable p_i is an (atomic) fórmula.
4. If φ is a fórmula, then $\neg\varphi$ is a fórmula.
5. If φ and ψ are fórmulas, then $(\varphi \wedge \psi)$ is a fórmula.
6. If φ and ψ are fórmulas, then $(\varphi \vee \psi)$ is a fórmula.
7. If φ and ψ are fórmulas, then $(\varphi \rightarrow \psi)$ is a fórmula.
8. If φ and ψ are fórmulas, then $(\varphi \leftrightarrow \psi)$ is a fórmula.
9. If φ is a fórmula, then $P\varphi, H\varphi, F\varphi, G\varphi$ are all fórmulas.
10. Nothing else is a fórmula.

The semantics of temporal logics are given in terms of relational models, as with other kinds of intensional logics.

Definição 56.3. A *model* for temporal language is a triple $\mathfrak{M} = \langle T, \prec, V \rangle$, where

1. T is a nonempty set, interpreted as points in time.
2. $\prec$ is a binary relation on T .
3. V is a function assigning to each variável proposicional p a set $V(p)$ of points in time.

When $t \prec t'$ holds, we say that t *precedes* t' . When $t \in V(p)$ we say p is *true at t* .

For now, you will notice that we do not impose any conditions on our precedence relation $\prec$. This means that at present, there are no restrictions on the structure of our temporal models, so we could have models in which time is linear, branching, circular, or has any structure whatsoever.

Just as with normal modal logic, every temporal model determines which fórmulas count as true at which points in it. We use the same notation “model $\mathfrak{M}$ makes fórmula φ true at point t ” for the basic notion of relational semantics. The relation is defined inductively and is identical to the normal modal case for all non-modal operators.

Definição 56.4. *Truth of a fórmula φ at t in a $\mathfrak{M}$* , in symbols: $\mathfrak{M}, t \Vdash \varphi$, is defined inductively as follows:

1. $\varphi \equiv \perp$: Never $\mathfrak{M}, t \Vdash \perp$.
2. $\varphi \equiv \top$: Always $\mathfrak{M}, t \Vdash \top$.
3. $\mathfrak{M}, t \Vdash p$ iff $t \in V(p)$

aml:tl:sem:
defn:tmodels

56.3. PROPERTIES OF TEMPORAL FRAMES

4. $\varphi \equiv \neg\psi$: $\mathfrak{M}, t \Vdash \varphi$ iff $\mathfrak{M}, t \nVdash \psi$.
5. $\varphi \equiv (\psi \wedge \chi)$: $\mathfrak{M}, t \Vdash \varphi$ iff $\mathfrak{M}, t \Vdash \psi$ and $\mathfrak{M}, t \Vdash \chi$.
6. $\varphi \equiv (\psi \vee \chi)$: $\mathfrak{M}, t \Vdash \varphi$ iff $\mathfrak{M}, t \Vdash \psi$ or $\mathfrak{M}, t \Vdash \chi$ (or both).
7. $\varphi \equiv (\psi \rightarrow \chi)$: $\mathfrak{M}, t \Vdash \varphi$ iff $\mathfrak{M}, t \nVdash \psi$ or $\mathfrak{M}, t \Vdash \chi$.
8. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\mathfrak{M}, t \Vdash \varphi$ iff either both $\mathfrak{M}, t \Vdash \psi$ and $\mathfrak{M}, t \Vdash \chi$ or neither $\mathfrak{M}, t \Vdash \psi$ nor $\mathfrak{M}, t \Vdash \chi$.
9. $\varphi \equiv P\psi$: $\mathfrak{M}, t \Vdash \varphi$ iff $\mathfrak{M}, t' \Vdash \psi$ for some $t' \in T$ with $t' \prec t$
10. $\varphi \equiv H\psi$: $\mathfrak{M}, t \Vdash \varphi$ iff $\mathfrak{M}, t' \Vdash \psi$ for every $t' \in T$ with $t' \prec t$
11. $\varphi \equiv F\psi$: $\mathfrak{M}, t \Vdash \varphi$ iff $\mathfrak{M}, t' \Vdash \psi$ for some $t' \in T$ with $t \prec t'$
12. $\varphi \equiv G\psi$: $\mathfrak{M}, t \Vdash \varphi$ iff $\mathfrak{M}, t' \Vdash \psi$ for every $t' \in T$ with $t \prec t'$

Based on the semantics, you might be able to see that the operators P and H are duals, as well as the operators F and G , such that we could define $H\varphi$ as $\neg P\neg\varphi$, and the same with G and F .

-NoValue-

content/applied-modal-logic/temporal-logic/properties-accessibility.tex

56.3 Properties of Temporal Frames

Given that our temporal models do not impose any conditions on the relation $\prec$, the only one of our familiar axioms that holds in all models is K , or its analogues K_G and K_H :

$$\begin{aligned} G(p \rightarrow q) &\rightarrow (Gp \rightarrow Gq) && (K_G) \\ H(p \rightarrow q) &\rightarrow (Hp \rightarrow Hq) && (K_H) \end{aligned}$$

However, if we want our models to impose stricter conditions on how time is represented, for instance by ensuring that $\prec$ is a linear order, then we will end up with other validities in our models.

Several of the properties from [Tabela 56.1](#) might seem like desirable features for a model that is intended to represent time. However, it is worth noting that, even though we can impose whichever conditions we like on the $\prec$ relation, not all conditions correspond to [fórmulas](#) that can be expressed in the language of temporal logic. For example, irreflexivity, or the idea that $\forall w \neg(w \prec w)$, does not have a corresponding formula in temporal logic.

-NoValue-

content/applied-modal-logic/temporal-logic/extra-temporal-operators.tex

<i>If $\prec$ is ...</i>	<i>then ... is true in $\mathfrak{M}$:</i>
<i>transitive:</i> $\forall u \forall v \forall w ((u \prec v \wedge v \prec w) \rightarrow u \prec w)$	$\text{FF}p \rightarrow \text{F}p$
<i>linear:</i> $\forall w \forall v (w \prec v \vee w = v \vee v \prec w)$	$(\text{F}p \vee \text{P}p) \rightarrow (\text{P}p \vee p \vee \text{F}p)$
<i>dense:</i> $\forall w \forall v (w \prec v \rightarrow \exists u (w \prec u \wedge u \prec v))$	$\text{F}p \rightarrow \text{FF}p$
<i>unbounded (past):</i> $\forall w \exists v (v \prec w)$	$\text{H}p \rightarrow \text{P}p$
<i>unbounded (future):</i> $\forall w \exists v (w \prec v)$	$\text{G}p \rightarrow \text{F}p$

Tabela 56.1: Some temporal frame correspondence properties.

aml:tl:acc:
tab:correspondence

56.4 Additional Operators for Temporal Logic

In addition to the unary operators for past and future, temporal logics also sometimes include binary operators **S** and **U**, intended to symbolize “since” and “until”. This means adding **S** and **U** into the language of temporal logic and adding the following clause into the definition of a temporal **fórmula**:

aml:tl:ext:
sec

If φ and ψ are **fórmulas**, then $(\text{S}\varphi\psi)$ and $(\text{U}\varphi\psi)$ are both **fórmulas**.

The semantics for these operators are then given as follows:

Definição 56.5. *Truth of a fórmula φ at t in a $\mathfrak{M}$:*

aml:tl:ext:
defn:since-until
aml:tl:ext:
defn:sub:mmodels-since
aml:tl:ext:
defn:sub:mmodels-until

1. $\varphi \equiv \text{S}\psi\chi$: $\mathfrak{M}, t \Vdash \varphi$ iff $\mathfrak{M}, t' \Vdash \psi$ for some $t' \in T$ with $t' \prec t$, and for all s with $t' \prec s \prec t$, $\mathfrak{M}, s \Vdash \chi$
2. $\varphi \equiv \text{U}\psi\chi$: $\mathfrak{M}, t \Vdash \varphi$ iff $\mathfrak{M}, t' \Vdash \psi$ for some $t' \in T$ with $t \prec t'$, and for all s with $t \prec s \prec t'$, $\mathfrak{M}, s \Vdash \chi$

The intuitive reading of $\text{S}\psi\chi$ is “Since ψ was the case, χ has been the case.” And the intuitive reading of $\text{U}\psi\chi$ is “Until ψ will be the case, χ will be the case.”

-NoValue-

content/applied-modal-logic/temporal-logic/possible-histories.tex

56.5 Possible Histories

The relational models of temporal logic that we have been using are extremely flexible, since we do not have to place any restrictions on the accessibility relation. This means that temporal models can branch in the past and in the future, but we might want to consider a more “modal” conception of branching,

aml:tl:poss:
sec

in which we consider sequences of events as possible histories. This does not necessarily require changing our language, though we might also add our “ordinary” modal operators $\Box$ and $\Diamond$, and we could also consider adding epistemic accessibility relations to represent changes in agents’ knowledge over time.

Definição 56.6. A *possible histories model* for the temporal language is a triple $\mathfrak{M} = \langle T, C, V \rangle$, where

1. T is a nonempty set, interpreted as states in time.
2. C is a set of computational paths, or *possible histories* of a system. In other words, C is a set of sequences σ of states $s_1, s_2, s_3, \dots$, where every $s_i \in T$.
3. V is a function assigning to each **variável proposicional** p a set $V(p)$ of points in time.

To make things simpler, we will also generally assume that when a history is in C , then so are all of its suffixes. For example, if s_1, s_2, s_3 is a sequence in C , then so are s_2, s_3 and s_3 . Also, when two states s_i and s_j appear in a sequence σ , we say that $s_i \prec_\sigma s_j$ when $i < j$. When $t \in V(p)$ we say p is *true at t* .

The one relevant change is that when we evaluate the truth of a **fórmula** at a point in time t in a model $\mathfrak{M}$, we do so relative to a history σ , in which t appears as a state. We do not need to change any of the semantics for **variáveis proposicionais** or for truth-functional connectives, though. All of those are exactly as they were in **Definição 56.4**, since none of those will make reference to σ . However, we now redefine our future operator F and add our $\Diamond$ operator with respect to these histories.

Definição 56.7. *Truth of a fórmula φ at t, σ in $\mathfrak{M}$, in symbols: $\mathfrak{M}, t, \sigma \Vdash \varphi$:*

1. $\varphi \equiv F\psi$: $\mathfrak{M}, t, \sigma \Vdash \varphi$ iff $\mathfrak{M}, t', \sigma \Vdash \psi$ for some $t' \in T$ such that $t \prec_\sigma t'$.
2. $\varphi \equiv \Diamond\psi$: $\mathfrak{M}, t, \sigma \Vdash \varphi$ iff $\mathfrak{M}, t, \sigma' \Vdash \psi$ for some $\sigma' \in C$ in which t occurs.

Other temporal and modal operators can be defined similarly. However, we can now represent claims that combine tense and modality. For example, we might symbolize “ p will not occur, but it might have occurred” using the formula $\neg Fp \wedge \Diamond Fp$. This would hold at a point and a history at which p does

not become true at a successor state, but there is an alternative history at which p will become true.

Capítulo 57

Epistemic Logics

This chapter covers the metatheory of epistemic logics. It is structured in a similar way to Aldo Antonelli's notes on classical basic modal logic, but has been rewritten by Audrey Yap in order to add material on bisimulation and dynamic epistemic logics.

-NoValue-

`content/applied-modal-logic/epistemic-logic/introduction.tex`

57.1 Introduction

Just as modal logic deals with *modal propositions* and the entailment relations among them, epistemic logic deals with *epistemic propositions* and the entailment relations among them. Rather than interpreting the modal operators as representing possibility and necessity, the unary connectives are interpreted in epistemic or doxastic ways, to model knowledge and belief. For example, we might want to express claims like the following:

aml:el:int:
sec

1. Richard knows that Calgary is in Alberta.
2. Audrey thinks it is possible that a dog is on the couch.
3. Richard knows that Audrey knows that her class is on Tuesdays.
4. Everyone knows that a year has 12 months.

Contemporary epistemic logic is often traced to Jaako Hintikka's *Knowledge and Belief*, from 1962, and it was written at a time when possible worlds semantics were becoming increasingly more used in logic. In fact, epistemic logics use most of the same semantic tools as other modal logics, but will

57.2. THE LANGUAGE OF EPISTEMIC LOGIC

interpret them differently. The main change is in what we take the *accessibility relation* to represent. In epistemic logics, they represent some form of *epistemic possibility*. We'll see that the epistemic notion that we're modelling will affect the constraints that we want to place on the accessibility relation. And we'll also see what happens to correspondence theory when it is given an epistemic interpretation. You'll notice that the examples above mention two agents: Richard and Audrey, and the relationship between the things that each one knows. The epistemic logics we'll consider will be multi-agent logics, in which such things can be expressed. In contrast, a single-agent epistemic logic would only talk about what one individual knows or believes.

-NoValue-

content/applied-modal-logic/epistemic-logic/language-epistemic-logic.tex

57.2 The Language of Epistemic Logic

aml:el:lan:
sec

Definição 57.1. Let G be a set of agent-symbols. The basic language of multi-agent epistemic logic contains

1. The propositional constant for **falsidade** $\perp$.
2. The propositional constant for **verdade** $\top$.
3. A **infinitos enumeráveis** set of **variáveis proposicionais**: $p_0, p_1, p_2, \dots$
4. The propositional connectives: $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\rightarrow$ (**condicional**), $\leftrightarrow$ (**bicondicional**).
5. The knowledge operator K_a where $a \in G$.

If we are only concerned with the knowledge of a single agent in our system, we can drop the reference to the set G , and individual agents. In that case, we only have the basic operator K .

Definição 57.2. **Fórmulas** of the epistemic language are inductively defined as follows:

1. $\perp$ is an atomic **fórmula**.
2. $\top$ is an atomic **fórmula**.
3. Every propositional variable p_i is an (atomic) **fórmula**.
4. If φ is a **fórmula**, then $\neg\varphi$ is a **fórmula**.
5. If φ and ψ are **fórmulas**, then $(\varphi \wedge \psi)$ is a **fórmula**.
6. If φ and ψ are **fórmulas**, then $(\varphi \vee \psi)$ is a **fórmula**.

7. If φ and ψ are fórmulas, then $(\varphi \rightarrow \psi)$ is a fórmula.
8. If φ and ψ are fórmulas, then $(\varphi \leftrightarrow \psi)$ is a fórmula.
9. If φ is a fórmula and $a \in G$, then $K_a\varphi$ is a fórmula.
10. Nothing else is a fórmula.

If a fórmula φ does not contain K_a , we say it is *modal-free*.

Definição 57.3. While the K operator is intended to symbolize individual knowledge, E , often read as “everybody knows,” symbolizes group knowledge. Where $G' \subseteq G$, we define $E_{G'}\varphi$ as an abbreviation for

$$\bigwedge_{b \in G'} K_b\varphi.$$

We can also define an even stronger sense of knowledge, namely *common knowledge* among a group of agents G . When a piece of information is common knowledge among a group of agents, it means that for every combination of agents in that group, they all know that each other knows that each other knows . . . ad infinitum. This is significantly stronger than group knowledge, and it is easy to come up with relational models in which a fórmula is group knowledge, but not common knowledge. We will use $C_G\varphi$ to symbolize “it is common knowledge among G that φ .”

-NoValue-

`content/applied-modal-logic/epistemic-logic/relational-models.tex`

57.3 Relational Models

The basic semantic concept for epistemic logics is the same as that of ordinary modal logics. Relational models still consist of a set of worlds, and an assignment that determines which *variáveis proposicionais* count as “true” at which worlds. And if we are only dealing with a single agent, we have a single accessibility relation as usual. However, if we have a multi-agent epistemic logic, then our single accessibility relation becomes a set of accessibility relations, one for each a in our set of agent symbols G .

aml:el:rel:
sec

A *relational model* consists of a set of worlds, which are related by binary accessibility relations—one for each agent—together with an assignment which determines which *variáveis proposicionais* are true at which worlds.

Definição 57.4. A *model* for the multi-agent epistemic language is a triple $\mathfrak{M} = \langle W, R, V \rangle$, where

1. W is a nonempty set of “worlds,”
2. For each $a \in G$, R_a is a binary accessibility relation on W , and

57.4. TRUTH AT A WORLD

3. V is a function assigning to each **variável proposicional** p a set $V(p)$ of possible worlds.

When $R_a ww'$ holds, we say that w' is *accessible by a from w* . When $w \in V(p)$ we say p is *true at w* .

The mechanics are just like the mechanics for normal modal logic, just with more accessibility relations added in. For a given agent, we will generally interpret their accessibility relation as representing something about their informational states. For example, we often treat $R_a ww'$, as expressing that w' is consistent with a 's information at w . Or to put it another way, at w , they cannot tell the difference between world w and world w' .

-NoValue-

content/applied-modal-logic/epistemic-logic/truth-at-w.tex

57.4 Truth at a World

aml:el:trw:
sec Just as with normal modal logic, every epistemic model determines which **fórmulas** count as true at which worlds in it. We use the same notation “model $\mathfrak{M}$ makes **fórmula** φ true at world w ” for the basic notion of relational semantics. The relation is defined inductively and is identical to the normal modal case for all non-modal operators.

aml:el:trw:
defn:mmodels **Definição 57.5.** *Truth of a fórmula φ at w in a $\mathfrak{M}$, in symbols: $\mathfrak{M}, w \Vdash \varphi$, is defined inductively as follows:*

1. $\varphi \equiv \perp$: Never $\mathfrak{M}, w \Vdash \perp$.
2. $\varphi \equiv \top$: Always $\mathfrak{M}, w \Vdash \top$.
3. $\mathfrak{M}, w \Vdash p$ iff $w \in V(p)$
4. $\varphi \equiv \neg\psi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \not\Vdash \psi$.
5. $\varphi \equiv (\psi \wedge \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ and $\mathfrak{M}, w \Vdash \chi$.
6. $\varphi \equiv (\psi \vee \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ or $\mathfrak{M}, w \Vdash \chi$ (or both).
7. $\varphi \equiv (\psi \rightarrow \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \not\Vdash \psi$ or $\mathfrak{M}, w \Vdash \chi$.
8. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff either both $\mathfrak{M}, w \Vdash \psi$ and $\mathfrak{M}, w \Vdash \chi$ or neither $\mathfrak{M}, w \Vdash \psi$ nor $\mathfrak{M}, w \Vdash \chi$.
9. $\varphi \equiv K_a \psi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w' \Vdash \psi$ for all $w' \in W$ with $R_a ww'$

aml:el:trw:
defn:sub:mmodels-box

Here's where we need to think about restrictions on our accessibility relations, though. After all, by clause (9), a **fórmula** $K_a \psi$ is true at w whenever there are no w' with $R_a ww'$. This is the same clause as in normal modal logic; when a world has no successors, all $\Box$ -formulas are vacuously true there. This seems

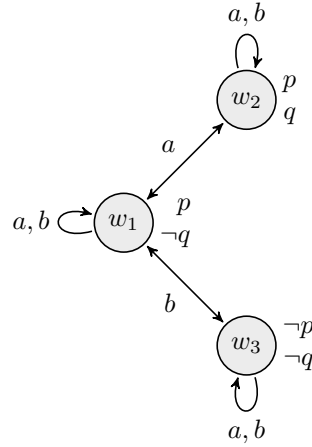


Figura 57.1: A simple epistemic model.

aml:el:trw:
fig:simple

extremely counterintuitive if we think about K as representing *knowledge*. After all, we tend to think that there are *no* circumstances under which an agent might know both φ and $\neg\varphi$ at the same time.

One solution is to ensure that our accessibility relation in epistemic logic will always be *reflexive*. This roughly corresponds to the idea that the actual world is consistent with an agent's information. In fact, epistemic logics typically use S5, but others might use weaker systems depending on what exactly they want the K_a relation to represent.

Problema 57.1. Consider which of the following hold in **Figura 57.1**:

1. $\mathfrak{M}, w_1 \models \neg q$;
2. $\mathfrak{M}, w_1 \models K_a \neg q$;
3. $\mathfrak{M}, w_1 \models K_b \neg q$;
4. $\mathfrak{M}, w_2 \models K_b q \vee K_b \neg q$;
5. $\mathfrak{M}, w_2 \models K_a (K_b q \vee K_b \neg q)$;
6. $\mathfrak{M}, w_3 \models E_{\{a,b\}} \neg q$;

Now that we have given our basic definition of truth at a world, the other semantic concepts from normal modal logic, such as modal validity and entailment, simply carry over, applied to this new way of thinking about the interpretation for the modal operators.

We are now also in a position to give truth conditions for the common knowledge operator C_G . Recall from **Seção 2.8** that the *transitive closure* R^+

57.5. ACCESSIBILITY RELATIONS AND EPISTEMIC PRINCIPLES

If R is ...	then ... is true in $\mathfrak{M}$:
	$\mathsf{K}(p \rightarrow q) \rightarrow (\mathsf{K}p \rightarrow \mathsf{K}q)$ (Closure)
<i>reflexive</i> : $\forall w Rww$	$\mathsf{K}p \rightarrow p$ (Veridicality)
<i>transitive</i> : $\forall u \forall v \forall w ((Ruv \wedge Rvw) \rightarrow Ruw)$	$\mathsf{K}p \rightarrow \mathsf{K}\mathsf{K}p$ (Positive Introspection)
<i>euclidean</i> : $\forall w \forall u \forall v ((Rwu \wedge Rwv) \rightarrow Ruw)$	$\neg \mathsf{K}p \rightarrow \mathsf{K}\neg \mathsf{K}p$ (Negative Introspection)

Tabela 57.1: Four epistemic principles.

aml:el:acc:
tab:four

of a relation R is defined as

$$R^+ = \bigcup_{n \in \mathbb{N}} R^n,$$

where

$$R^0 = R \text{ and } R^{n+1} = \{ \langle x, z \rangle : \exists y (R^n xy \wedge Ryz) \}.$$

Then, where G is a group of agents, we define $R_G = (\bigcup_{b \in G} R_b)^+$ to be the transitive closure of the union of all agents' accessibility relations.

Definição 57.6. If $G' \subseteq G$, we let $\mathfrak{M}, w \Vdash \mathsf{C}_{G'} \varphi$ iff for every w' such that $R_{G'} ww'$, $\mathfrak{M}, w' \Vdash \varphi$.

-NoValue-

content/applied-modal-logic/epistemic-logic/properties-accessibility.tex

57.5 Accessibility Relations and Epistemic Principles

aml:el:acc:
sec

Given what we already know about frame correspondence in normal modal logics, we might want to see what the characteristic **fórmulas** look like given epistemic interpretations. We have already said that epistemic logics are typically interpreted in S5. So let's take a look at how various epistemic principles are represented, and consider how they correspond to various frame conditions.

Recall from normal modal logic, that different modal **fórmulas** characterized different properties of accessibility relations. This table picks out a few that correspond to particular epistemic principles.

Veridicality, corresponding to the T axiom, is often treated as the most uncontroversial of these principles, as it represents that claim that if a **fórmula** is known, then it must be true. Closure, as well as Positive and Negative Introspection are much more contested.

Closure, corresponding to the K axiom, represents the idea that an agent's knowledge is closed under implication. This might seem plausible to us in some cases. For instance, I might know that if I am in Victoria, then I am on Vancouver Island. Barring odd skeptical scenarios, I do know that I am in Victoria, and this should also suggest that I know I am on Vancouver Island. So in this case, the logical closure of my knowledge might seem relatively intuitive. On the other hand, we do not always think through the consequences of our knowledge, and so this might lead to less intuitive results in other cases.

Positive Introspection, sometimes known as the KK-principle, is sometimes articulated as the statement that if I know something, then I know that I know. It is the epistemic counterpart of the 4 axiom. Correspondingly, negative introspection is articulated as the statement that if I *don't* know something, then I know that I don't know it, which is the counterpart of the 5 axiom. Both of these seem to admit of relatively ordinary counterexamples, in which I am unsure whether or not I know something that I do in fact know.

-NoValue-

[content/applied-modal-logic/epistemic-logic/bisimulations.tex](#)

57.6 Bisimulations

One remaining question that we might have about the expressive power of our epistemic language has to do with the relationship between models and the formulas that hold in them. We have seen from our frame correspondence results that when certain formulas are valid in a frame, they will also ensure that those frames satisfy certain properties. But does our modal language, for example, allow us to distinguish between a world at which there is a reflexive arrow, and an infinite chain of worlds, each of which leads to the next? That is, is there any formula A that might hold at only one of these two worlds?

aml:el:bsd:
sec

Bisimulation is a relationship that we can define between relational models to say that they have effectively the same structure. And as we will see, it will capture a sense of equivalence between models that can be captured in our epistemic language.

Definição 57.7 (Bisimulation). Let $M_1 = \langle W_1, R_1, V_1 \rangle$ and $M_2 = \langle W_2, R_2, V_2 \rangle$ be two relational models. And let $\mathcal{R} \subseteq W_1 \times W_2$ be a binary relation. We say that $\mathcal{R}$ is a *bisimulation* when for every $\langle w_1, w_2 \rangle \in \mathcal{R}$, we have:

1. $w_1 \in V_1(p)$ iff $w_2 \in V_2(p)$ for all **variáveis proposicionais** p .
2. For all agents $a \in A$ and worlds $v_1 \in W_1$, if $R_{1_a} w_1 v_1$ then there is some $v_2 \in W_2$ such that $R_{2_a} w_2 v_2$, and $\langle v_1, v_2 \rangle \in \mathcal{R}$.
3. For all agents $a \in A$ and worlds $v_2 \in W_2$, if $R_{2_a} w_2 v_2$ then there is some $v_1 \in W_1$ such that $R_{1_a} w_1 v_1$, and $\langle v_1, v_2 \rangle \in \mathcal{R}$.

57.7. PUBLIC ANNOUNCEMENT LOGIC

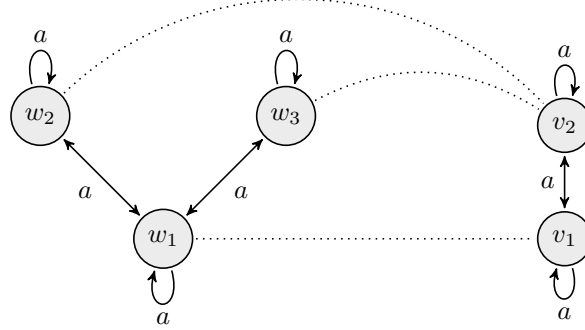


Figura 57.2: Two bisimilar models.

aml:el:bsd:
fig:bisimilar

When there is a bisimulation between M_1 and M_2 that links worlds w_1 and w_2 , we can also write $\langle M_1, w_1 \rangle \approx \langle M_2, w_2 \rangle$, and call $\langle M_1, w_1 \rangle$ and $\langle M_2, w_2 \rangle$ *bisimilar*.

The different clauses in the bisimulation relation ensure different things. Clause 1 ensures that bisimilar worlds will satisfy the same modal-free formulas, since it ensures agreement on all *variáveis proposicionais*. The other two clauses, sometimes referred to as “forth” and “back,” respectively, ensure that the accessibility relations will have the same structure.

Teorema 57.8. *If $\langle M_1, w_1 \rangle \approx \langle M_2, w_2 \rangle$, then for every *fórmula* φ , we have that $\mathfrak{M}_1, w_1 \Vdash \varphi$ iff $\mathfrak{M}_2, w_2 \Vdash \varphi$.*

Even though the two models pictured in [Figura 57.2](#) aren’t quite the same as each other, there is a bisimulation linking worlds w_1 and v_1 . This bisimulation will also link both w_2 and w_3 to v_2 , with the idea being that there is nothing expressible in our modal language that can really distinguish between them. The situation would be different if w_2 and w_3 satisfied different *variáveis proposicionais*, however.

-NoValue-

content/applied-modal-logic/epistemic-logic/public-announcement-logic-lang.tex

57.7 Public Announcement Logic

aml:el:pal:
sec

Dynamic epistemic logics allow us to represent the ways in which agents’ knowledge changes over time, or as they gain new information. Many of these represent changes in knowledge using informational *events* or *updates*. The most basic kind of update is a public announcement in which some formula is truthfully announced and all of the agents witness this taking place together. To do this, we expand the language as follows

Definição 57.9. Let G be a set of agent-symbols. The basic language of multi-agent epistemic logic with public announcements contains

1. The propositional constant for **falsidade** $\perp$.
2. The propositional constant for **verdade** $\top$.
3. A **infinitos enumeráveis** set of **variáveis proposicionais**: $p_0, p_1, p_2, \dots$
4. The propositional connectives: $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\rightarrow$ (**condicional**)
5. The knowledge operator K_a where $a \in G$.
6. The public announcement operator $[\psi]$ where ψ is a **fórmula**.

The public announcement operator functions as a box operator, and our inductive definition of the language is given accordingly:

Definição 57.10. **Fórmulas** of the epistemic language are inductively defined as follows:

1. $\perp$ is an atomic **fórmula**.
2. $\top$ is an atomic **fórmula**.
3. Every propositional variable p_i is an (atomic) **fórmula**.
4. If φ is a **fórmula**, then $\neg\varphi$ is a **fórmula**.
5. If φ and ψ are **fórmulas**, then $(\varphi \wedge \psi)$ is a **fórmula**.
6. If φ and ψ are **fórmulas**, then $(\varphi \vee \psi)$ is a **fórmula**.
7. If φ and ψ are **fórmulas**, then $(\varphi \rightarrow \psi)$ is a **fórmula**.
8. If φ and ψ are **fórmulas**, then $(\varphi \leftrightarrow \psi)$ is a **fórmula**.
9. If φ is a **fórmula** and $a \in G$, then $K_a\varphi$ is a **fórmula**.
10. If φ and ψ are **fórmulas**, then $[\varphi]\psi$ is a **fórmula**.
11. Nothing else is a **fórmula**.

The intended reading of the **fórmula** $[\varphi]\psi$ is “After φ is truthfully announced, ψ holds. It will sometimes also be useful to talk about common knowledge in the context of public announcements, so the language may also include the common knowledge operator $C_G\varphi$.

-NoValue-

content/applied-modal-logic/epistemic-logic/public-announcement-logic-semantics.tex

57.8 Semantics of Public Announcement Logic

aml:el:psm: Relational models for public announcement logics are the same as they were in
sec epistemic logics. However, the semantics for the public announcement operator are something new.

aml:el:psm: **Definição 57.11.** Truth of a fórmula φ at w in a $\mathfrak{M} = \langle W, R, V \rangle$, in symbols:
defn:mmodels $\mathfrak{M}, w \Vdash \varphi$, is defined inductively as follows:

1. $\varphi \equiv \perp$: Never $\mathfrak{M}, w \Vdash \perp$.
2. $\varphi \equiv \top$: Always $\mathfrak{M}, w \Vdash \top$.
3. $\mathfrak{M}, w \Vdash p$ iff $w \in V(p)$
4. $\varphi \equiv \neg\psi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \nVdash \psi$.
5. $\varphi \equiv (\psi \wedge \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ and $\mathfrak{M}, w \Vdash \chi$.
6. $\varphi \equiv (\psi \vee \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ or $\mathfrak{M}, w \Vdash \chi$ (or both).
7. $\varphi \equiv (\psi \rightarrow \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \nVdash \psi$ or $\mathfrak{M}, w \Vdash \chi$.
8. $\varphi \equiv (\psi \leftrightarrow \chi)$: $\mathfrak{M}, w \Vdash \varphi$ iff either both $\mathfrak{M}, w \Vdash \psi$ and $\mathfrak{M}, w \Vdash \chi$ or neither $\mathfrak{M}, w \Vdash \psi$ nor $\mathfrak{M}, w \Vdash \chi$.
9. $\varphi \equiv K_a\psi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w' \Vdash \psi$ for all $w' \in W$ with $R_a w w'$
10. $\varphi \equiv [\psi]\chi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ implies $\mathfrak{M} \mid \psi, w \Vdash \chi$

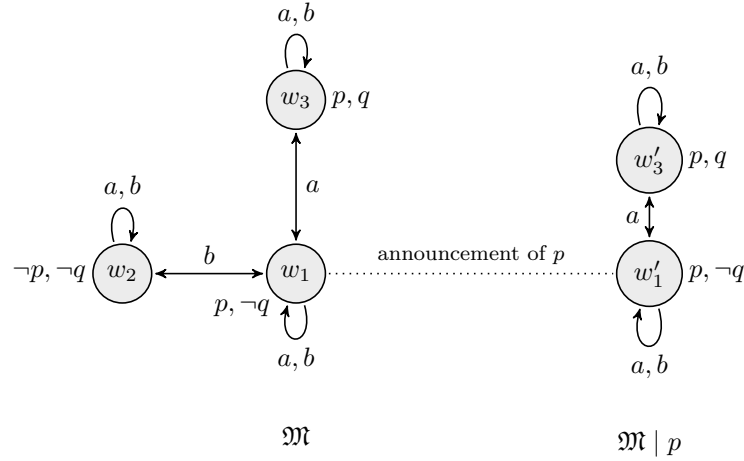
Where $\mathfrak{M} \mid \psi = \langle W', R', V' \rangle$ is defined as follows:

- a) $W' = \{u \in W : \mathfrak{M}, u \Vdash \psi\}$. So the worlds of $\mathfrak{M} \mid \psi$ are the worlds in $\mathfrak{M}$ at which ψ holds.
- b) $R'_a = R_a \cap (W' \times W')$. Each agent's accessibility relation is simply restricted to the worlds that remain in W' .
- c) $V'(p) = \{u \in W' : u \in V(p)\}$. Similarly, the propositional valuations at worlds remain the same, representing the idea that informational events will not change the truth value of propositional variables.

What is distinctive, then, about public announcement logics, is that the truth of a fórmula at $\mathfrak{M}$ can sometimes only be decided by referring to a model other than $\mathfrak{M}$ itself.

Notice also that our semantics treats the announcement operator as a $\Box$ operator, and so if a fórmula φ cannot be truthfully announced at a world, then $[\varphi]B$ will hold there trivially, just as all $\Box$ fórmulas hold at endpoints.

We can see the public announcement of a fórmula as shrinking a model, or restricting it to the worlds at which the fórmula was true. Figura 57.3 gives an example of the effects of publicly announcing p . One notable thing about

Figura 57.3: Before and after the public announcement of p .

that model is that agent b learns that p as a result of the announcement, while agent a does not (since a already knew that p was true).

More formally, we have $\mathfrak{M}, w_1 \Vdash \neg K_b p$ but $\mathfrak{M} \mid p, w'_1 \Vdash K_b p$. This implies that $\mathfrak{M}, w_1 \Vdash [p]K_b p$. But we have some even stronger claims that we can make about the result of the announcement. In fact, it is the case that $\mathfrak{M}, w_1 \Vdash [p]C_{\{a,b\}}p$. In other words, after p is announced, it becomes *common knowledge*.

We might wonder, though, whether this holds in the general case, and whether a truthful announcement of φ will *always* result in φ becoming common knowledge. It may be surprising that the answer is no. And in fact, it is possible to truthfully announce *fórmulas* that will no longer be true once they are announced. For example, consider the effects of announcing $p \wedge \neg K_b p$ at w_1 in **Figura 57.3**. In fact, $\mathfrak{M} \mid p$ and $\mathfrak{M} \mid (p \wedge \neg K_b p)$ are the same model. However, as we have already noted, $\mathfrak{M} \mid p, w'_1 \Vdash K_b p$. Therefore, $\mathfrak{M} \mid (p \wedge \neg K_b p), w'_1 \Vdash \neg(p \wedge \neg K_b p)$, so this is a *fórmula* that becomes false once it has been announced.

aml:el:psm:
fig:announcement-example

Parte XIII

Intuitionistic Logic

Capítulo 58

Introduction

-NoValue-

`content/intuitionistic-logic/introduction/constructive-reasoning.tex`

58.1 Constructive Reasoning

In contrast to extensions of classical logic by modal operators or second-order quantifiers, intuitionistic logic is “non-classical” in that it restricts classical logic. Classical logic is *non-constructive* in various ways. Intuitionistic logic is intended to capture a more “constructive” kind of reasoning characteristic of a kind of constructive mathematics. The following examples may serve to illustrate some of the underlying motivations.

Suppose someone claimed that they had determined a natural number n with the property that if n is even, the Riemann hypothesis is true, and if n is odd, the Riemann hypothesis is false. Great news! Whether the Riemann hypothesis is true or not is one of the big open questions of mathematics, and they seem to have reduced the problem to one of calculation, that is, to the determination of whether a specific number is even or not.

What is the magic value of n ? They describe it as follows: n is the natural number that is equal to 2 if the Riemann hypothesis is true, and 3 otherwise.

Angrily, you demand your money back. From a classical point of view, the description above does in fact determine a unique value of n ; but what you really want is a value of n that is given *explicitly*.

To take another, perhaps less contrived example, consider the following question. We know that it is possible to raise an irrational number to a rational power, and get a rational result. For example, $\sqrt{2}^2 = 2$. What is less clear is whether or not it is possible to raise an irrational number to an *irrational*

power, and get a rational result. The following theorem answers this in the affirmative:

Teorema 58.1. *There are irrational numbers a and b such that a^b is rational.*

Prova. Consider $\sqrt{2}^{\sqrt{2}}$. If this is rational, we are done: we can let $a = b = \sqrt{2}$. Otherwise, it is irrational. Then we have

$$(\sqrt{2}^{\sqrt{2}})^{\sqrt{2}} = \sqrt{2}^{\sqrt{2} \cdot \sqrt{2}} = \sqrt{2}^2 = 2,$$

which is rational. So, in this case, let a be $\sqrt{2}^{\sqrt{2}}$, and let b be $\sqrt{2}$. □

Does this constitute a valid proof? Most mathematicians feel that it does. But again, there is something a little bit unsatisfying here: we have proved the existence of a pair of real numbers with a certain property, without being able to say *which* pair of numbers it is. It is possible to prove the same result, but in such a way that the pair a, b is given in the proof: take $a = \sqrt{3}$ and $b = \log_3 4$. Then

$$a^b = \sqrt{3}^{\log_3 4} = 3^{1/2 \cdot \log_3 4} = (3^{\log_3 4})^{1/2} = 4^{1/2} = 2,$$

since $3^{\log_3 x} = x$.

Intuitionistic logic is designed to capture a kind of reasoning where moves like the one in the first proof are disallowed. Proving the existence of an x satisfying $\varphi(x)$ means that you have to give a specific x , and a proof that it satisfies φ , like in the second proof. Proving that φ or ψ holds requires that you can prove one or the other.

Formally speaking, intuitionistic logic is what you get if you restrict a **derivacão** system for classical logic in a certain way. From the mathematical point of view, these are just formal deductive systems, but, as already noted, they are intended to capture a kind of mathematical reasoning. One can take this to be the kind of reasoning that is justified on a certain philosophical view of mathematics (such as Brouwer's intuitionism); one can take it to be a kind of mathematical reasoning which is more "concrete" and satisfying (along the lines of Bishop's constructivism); and one can argue about whether or not the formal description captures the informal motivation. But whatever philosophical positions we may hold, we can study intuitionistic logic as a formally presented logic; and for whatever reasons, many mathematical logicians find it interesting to do so.

-NoValue-

content/intuitionistic-logic/introduction/syntax.tex

58.2 Syntax of Intuitionistic Logic

int:int:syn:
sec

58.2. SYNTAX OF INTUITIONISTIC LOGIC

The syntax of intuitionistic logic is the same as that for propositional logic. In classical propositional logic it is possible to define connectives by others, p.ex., one can define $\varphi \rightarrow \psi$ by $\neg\varphi \vee \psi$, or $\varphi \vee \psi$ by $\neg(\neg\varphi \wedge \neg\psi)$. Thus, presentations of classical logic often introduce some connectives as abbreviations for these definitions. This is not so in intuitionistic logic, with two exceptions: $\neg\varphi$ can be—and often is—defined as an abbreviation for $\varphi \rightarrow \perp$. Then, of course, $\perp$ must not itself be defined! Also, $\varphi \leftrightarrow \psi$ can be defined, as in classical logic, as $(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$.

Fórmulas of propositional intuitionistic logic are built up from *variáveis proposicionais* and the propositional constant $\perp$ using *logical connectives*. We have:

1. A infinito enumerável set At_0 of *variáveis proposicionais* $p_0, p_1, \dots$
2. The propositional constant for *falsidade* $\perp$.
3. The logical connectives: $\wedge$ (conjunction), $\vee$ (disjunction), $\rightarrow$ (conditional)
4. Punctuation marks: $(,)$, and the comma.

int:int:syn:
defn:formulas

Definição 58.2 (Formula). The set $\text{Frm}(\mathcal{L}_0)$ of *fórmulas* of propositional intuitionistic logic is defined inductively as follows:

1. $\perp$ is an atomic *fórmula*.
2. Every *variável proposicional* p_i is an atomic *fórmula*.
3. If φ and ψ are *fórmulas*, then $(\varphi \wedge \psi)$ is a *fórmula*.
4. If φ and ψ are *fórmulas*, then $(\varphi \vee \psi)$ is a *fórmula*.
5. If φ and ψ are *fórmulas*, then $(\varphi \rightarrow \psi)$ is a *fórmula*.
6. Nothing else is a *fórmula*.

In addition to the primitive connectives introduced above, we also use the following *defined* symbols: $\neg$ (negation) and $\leftrightarrow$ (*bicondicional*). Formulas constructed using the defined operators are to be understood as follows:

1. $\neg\varphi$ abbreviates $\varphi \rightarrow \perp$.
2. $\varphi \leftrightarrow \psi$ abbreviates $(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$.

Although $\neg$ is officially treated as an abbreviation, we will sometimes give explicit rules and clauses in definitions for $\neg$ as if it were primitive. This is mostly so we can state practice problems.

-NoValue-

content/intuitionistic-logic/introduction/bhk-interpretation.tex

58.3 The Brouwer–Heyting–Kolmogorov Interpretation

int:int:bhk:
sec

Proofs of validity of intuitionistic propositions using the BHK interpretation are confusing; they have to be explained better.

There is an informal constructive interpretation of the intuitionist connectives, usually known as the Brouwer–Heyting–Kolmogorov interpretation. It uses the notion of a “construction,” which you may think of as a constructive proof. (We don’t use “proof” in the BHK interpretation so as not to get confused with the notion of a *derivação* in a formal *derivação* system.) Based on this intuitive notion, the BHK interpretation explains the meanings of the intuitionistic connectives.

1. We assume that we know what constitutes a construction of an atomic statement.
2. A construction of $\varphi_1 \wedge \varphi_2$ is a pair $\langle M_1, M_2 \rangle$ where M_1 is a construction of φ_1 and M_2 is a construction of φ_2 .
3. A construction of $\varphi_1 \vee \varphi_2$ is a pair $\langle s, M \rangle$ where s is 1 and M is a construction of φ_1 , or s is 2 and M is a construction of φ_2 .
4. A construction of $\varphi \rightarrow \psi$ is a function that converts a construction of φ into a construction of ψ .
5. There is no construction for $\perp$ (absurdity).
6. $\neg\varphi$ is defined as synonym for $\varphi \rightarrow \perp$. That is, a construction of $\neg\varphi$ is a function converting a construction of φ into a construction of $\perp$.

Exemplo 58.3. Take $\neg\perp$ for example. A construction of it is a function which, given any construction of $\perp$ as input, provides a construction of $\perp$ as output. Obviously, the identity function Id is such a construction: given a construction M of $\perp$, $\text{Id}(M) = M$ yields a construction of $\perp$.

Generally speaking, $\neg\varphi$ means “A construction of φ is impossible”.

Exemplo 58.4. Let us prove $\varphi \rightarrow \neg\neg\varphi$ for any proposition φ , which is $\varphi \rightarrow ((\varphi \rightarrow \perp) \rightarrow \perp)$. The construction should be a function f that, given a construction M of φ , returns a construction $f(M)$ of $(\varphi \rightarrow \perp) \rightarrow \perp$. Here is how f constructs the construction of $(\varphi \rightarrow \perp) \rightarrow \perp$: We have to define a function g which, when given a construction h of $\varphi \rightarrow \perp$ as input, outputs a construction of $\perp$. We can define g as follows: apply the input h to the construction M of φ (that we received earlier). Since the output $h(M)$ of h is a construction of $\perp$, $f(M)(h) = h(M)$ is a construction of $\perp$ if M is a construction of φ .

58.3. THE BROUWER–HEYTING–KOLMOGOROV INTERPRETATION

Exemplo 58.5. Let us give a construction for $\neg(\varphi \wedge \neg\varphi)$, O.s., $(\varphi \wedge (\varphi \rightarrow \perp)) \rightarrow \perp$. This is a function f which, given as input a construction M of $\varphi \wedge (\varphi \rightarrow \perp)$, yields a construction of $\perp$. A construction of a conjunction $\psi_1 \wedge \psi_2$ is a pair $\langle N_1, N_2 \rangle$ where N_1 is a construction of ψ_1 and N_2 is a construction of ψ_2 . We can define functions p_1 and p_2 which recover from a construction of $\psi_1 \wedge \psi_2$ the constructions of ψ_1 and ψ_2 , respectively:

$$\begin{aligned} p_1(\langle N_1, N_2 \rangle) &= N_1 \\ p_2(\langle N_1, N_2 \rangle) &= N_2 \end{aligned}$$

Here is what f does: First it applies p_1 to its input M . That yields a construction of φ . Then it applies p_2 to M , yielding a construction of $\varphi \rightarrow \perp$. Such a construction, in turn, is a function $p_2(M)$ which, if given as input a construction of φ , yields a construction of $\perp$. In other words, if we apply $p_2(M)$ to $p_1(M)$, we get a construction of $\perp$. Thus, we can define $f(M) = p_2(M)(p_1(M))$.

Exemplo 58.6. Let us give a construction of $((\varphi \wedge \psi) \rightarrow \chi) \rightarrow (\varphi \rightarrow (\psi \rightarrow \chi))$, O.s., a function f which turns a construction g of $(\varphi \wedge \psi) \rightarrow \chi$ into a construction of $(\varphi \rightarrow (\psi \rightarrow \chi))$. The construction g is itself a function (from constructions of $\varphi \wedge \psi$ to constructions of χ). And the output $f(g)$ is a function h_g from constructions of φ to functions from constructions of ψ to constructions of χ .

Ok, this is confusing. We have to construct a certain function h_g , which will be the output of f for input g . The input of h_g is a construction M of φ . The output of $h_g(M)$ should be a function k_M from constructions N of ψ to constructions of χ . Let $k_{g,M}(N) = g(\langle M, N \rangle)$. Remember that $\langle M, N \rangle$ is a construction of $\varphi \wedge \psi$. So $k_{g,M}$ is a construction of $\psi \rightarrow \chi$: it maps constructions N of ψ to constructions of χ . Now let $h_g(M) = k_{g,M}$. That's a function that maps constructions M of φ to constructions $k_{g,M}$ of $\psi \rightarrow \chi$. Now let $f(g) = h_g$. That's a function that maps constructions g of $(\varphi \wedge \psi) \rightarrow \chi$ to constructions of $\varphi \rightarrow (\psi \rightarrow \chi)$. Whew!

The statement $\varphi \vee \neg\varphi$ is called the Law of Excluded Middle. We can prove it for some specific φ (p.ex., $\perp \vee \neg\perp$), but not in general. This is because the intuitionistic disjunction requires a construction of one of the disjuncts, but there are statements which currently can neither be proved nor refuted (say, Goldbach's conjecture). However, you can't refute the law of excluded middle either: that is, $\neg\neg(\varphi \vee \neg\varphi)$ holds.

Exemplo 58.7. To prove $\neg\neg(\varphi \vee \neg\varphi)$, we need a function f that transforms a construction of $\neg(\varphi \vee \neg\varphi)$, O.s., of $(\varphi \vee (\varphi \rightarrow \perp)) \rightarrow \perp$, into a construction of $\perp$. In other words, we need a function f such that $f(g)$ is a construction of $\perp$ if g is a construction of $\neg(\varphi \vee \neg\varphi)$.

Suppose g is a construction of $\neg(\varphi \vee \neg\varphi)$, O.s., a function that transforms a construction of $\varphi \vee \neg\varphi$ into a construction of $\perp$. A construction of $\varphi \vee \neg\varphi$ is a pair $\langle s, M \rangle$ where either $s = 1$ and M is a construction of φ , or $s = 2$ and M is a construction of $\neg\varphi$. Let h_1 be the function mapping a construction M_1 of φ to a construction of $\varphi \vee \neg\varphi$: it maps M_1 to $\langle 1, M_1 \rangle$. And let h_2 be the function

mapping a construction M_2 of $\neg\varphi$ to a construction of $\varphi \vee \neg\varphi$: it maps M_2 to $\langle 2, M_2 \rangle$.

Let k be $g \circ h_1$: it is a function which, if given a construction of φ , returns a construction of $\perp$, O.s., it is a construction of $\varphi \rightarrow \perp$ or $\neg\varphi$. Now let l be $g \circ h_2$. It is a function which, given a construction of $\neg\varphi$, provides a construction of $\perp$. Since k is a construction of $\neg\varphi$, $l(k)$ is a construction of $\perp$.

Together, what we've done is describe how we can turn a construction g of $\neg(\varphi \vee \neg\varphi)$ into a construction of $\perp$, O.s., the function f mapping a construction g of $\neg(\varphi \vee \neg\varphi)$ to the construction $l(k)$ of $\perp$ is a construction of $\neg\neg(\varphi \vee \neg\varphi)$.

As you can see, using the BHK interpretation to show the intuitionistic validity of *fórmulas* quickly becomes cumbersome and confusing. Luckily, there are better *derivação* systems for intuitionistic logic, and more precise semantic interpretations.

-NoValue-

`content/intuitionistic-logic/introduction/natural-deduction.tex`

58.4 Natural Deduction

Natural deduction without the $\perp_C$ rules is a standard *derivação* system for intuitionistic logic. We repeat the rules here and indicate the motivation using the BHK interpretation. In each case, we can think of a rule which allows us to conclude that if the premises have constructions, so does the conclusion. int:int:ntd:sec

Since natural deduction *derivações* have undischarged assumptions, we should consider such a *derivação*, say, of φ from *não descarregada* assumptions Γ , as a function that turns constructions of all $\psi \in \Gamma$ into a construction of φ . If there is a *derivação* of φ from no *não descarregada* assumptions, then there is a construction of φ in the sense of the BHK interpretation. For the purpose of the discussion, however, we'll suppress the Γ when not needed.

An assumption φ by itself is a *derivação* of φ from the *não descarregada* assumption φ . This agrees with the BHK-interpretation: the identity function on constructions turns any construction of φ into a construction of φ .

Conjunction

$\frac{\varphi \quad \psi}{\varphi \wedge \psi} \wedge\text{Intro}$	$\frac{\varphi \wedge \psi}{\varphi} \wedge\text{Elim}$
	$\frac{\varphi \wedge \psi}{\psi} \wedge\text{Elim}$

58.4. NATURAL DEDUCTION

Suppose we have constructions N_1, N_2 of φ_1 and φ_2 , respectively. Then we also have a construction $\varphi_1 \wedge \varphi_2$, namely the pair $\langle N_1, N_2 \rangle$.

A construction of $\varphi_1 \wedge \varphi_2$ on the BHK interpretation is a pair $\langle N_1, N_2 \rangle$. So assume we have such a pair. Then we also have a construction of each conjunct: N_1 is a construction of φ_1 and N_2 is a construction of φ_2 .

Conditional

$$\begin{array}{c}
 [\varphi]^u \\
 \vdots \\
 \psi \\
 \hline
 u \frac{\psi}{\varphi \rightarrow \psi} \rightarrow \text{Intro}
 \end{array}
 \qquad
 \frac{\varphi \rightarrow \psi \quad \varphi}{\psi} \rightarrow \text{Elim}$$

If we have a **derivação** of ψ from **não descarregada** assumption φ , then there is a function f that turns constructions of φ into constructions of ψ . That same function is a construction of $\varphi \rightarrow \psi$. So, if the premise of $\rightarrow$ Intro has a construction conditional on a construction of φ , the conclusion $\varphi \rightarrow \psi$ has a construction.

On the other hand, suppose there are constructions N of φ and f of $\varphi \rightarrow \psi$. A construction of $\varphi \rightarrow \psi$ is a function that turns constructions of φ into constructions of ψ . So, $f(N)$ is a construction of ψ , O.s., the conclusion of $\rightarrow$ Elim has a construction.

Disjunction

$$\begin{array}{c}
 \frac{\varphi}{\varphi \vee \psi} \vee \text{Intro} \\
 \frac{\psi}{\varphi \vee \psi} \vee \text{Intro}
 \end{array}
 \qquad
 \begin{array}{c}
 [\varphi]^n \quad [\psi]^n \\
 \vdots \quad \vdots \\
 \vdots \quad \vdots \\
 \chi \quad \chi \\
 \hline
 n \frac{\varphi \vee \psi}{\chi} \vee \text{Elim}
 \end{array}$$

If we have a construction N_i of φ_i we can turn it into a construction $\langle i, N_i \rangle$ of $\varphi_1 \vee \varphi_2$. On the other hand, suppose we have a construction of $\varphi_1 \vee \varphi_2$, O.s., a pair $\langle i, N_i \rangle$ where N_i is a construction of φ_i , and also functions f_1, f_2 , which turn constructions of φ_1, φ_2 , respectively, into constructions of χ . Then $f_i(N_i)$ is a construction of χ , the conclusion of $\vee$ Elim.

Absurdity

$$\frac{\perp}{\varphi} \perp_I$$

If we have a *derivação* of $\perp$ from *não descarregada* assumptions $\psi_1, \dots, \psi_n$, then there is a function $f(M_1, \dots, M_n)$ that turns constructions of $\psi_1, \dots, \psi_n$ into a construction of $\perp$. Since $\perp$ has no construction, there cannot be any constructions of all of $\psi_1, \dots, \psi_n$ either. Hence, f also has the property that if $M_1, \dots, M_n$ are constructions of $\psi_1, \dots, \psi_n$, respectively, then $f(M_1, \dots, M_n)$ is a construction of φ .

Rules for $\neg$

Since $\neg\varphi$ is defined as $\varphi \rightarrow \perp$, we strictly speaking do not need rules for $\neg$. But if we did, this is what they'd look like:

$$\begin{array}{c} [\varphi]^n \\ \vdots \\ \vdots \\ \vdots \\ \perp \\ n \frac{}{\neg\varphi} \neg\text{Intro} \end{array} \qquad \frac{\neg\varphi \quad \varphi}{\perp} \neg\text{Elim}$$

Examples of Derivações

1. $\vdash \varphi \rightarrow (\neg\varphi \rightarrow \perp)$, O.s., $\vdash \varphi \rightarrow ((\varphi \rightarrow \perp) \rightarrow \perp)$

$$\begin{array}{c} \frac{[\varphi]^2 \quad [\varphi \rightarrow \perp]^1}{\perp} \rightarrow\text{Elim} \\ 1 \frac{}{(\varphi \rightarrow \perp) \rightarrow \perp} \rightarrow\text{Intro} \\ 2 \frac{}{\varphi \rightarrow ((\varphi \rightarrow \perp) \rightarrow \perp)} \rightarrow\text{Intro} \end{array}$$

2. $\vdash ((\varphi \wedge \psi) \rightarrow \chi) \rightarrow (\varphi \rightarrow (\psi \rightarrow \chi))$

$$\begin{array}{c} \frac{[(\varphi \wedge \psi) \rightarrow \chi]^3 \quad \frac{[\varphi]^2 \quad [\psi]^1}{\varphi \wedge \psi} \wedge\text{Intro}}{\chi} \rightarrow\text{Elim} \\ 1 \frac{}{\psi \rightarrow \chi} \rightarrow\text{Intro} \\ 2 \frac{}{\varphi \rightarrow (\psi \rightarrow \chi)} \rightarrow\text{Intro} \\ 3 \frac{}{((\varphi \wedge \psi) \rightarrow \chi) \rightarrow (\varphi \rightarrow (\psi \rightarrow \chi))} \rightarrow\text{Intro} \end{array}$$

3. $\vdash \neg(\varphi \wedge \neg\varphi)$, O.s., $\vdash (\varphi \wedge (\varphi \rightarrow \perp)) \rightarrow \perp$

58.5. AXIOMATIC DERIVAÇÕES

$$\frac{\frac{[\varphi \wedge (\varphi \rightarrow \perp)]^1}{\varphi \rightarrow \perp} \wedge \text{Elim} \quad \frac{[\varphi \wedge (\varphi \rightarrow \perp)]^1}{\varphi} \wedge \text{Elim}}{1 \quad \frac{\perp}{(\varphi \wedge (\varphi \rightarrow \perp)) \rightarrow \perp} \rightarrow \text{Intro}} \rightarrow \text{Intro}$$

4. $\vdash \neg\neg(\varphi \vee \neg\varphi)$, O.s., $\vdash ((\varphi \vee (\varphi \rightarrow \perp)) \rightarrow \perp) \rightarrow \perp$

$$\frac{\frac{[(\varphi \vee (\varphi \rightarrow \perp)) \rightarrow \perp]^2 \quad \frac{[\varphi]^1}{\varphi \vee (\varphi \rightarrow \perp)} \vee \text{Intro}}{1 \quad \frac{\perp}{\varphi \rightarrow \perp} \rightarrow \text{Intro}} \rightarrow \text{Elim}}{2 \quad \frac{[(\varphi \vee (\varphi \rightarrow \perp)) \rightarrow \perp]^2 \quad \frac{\perp}{((\varphi \vee (\varphi \rightarrow \perp)) \rightarrow \perp) \rightarrow \perp} \rightarrow \text{Intro}}{[(\varphi \vee (\varphi \rightarrow \perp)) \rightarrow \perp] \rightarrow \perp} \rightarrow \text{Intro}}$$

Proposição 58.8. *If $\Gamma \vdash \varphi$ in intuitionistic logic, $\Gamma \vdash \varphi$ in classical logic. In particular, if φ is an intuitionistic theorem, it is also a classical theorem.*

Prova. Every natural deduction rule is also a rule in classical natural deduction, so every **derivação** in intuitionistic logic is also a **derivação** in classical logic. $\square$

Problema 58.1. Give **derivações** in intuitionistic logic of the following **fórmulas**:

1. $(\neg\varphi \vee \psi) \rightarrow (\varphi \rightarrow \psi)$
2. $\neg\neg\varphi \rightarrow \neg\varphi$
3. $\neg\neg(\varphi \wedge \psi) \leftrightarrow (\neg\neg\varphi \wedge \neg\neg\psi)$
4. $\neg(\varphi \vee \psi) \leftrightarrow (\neg\varphi \wedge \neg\psi)$
5. $(\neg\varphi \vee \neg\psi) \rightarrow \neg(\varphi \wedge \psi)$
6. $\neg\neg(\varphi \wedge \psi) \rightarrow (\neg\neg\varphi \vee \neg\neg\psi)$

-NoValue-

`content/intuitionistic-logic/introduction/axiomatic-derivations.tex`

58.5 Axiomatic Derivações

int:int:axd:
sec Axiomatic **derivações** for intuitionistic propositional logic are the conceptually simplest, and historically first, **derivação** systems. They work just as in classical propositional logic.

Definição 58.9 (Derivabilidade). If Γ is a set of **fórmulas** of $\mathcal{L}$ then a **derivação** from Γ is a finite sequence $\varphi_1, \dots, \varphi_n$ of **fórmulas** where for each $i \leq n$ one of the following holds:

1. $\varphi_i \in \Gamma$; or
2. φ_i is an axiom; or
3. φ_i follows from some φ_j and φ_k with $j < i$ and $k < i$ by modus ponens,
O.s., $\varphi_k \equiv \varphi_j \rightarrow \varphi_i$.

Definição 58.10 (Axioms). The set of Ax_0 of *axioms* for the intuitionistic propositional logic are all *fórmulas* of the following forms:

$(\varphi \wedge \psi) \rightarrow \varphi$	(58.1)	int:int:axd: ax:land1 int:int:axd:
$(\varphi \wedge \psi) \rightarrow \psi$	(58.2)	ax:land2 int:int:axd:
$\varphi \rightarrow (\psi \rightarrow (\varphi \wedge \psi))$	(58.3)	ax:land3 int:int:axd:
$\varphi \rightarrow (\varphi \vee \psi)$	(58.4)	ax:lor1 int:int:axd:
$\varphi \rightarrow (\psi \vee \varphi)$	(58.5)	ax:lor2 int:int:axd:
$(\varphi \rightarrow \chi) \rightarrow ((\psi \rightarrow \chi) \rightarrow ((\varphi \vee \psi) \rightarrow \chi))$	(58.6)	ax:lor3 int:int:axd:
$\varphi \rightarrow (\psi \rightarrow \varphi)$	(58.7)	ax:lif1 int:int:axd:
$(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$	(58.8)	ax:lif2 int:int:axd:
$\perp \rightarrow \varphi$	(58.9)	ax:lfalse1

Definição 58.11 (Derivabilidade). A *fórmula* φ is *derivável* from Γ , written $\Gamma \vdash \varphi$, if there is a *derivação* from Γ ending in φ .

Definição 58.12 (Theorems). A *fórmula* φ is a *theorem* if there is a *derivação* of φ from the empty set. We write $\vdash \varphi$ if φ is a theorem and $\nvdash \varphi$ if it is not.

Proposição 58.13. If $\Gamma \vdash \varphi$ in intuitionistic logic, $\Gamma \vdash \varphi$ in classical logic. In particular, if φ is an intuitionistic theorem, it is also a classical theorem.

Prova. Every intuitionistic axiom is also a classical axiom, so every *derivação* in intuitionistic logic is also a *derivação* in classical logic. $\square$

Capítulo 59

Semantics

This chapter collects definitions for semantics for intuitionistic logic. So far only Kripke and topological semantics are covered. There are no examples yet, either of how models make formulas true or of proofs that formulas are valid.

-NoValue-

`content/intuitionistic-logic/semantics/introduction.tex`

59.1 Introduction

int.sem:int:
sec

No logic is satisfactorily described without a semantics, and intuitionistic logic is no exception. Whereas for classical logic, the semantics based on **valorações** is canonical, there are several competing semantics for intuitionistic logic. None of them are completely satisfactory in the sense that they give an intuitionistically acceptable account of the meanings of the connectives.

The semantics based on **modelos relacionais**, similar to the semantics for modal logics, is perhaps the most popular one. In this semantics, **variáveis proposicionais** are assigned to worlds, and these worlds are related by an accessibility relation. That relation is always a partial order, O.s., it is reflexive, antisymmetric, and transitive.

Intuitively, you might think of these worlds as states of knowledge or “evidentiary situations.” A state w' is accessible from w iff, for all we know, w' is a possible (future) state of knowledge, O.s., one that is compatible with what’s known at w . Once a proposition is known, it can’t become un-known, O.s., whenever φ is known at w and Rww' , φ is known at w' as well. So “knowledge” is monotonic with respect to the accessibility relation.

If we define “ φ is known” as in epistemic logic as “true in all epistemic alternatives,” then $\varphi \wedge \psi$ is known at w if in all epistemic alternatives, both φ and ψ are known. But since knowledge is monotonic and R is reflexive, that

means that $\varphi \wedge \psi$ is known at w iff φ and ψ are known at w . For the same reason, $\varphi \vee \psi$ is known at w iff at least one of them is known. So for $\wedge$ and $\vee$, the truth conditions of the connectives coincide with those in classical logic.

The truth conditions for the conditional, however, differ from classical logic. $\varphi \rightarrow \psi$ is known at w iff at no w' with Rww' , φ is known without ψ also being known. This is not the same as the condition that φ is unknown or ψ is known at w . For if we know neither φ nor ψ at w , there might be a future epistemic state w' with Rww' such that at w' , φ is known without also coming to know ψ .

We know $\neg\varphi$ only if there is no possible future epistemic state in which we know φ . Here the idea is that if φ were knowable, then in some possible future epistemic state φ becomes known. Since we can't know $\perp$, in that future epistemic state, we would know φ but not know $\perp$.

On this interpretation the principle of excluded middle fails. For there are some φ which we don't yet know, but which we might come to know. For such a fórmula φ , both φ and $\neg\varphi$ are unknown, so $\varphi \vee \neg\varphi$ is not known. But we do know, p.ex., that $\neg(\varphi \wedge \neg\varphi)$. For no future state in which we know both φ and $\neg\varphi$ is possible, and we know this independently of whether or not we know φ or $\neg\varphi$.

Modelos relacionais are not the only available semantics for intuitionistic logic. The topological semantics is another: here propositions are interpreted as open sets in a topological space, and the connectives are interpreted as operations on these sets (p.ex., $\wedge$ corresponds to intersection).

-NoValue-

`content/intuitionistic-logic/semantics/relational-models.tex`

59.2 Modelos relacionais

In order to give a precise semantics for intuitionistic propositional logic, we have to give a definition of what counts as a model relative to which we can evaluate fórmulas. On the basis of such a definition it is then also possible to define semantics notions such as validity and entailment. One such semantics is given by modelos relacionais.

int:sem:rel:
sec

Definição 59.1. A modelo relacional for intuitionistic propositional logic is a triple $\mathfrak{M} = \langle W, R, V \rangle$, where

1. W is a non-empty set,
2. R is a partial order (O.s., a reflexive, antisymmetric, and transitive binary relation) on W , and
3. V is a function assigning to each variável proposicional p a subset of W , such that
4. V is monotone with respect to R , O.s., if $w \in V(p)$ and Rww' , then $w' \in V(p)$.

59.3. SEMANTIC NOTIONS

int:sem:rel:
defn:true-at-w

Definição 59.2. We define the notion of φ being true at w in $\mathfrak{M}$, $\mathfrak{M}, w \Vdash \varphi$, inductively as follows:

1. $\varphi \equiv p$: $\mathfrak{M}, w \Vdash \varphi$ iff $w \in V(p)$.
2. $\varphi \equiv \perp$: not $\mathfrak{M}, w \Vdash \varphi$.
3. $\varphi \equiv \neg\psi$: $\mathfrak{M}, w \Vdash \varphi$ iff for no w' such that Rww' , $\mathfrak{M}, w' \Vdash \psi$.
4. $\varphi \equiv \psi \wedge \chi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ and $\mathfrak{M}, w \Vdash \chi$.
5. $\varphi \equiv \psi \vee \chi$: $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}, w \Vdash \psi$ or $\mathfrak{M}, w \Vdash \chi$ (or both).
6. $\varphi \equiv \psi \rightarrow \chi$: $\mathfrak{M}, w \Vdash \varphi$ iff for every w' such that Rww' , not $\mathfrak{M}, w' \Vdash \psi$ or $\mathfrak{M}, w' \Vdash \chi$ (or both).

We write $\mathfrak{M}, w \nVdash \varphi$ if not $\mathfrak{M}, w \Vdash \varphi$. If Γ is a set of fórmulas, $\mathfrak{M}, w \Vdash \Gamma$ means $\mathfrak{M}, w \Vdash \psi$ for all $\psi \in \Gamma$.

Problema 59.1. Show that according to Definição 59.2, $\mathfrak{M}, w \Vdash \neg\varphi$ iff $\mathfrak{M}, w \nVdash \varphi \rightarrow \perp$.

int:sem:rel:
prop:true-monotonic

Proposição 59.3. Truth at worlds is monotonic with respect to R , O.s., if $\mathfrak{M}, w \Vdash \varphi$ and Rww' , then $\mathfrak{M}, w' \Vdash \varphi$.

Prova. Exercise. □

Problema 59.2. Prove Proposição 59.3.

-NoValue-

content/intuitionistic-logic/semantics/semantic-notions.tex

59.3 Semantic Notions

int:sem:sem:
sec

Definição 59.4. We say φ is true in the model $\mathfrak{M} = \langle W, R, V \rangle$, $\mathfrak{M} \Vdash \varphi$, iff $\mathfrak{M}, w \Vdash \varphi$ for all $w \in W$. φ is valid, $\models \varphi$, iff it is true in all models. We say a set of fórmulas Γ entails φ , $\Gamma \models \varphi$, iff for every model $\mathfrak{M}$ and every w such that $\mathfrak{M}, w \Vdash \Gamma$, $\mathfrak{M}, w \Vdash \varphi$.

int:sem:sem:
prop:sat-entails

Proposição 59.5.

int:sem:sem:
prop:sat-entails1

1. If $\mathfrak{M}, w \Vdash \Gamma$ and $\Gamma \models \varphi$, then $\mathfrak{M}, w \Vdash \varphi$.

int:sem:sem:
prop:sat-entails2

2. If $\mathfrak{M} \Vdash \Gamma$ and $\Gamma \models \varphi$, then $\mathfrak{M} \Vdash \varphi$.

Prova. 1. Suppose $\mathfrak{M} \Vdash \Gamma$. Since $\Gamma \models \varphi$, we know that if $\mathfrak{M}, w \Vdash \Gamma$, then $\mathfrak{M}, w \Vdash \varphi$. Since $\mathfrak{M}, u \Vdash \Gamma$ for all every $u \in W$, $\mathfrak{M}, w \Vdash \Gamma$. Hence $\mathfrak{M}, w \Vdash \varphi$.

2. Follows immediately from (1). $\square$

Definição 59.6. Suppose $\mathfrak{M}$ is a relational model and $w \in W$. The *restriction* $\mathfrak{M}_w = \langle W_w, R_w, V_w \rangle$ of $\mathfrak{M}$ to w is given by: int:sem:sem:
defn:restrict

$$\begin{aligned} W_w &= \{u \in W : Rwu\}, \\ R_w &= R \cap (W_w)^2, \text{ and} \\ V_w(p) &= V(p) \cap W_w. \end{aligned}$$

Proposição 59.7. $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}_w \Vdash \varphi$. int:sem:sem:
prop:restrict

Problema 59.3. Prove **Proposição 59.7**.

Proposição 59.8. Suppose for every model $\mathfrak{M}$ such that $\mathfrak{M} \Vdash \Gamma$, $\mathfrak{M} \Vdash \varphi$. Then $\Gamma \models \varphi$.

Prova. Suppose that $\mathfrak{M}, w \Vdash \Gamma$. By the **Proposição 59.7** applied to every $\psi \in \Gamma$, we have $\mathfrak{M}_w \Vdash \Gamma$. By the assumption, we have $\mathfrak{M}_w \Vdash \varphi$. By **Proposição 59.7** again, we get $\mathfrak{M}, w \Vdash \varphi$. $\square$

-NoValue-

content/intuitionistic-logic/semantics/topological-semantics.tex

59.4 Topological Semantics

Another way to provide a semantics for intuitionistic logic is using the mathematical concept of a topology. int:sem:top:
sec

Definição 59.9. Let X be a set. A *topology on X* is a set $\mathcal{O} \subseteq \wp(X)$ that satisfies the properties below. The **membros** of $\mathcal{O}$ are called the *open sets* of the topology. The set X together with $\mathcal{O}$ is called a *topological space*.

1. The empty set and the entire space are open: $\emptyset, X \in \mathcal{O}$.
2. Open sets are closed under finite intersections: if $U, V \in \mathcal{O}$ then $U \cap V \in \mathcal{O}$
3. Open sets are closed under arbitrary unions: if $U_i \in \mathcal{O}$ for all $i \in I$, then $\bigcup \{U_i : i \in I\} \in \mathcal{O}$.

We may write X for a topology if the collection of open sets can be inferred from the context; note that, still, only after X is endowed with open sets can it be called a topology.

Definição 59.10. A *topological model* of intuitionistic propositional logic is a triple $\mathfrak{X} = \langle X, \mathcal{O}, V \rangle$ where $\mathcal{O}$ is a topology on X and V is a function assigning an open set in $\mathcal{O}$ to each propositional variable.

Given a topological model $\mathfrak{X}$, we can define $[\varphi]_{\mathfrak{X}}$ inductively as follows:

1. $[\perp]_{\mathfrak{X}} = \emptyset$
2. $[p]_{\mathfrak{X}} = V(p)$
3. $[\varphi \wedge \psi]_{\mathfrak{X}} = [\varphi]_{\mathfrak{X}} \cap [\psi]_{\mathfrak{X}}$
4. $[\varphi \vee \psi]_{\mathfrak{X}} = [\varphi]_{\mathfrak{X}} \cup [\psi]_{\mathfrak{X}}$
5. $[\varphi \rightarrow \psi]_{\mathfrak{X}} = \text{Int}((X \setminus [\varphi]_{\mathfrak{X}}) \cup [\psi]_{\mathfrak{X}})$

Here, $\text{Int}(V)$ is the function that maps a set $V \subseteq X$ to its *interior*, that is, the union of all open sets it contains. In other words,

$$\text{Int}(V) = \bigcup \{U : U \subseteq V \text{ and } U \in \mathcal{O}\}.$$

Note that the interior of any set is always open, since it is a union of open sets. Thus, $[\varphi]_{\mathfrak{X}}$ is always an open set.

Although topological semantics is highly abstract, there are ways to think about it that might motivate it. Suppose that the **membros**, or “points,” of X are points at which statements can be evaluated. The set of all points where φ is true is the proposition expressed by φ . Not every set of points is a potential proposition; only the **membros** of $\mathcal{O}$ are. $\varphi \models \psi$ iff ψ is true at every point at which φ is true, O.s., $[\varphi]_{\mathfrak{X}} \subseteq [\psi]_{\mathfrak{X}}$, for all X . The absurd statement $\perp$ is never true, so $[\perp]_{\mathfrak{X}} = \emptyset$.

How must the propositions expressed by $\psi \wedge \chi$, $\psi \vee \chi$, and $\psi \rightarrow \chi$ be related to those expressed by ψ and χ for the intuitionistically valid laws to hold, O.s., so that $\varphi \vdash \psi$ iff $[\varphi]_{\mathfrak{X}} \subseteq [\psi]_{\mathfrak{X}}$? We require $\perp \vdash \varphi$ for any φ , which is satisfied because $\emptyset \subseteq U$ for all U . Since $\psi \wedge \chi \vdash \psi$, we require that $[\psi \wedge \chi]_{\mathfrak{X}} \subseteq [\psi]_{\mathfrak{X}}$, and similarly $[\psi \wedge \chi]_{\mathfrak{X}} \subseteq [\chi]_{\mathfrak{X}}$. The largest set satisfying $W \subseteq U$ and $W \subseteq V$ is $U \cap V$. Conversely, $\psi \vdash \psi \vee \chi$ and $\chi \vdash \psi \vee \chi$, and so we require that $[\psi]_{\mathfrak{X}} \subseteq [\psi \vee \chi]_{\mathfrak{X}}$ and $[\chi]_{\mathfrak{X}} \subseteq [\psi \vee \chi]_{\mathfrak{X}}$. The smallest set W such that $U \subseteq W$ and $V \subseteq W$ is $U \cup V$.

The definition for $\rightarrow$ is tricky: $\varphi \rightarrow \psi$ expresses the weakest proposition that, combined with φ , entails ψ . That $\varphi \rightarrow \psi$ combined with φ entails ψ is

clear from $(\varphi \rightarrow \psi) \wedge \varphi \vdash \psi$. So $[\varphi \rightarrow \psi]_{\mathfrak{X}}$ should be the greatest open set such that $[\varphi \rightarrow \psi]_{\mathfrak{X}} \cap [\varphi]_{\mathfrak{X}} \subset [\psi]_{\mathfrak{X}}$, leading to our definition.

Capítulo 60

Soundness and Completeness

This chapter collects soundness and completeness results for propositional intuitionistic logic. It needs an introduction. The completeness proof makes use of facts about provability that should be stated and proved explicitly somewhere.

-NoValue-

`content/intuitionistic-logic/soundness-completeness/soundness-axd.tex`

60.1 Soundness of Axiomatic Derivações

int:sc:sax:
sec

The soundness proof relies on the fact that all axioms are intuitionistically valid; this still needs to be proved, p.ex., in the Semantics chapter.

Teorema 60.1 (Soundness). *If $\Gamma \vdash \varphi$, then $\Gamma \models \varphi$.*

int:sc:sax:
thm:soundness

Prova. We prove that if $\Gamma \vdash \varphi$, then $\Gamma \models \varphi$. The proof is by induction on the number n of **fórmulas** in the **derivação** of φ from Γ . We show that if $\varphi_1, \dots, \varphi_n = \varphi$ is a **derivação** from Γ , then $\Gamma \models \varphi_n$. Note that if $\varphi_1, \dots, \varphi_n$ is a **derivação**, so is $\varphi_1, \dots, \varphi_k$ for any $k < n$.

There are no **derivações** of length 0, so for $n = 0$ the claim holds vacuously. So the claim holds for all **derivações** of length $< n$. We distinguish cases according to the justification of φ_n .

1. φ_n is an axiom. All axioms are valid, so $\Gamma \models \varphi_n$ for any Γ .

60.2. SOUNDNESS OF NATURAL DEDUCTION

2. $\varphi_n \in \Gamma$. Then for any $\mathfrak{M}$ and w , if $\mathfrak{M}, w \Vdash \Gamma$, obviously $\mathfrak{M} \Vdash \Gamma\varphi_n[w]$, O.s., $\Gamma \models \varphi$.
3. φ_n follows by MP from φ_i and $\varphi_j \equiv \varphi_i \rightarrow \varphi_n$. $\varphi_1, \dots, \varphi_i$ and $\varphi_1, \dots, \varphi_j$ are **derivações** from Γ , so by inductive hypothesis, $\Gamma \models \varphi_i$ and $\Gamma \models \varphi_i \rightarrow \varphi_n$.

Suppose $\mathfrak{M}, w \Vdash \Gamma$. Since $\mathfrak{M}, w \Vdash \Gamma$ and $\Gamma \models \varphi_i \rightarrow \varphi_n$, $\mathfrak{M}, w \Vdash \varphi_i \rightarrow \varphi_n$. By definition, this means that for all w' such that Rww' , if $\mathfrak{M}, w' \Vdash \varphi_i$ then $\mathfrak{M}, w' \Vdash \varphi_n$. Since R is reflexive, w is among the w' such that Rww' , O.s., we have that if $\mathfrak{M}, w \Vdash \varphi_i$ then $\mathfrak{M}, w \Vdash \varphi_n$. Since $\Gamma \models \varphi_i$, $\mathfrak{M}, w \Vdash \varphi_i$. So, $\mathfrak{M}, w \Vdash \varphi_n$, as we wanted to show. $\square$

-NoValue-

content/intuitionistic-logic/soundness-completeness/soundness-nd.tex

60.2 Soundness of Natural Deduction

int:sc:snd:
sec We will now prove soundness of natural deduction with regards to the relational semantics, that is, showing that if a **fórmula** is **derivável** from a set of assumptions then the set of assumptions entails the **fórmula**.

int:sc:snd:
thm:soundness **Teorema 60.2 (Soundness).** *If $\Gamma \vdash \varphi$, then $\Gamma \models \varphi$.*

Prova. We prove that if $\Gamma \vdash \varphi$, then $\Gamma \models \varphi$. The proof is by induction on the **derivação** of φ from Γ .

1. If the **derivação** consists of just the assumption φ , we have $\varphi \vdash \varphi$, and want to show that $\varphi \models \varphi$. Suppose that $\mathfrak{M}, w \Vdash \varphi$. Then trivially $\mathfrak{M}, w \Vdash \varphi$.
2. The **derivação** ends in $\wedge$ Intro: The **derivações** of the premises ψ from **não descarregada** assumptions Γ and of χ from **não descarregada** assumptions Δ show that $\Gamma \vdash \psi$ and $\Delta \vdash \chi$. By induction hypothesis we have that $\Gamma \models \psi$ and $\Delta \models \chi$. We have to show that $\Gamma \cup \Delta \models \varphi \wedge \psi$, since the **não descarregada** assumptions of the entire **derivação** are Γ together with Δ . So suppose $\mathfrak{M}, w \Vdash \Gamma \cup \Delta$. Then also $\mathfrak{M}, w \Vdash \Gamma$. Since $\Gamma \models \psi$, $\mathfrak{M}, w \Vdash \psi$. Similarly, $\mathfrak{M}, w \Vdash \chi$. So $\mathfrak{M}, w \Vdash \psi \wedge \chi$.
3. The **derivação** ends in $\wedge$ Elim: The **derivação** of the premise $\psi \wedge \chi$ from **não descarregada** assumptions Γ shows that $\Gamma \vdash \psi \wedge \chi$. By induction hypothesis, $\Gamma \models \psi \wedge \chi$. We have to show that $\Gamma \models \psi$. So suppose $\mathfrak{M}, w \Vdash \Gamma$. Since $\Gamma \models \psi \wedge \chi$, $\mathfrak{M}, w \Vdash \psi \wedge \chi$. Then also $\mathfrak{M}, w \Vdash \psi$. Similarly if $\wedge$ Elim ends in χ , then $\Gamma \models \chi$.
4. The **derivação** ends in $\vee$ Intro: Suppose the premise is ψ , and the **não descarregada** assumptions of the **derivação** ending in ψ are Γ . Then we

have $\Gamma \vdash \psi$ and by inductive hypothesis, $\Gamma \models \psi$. We have to show that $\Gamma \models \psi \vee \chi$. Suppose $\mathfrak{M}, w \Vdash \Gamma$. Since $\Gamma \models \psi$, $\mathfrak{M}, w \Vdash \psi$. But then also $\mathfrak{M}, w \Vdash \psi \vee \chi$. Similarly, if the premise is χ , we have that $\Gamma \models \chi$.

5. The *derivação* ends in $\vee$ Elim: The *derivações* ending in the premises are of $\psi \vee \chi$ from *não descarregada* assumptions Γ , of θ from *não descarregada* assumptions $\Delta_1 \cup \{\psi\}$, and of θ from *não descarregada* assumptions $\Delta_2 \cup \{\chi\}$. So we have $\Gamma \vdash \psi \vee \chi$, $\Delta_1 \cup \{\psi\} \vdash \theta$, and $\Delta_2 \cup \{\chi\} \vdash \theta$. By induction hypothesis, $\Gamma \models \psi \vee \chi$, $\Delta_1 \cup \{\psi\} \models \theta$, and $\Delta_2 \cup \{\chi\} \models \theta$. We have to prove that $\Gamma \cup \Delta_1 \cup \Delta_2 \models \theta$.

Suppose $\mathfrak{M}, w \Vdash \Gamma \cup \Delta_1 \cup \Delta_2$. Then $\mathfrak{M}, w \Vdash \Gamma$ and since $\Gamma \models \psi \vee \chi$, $\mathfrak{M}, w \Vdash \psi \vee \chi$. By definition of $\mathfrak{M} \Vdash$, either $\mathfrak{M}, w \Vdash \psi$ or $\mathfrak{M}, w \Vdash \chi$. So we distinguish cases: (a) $\mathfrak{M} \Vdash \psi[w]$. Then $\mathfrak{M}, w \Vdash \Delta_1 \cup \{\psi\}$. Since $\Delta_1 \cup \{\psi\} \models \theta$, we have $\mathfrak{M}, w \Vdash \theta$. (b) $\mathfrak{M}, w \Vdash \chi$. Then $\mathfrak{M}, w \Vdash \Delta_2 \cup \{\chi\}$. Since $\Delta_2 \cup \{\chi\} \models \theta$, we have $\mathfrak{M}, w \Vdash \theta$. So in either case, $\mathfrak{M}, w \Vdash \theta$, as we wanted to show.

6. The *derivação* ends with $\rightarrow$ Intro concluding $\psi \rightarrow \chi$. Then the premise is χ , and the *derivação* ending in the premise has *não descarregada* assumptions $\Gamma \cup \{\psi\}$. So we have that $\Gamma \cup \{\psi\} \vdash \chi$, and by induction hypothesis that $\Gamma \cup \{\psi\} \models \chi$. We have to show that $\Gamma \models \psi \rightarrow \chi$.

Suppose $\mathfrak{M}, w \Vdash \Gamma$. We want to show that for all w' such that Rww' , if $\mathfrak{M}, w' \Vdash \psi$, then $\mathfrak{M}, w' \Vdash \chi$. So assume that Rww' and $\mathfrak{M}, w' \Vdash \psi$. By **Proposição 59.3**, $\mathfrak{M}, w' \Vdash \Gamma$. Since $\Gamma \cup \{\psi\} \models \chi$, $\mathfrak{M}, w' \Vdash \chi$, which is what we wanted to show.

7. The *derivação* ends in $\rightarrow$ Elim and conclusion χ . The premises are $\psi \rightarrow \chi$ and ψ , with *derivações* from *não descarregada* assumptions Γ, Δ . So we have $\Gamma \vdash \psi \rightarrow \chi$ and $\Delta \vdash \psi$. By inductive hypothesis, $\Gamma \models \psi \rightarrow \chi$ and $\Delta \models \psi$. We have to show that $\Gamma \cup \Delta \models \chi$.

Suppose $\mathfrak{M}, w \Vdash \Gamma \cup \Delta$. Since $\mathfrak{M}, w \Vdash \Gamma$ and $\Gamma \models \psi \rightarrow \chi$, $\mathfrak{M}, w \Vdash \psi \rightarrow \chi$. By definition, this means that for all w' such that Rww' , if $\mathfrak{M}, w' \Vdash \psi$ then $\mathfrak{M}, w' \Vdash \chi$. Since R is reflexive, w is among the w' such that Rww' , O.s., we have that if $\mathfrak{M}, w \Vdash \psi$ then $\mathfrak{M}, w \Vdash \chi$. Since $\mathfrak{M}, w \Vdash \Delta$ and $\Delta \models \psi$, $\mathfrak{M}, w \Vdash \psi$. So, $\mathfrak{M}, w \Vdash \chi$, as we wanted to show.

8. The *derivação* ends in $\perp_I$, concluding φ . The premise is $\perp$ and the *não descarregada* assumptions of the *derivação* of the premise are Γ . Then $\Gamma \vdash \perp$. By inductive hypothesis, $\Gamma \models \perp$. We have to show $\Gamma \models \varphi$.

We proceed indirectly. If $\Gamma \not\models \varphi$ there is a model $\mathfrak{M}$ and world w such that $\mathfrak{M}, w \Vdash \Gamma$ and $\mathfrak{M}, w \not\Vdash \varphi$. Since $\Gamma \models \perp$, $\mathfrak{M}, w \Vdash \perp$. But that's impossible, since by definition, $\mathfrak{M}, w \not\Vdash \perp$. So $\Gamma \models \varphi$.

9. The *derivação* ends in $\neg$ Intro: Exercise.

10. The *derivação* ends in $\neg$ Elim: Exercise. □

60.3. LINDENBAUM'S LEMMA

Problema 60.1. Complete the proof of [Teorema 60.2](#). For the cases for $\neg$ -Intro and $\neg$ -Elim, use the definition of $\mathfrak{M}, w \Vdash \neg\varphi$ in [Definição 59.2](#), O.s., don't treat $\neg\varphi$ as defined by $\varphi \rightarrow \perp$.

Problema 60.2. Show that the following [fórmulas](#) are not [derivável](#) in intuitionistic logic:

1. $(\varphi \rightarrow \psi) \vee (\psi \rightarrow \varphi)$
2. $(\neg\neg\varphi \rightarrow \varphi) \rightarrow (\varphi \vee \neg\varphi)$
3. $(\varphi \rightarrow \psi \vee \chi) \rightarrow ((\varphi \rightarrow \psi) \vee (\varphi \rightarrow \chi))$

-NoValue-

[content/intuitionistic-logic/soundness-completeness/lindenbaum.tex](#)

60.3 Lindenbaum's Lemma

int:sc:lin:
sec The completeness theorem for intuitionistic logic is proved by assuming $\Gamma \not\vdash \varphi$ and constructing a model $\mathfrak{M} \Vdash \Gamma$ and $\mathfrak{M} \not\vdash \varphi$.

In classical logic the relation of [derivabilidade](#) can be reduced to the notion of consistency since a [fórmula](#) φ is [derivável](#) from a set of [fórmulas](#) iff the set together with the negation of φ is inconsistent. This is not possible in intuitionistic logic. In intuitionistic logic, if $\neg\varphi$ is inconsistent, we only get that $\vdash \neg\neg\varphi$. Since $\neg\neg\varphi \rightarrow \varphi$ does not hold intuitionistically in general, we cannot conclude that $\vdash \varphi$.

Thus, when constructing the model $\mathfrak{M}$, we will need to keep track of the non-[derivabilidade](#) of the [fórmula](#) φ and thus we will not be able to use a complete set $\Gamma^* \supseteq \Gamma$ to build the model $\mathfrak{M}$, as in every complete set Γ^* , we have $\Gamma^* \vdash \varphi \vee \neg\varphi$.

Instead of using a complete set Γ^* , we will use the notion of a prime set of formulas:

int:sc:lin:
defn:prime **Definição 60.3.** A set of [fórmulas](#) Γ is *prime* iff

1. Γ is consistent, O.s., $\Gamma \not\vdash \perp$;
2. if $\Gamma \vdash \varphi$ then $\varphi \in \Gamma$; and
3. if $\varphi \vee \psi \in \Gamma$ then $\varphi \in \Gamma$ or $\psi \in \Gamma$.

int:sc:lin:
lem:lindenbaum **Lema 60.4 (Lindenbaum's Lemma).** *If $\Gamma \not\vdash \varphi$, there is a $\Gamma^* \supseteq \Gamma$ such that Γ^* is prime and $\Gamma^* \not\vdash \varphi$.*

Prova. Let $\psi_1 \vee \chi_1, \psi_2 \vee \chi_2, \dots$, be an enumeration of all [fórmulas](#) of the form $\psi \vee \chi$. We'll define an increasing sequence of sets of [fórmulas](#) Γ_n , where each Γ_{n+1} is defined as Γ_n together with one new [fórmula](#). Γ^* will be the union of all Γ_n . The new [fórmulas](#) are selected so as to ensure that Γ^* is

prime and still $\Gamma^* \not\vdash \varphi$. This means that at each step we should find the first disjunction $\psi_i \vee \chi_i$ such that:

1. $\Gamma_n \vdash \psi_i \vee \chi_i$
2. $\psi_i \notin \Gamma_n$ and $\chi_i \notin \Gamma_n$

int:sc:lin:
gamma-1

int:sc:lin:
gamma-2

We add to Γ_n either ψ_i if $\Gamma_n \cup \{\psi_i\} \not\vdash \varphi$, or χ_i otherwise. We'll have to show that this works. For now, let's define $i(n)$ as the least i such that (1) and (2) hold.

Define $\Gamma_0 = \Gamma$ and

$$\Gamma_{n+1} = \begin{cases} \Gamma_n \cup \{\psi_{i(n)}\} & \text{if } \Gamma_n \cup \{\psi_{i(n)}\} \not\vdash \varphi \\ \Gamma_n \cup \{\chi_{i(n)}\} & \text{otherwise} \end{cases}$$

If $i(n)$ is undefined, O.s., whenever $\Gamma_n \vdash \psi \vee \chi$, either $\psi \in \Gamma_n$ or $\chi \in \Gamma_n$, we let $\Gamma_{n+1} = \Gamma_n$. Now let $\Gamma^* = \bigcup_{n=0}^{\infty} \Gamma_n$

First we show that for all n , $\Gamma_n \not\vdash \varphi$. We proceed by induction on n . For $n = 0$ the claim holds by the hypothesis of the theorem, O.s., $\Gamma \not\vdash \varphi$. If $n > 0$, we have to show that if $\Gamma_n \not\vdash \varphi$ then $\Gamma_{n+1} \not\vdash \varphi$. If $i(n)$ is undefined, $\Gamma_{n+1} = \Gamma_n$ and there is nothing to prove. So suppose $i(n)$ is defined. For simplicity, let $i = i(n)$.

We'll prove the contrapositive of the claim. Suppose $\Gamma_{n+1} \vdash \varphi$. By construction, $\Gamma_{n+1} = \Gamma_n \cup \{\psi_i\}$ if $\Gamma_n \cup \{\psi_i\} \not\vdash \varphi$, or else $\Gamma_{n+1} = \Gamma_n \cup \{\chi_i\}$. It clearly can't be the first, since then $\Gamma_{n+1} \not\vdash \varphi$. Hence, $\Gamma_n \cup \{\psi_i\} \vdash \varphi$ and $\Gamma_{n+1} = \Gamma_n \cup \{\chi_i\}$. By definition of $i(n)$, we have that $\Gamma_n \vdash \psi_i \vee \chi_i$. We have $\Gamma_n \cup \{\psi_i\} \vdash \varphi$. We also have $\Gamma_{n+1} = \Gamma_n \cup \{\chi_i\} \vdash \varphi$. Hence, $\Gamma_n \vdash \varphi$, which is what we wanted to show.

If $\Gamma^* \vdash \varphi$, there would be some finite subset $\Gamma' \subseteq \Gamma^*$ such that $\Gamma' \vdash \varphi$. Each $\theta \in \Gamma'$ must be in Γ_i for some i . Let n be the largest of these. Since $\Gamma_i \subseteq \Gamma_n$ if $i \leq n$, $\Gamma' \subseteq \Gamma_n$. But then $\Gamma_n \vdash \varphi$, contrary to our proof above that $\Gamma_n \not\vdash \varphi$.

Lastly, we show that Γ^* is prime, O.s., satisfies conditions (1), (2), and (3) of **Definição 60.3**.

First, $\Gamma^* \not\vdash \varphi$, so Γ^* is consistent, so (1) holds.

We now show that if $\Gamma^* \vdash \psi \vee \chi$, then either $\psi \in \Gamma^*$ or $\chi \in \Gamma^*$. This proves (3), since if $\psi \vee \chi \in \Gamma^*$ then also $\Gamma^* \vdash \psi \vee \chi$. So assume $\Gamma^* \vdash \psi \vee \chi$ but $\psi \notin \Gamma^*$ and $\chi \notin \Gamma^*$. Since $\Gamma^* \vdash \psi \vee \chi$, $\Gamma_n \vdash \psi \vee \chi$ for some n . $\psi \vee \chi$ appears on the enumeration of all disjunctions, say, as $\psi_j \vee \chi_j$. $\psi_j \vee \chi_j$ satisfies the properties in the definition of $i(n)$, namely we have $\Gamma_n \vdash \psi_j \vee \chi_j$, while $\psi_j \notin \Gamma_n$ and $\chi_j \notin \Gamma_n$. At each stage, at least one fewer disjunction $\psi_i \vee \chi_i$ satisfies the conditions (since at each stage we add either ψ_i or χ_i), so at some stage m we will have $j = i(m)$. But then either $\psi \in \Gamma_{m+1}$ or $\chi \in \Gamma_{m+1}$, contrary to the assumption that $\psi \notin \Gamma^*$ and $\chi \notin \Gamma^*$.

Now suppose $\Gamma^* \vdash \psi$. Then $\Gamma^* \vdash \psi \vee \psi$. But we've just proved that if $\Gamma^* \vdash \psi \vee \psi$ then $\psi \in \Gamma^*$. Hence, Γ^* satisfies (2) of **Definição 60.3**. $\square$

60.4. THE CANONICAL MODEL

Problema 60.3. Show that if $\Gamma \not\vdash \perp$ then Γ is consistent in classical logic, O.s., there is a valuation making all formulas in Γ true.

-NoValue-

content/intuitionistic-logic/soundness-completeness/canonical-model.tex

60.4 The Canonical Model

int:sc:mod: The worlds in our model will be finite sequences σ of natural numbers, O.s.,
sec $\sigma \in \mathbb{N}^*$. Note that $\mathbb{N}^*$ is inductively defined by:

1. $\Lambda \in \mathbb{N}^*$.
2. If $\sigma \in \mathbb{N}^*$ and $n \in \mathbb{N}$, then $\sigma.n \in \mathbb{N}^*$ (where $\sigma.n$ is $\sigma \frown \langle n \rangle$ and $\sigma \frown \sigma'$ is the concatenation of σ and σ').
3. Nothing else is in $\mathbb{N}^*$.

So we can use $\mathbb{N}^*$ to give inductive definitions.

Let $\langle \psi_1, \chi_1 \rangle, \langle \psi_2, \chi_2 \rangle, \dots$ be an enumeration of all pairs of **fórmulas**. Given a set of **fórmulas** Δ , define $\Delta(\sigma)$ by induction as follows:

1. $\Delta(\Lambda) = \Delta$
2. $\Delta(\sigma.n) = \begin{cases} (\Delta(\sigma) \cup \{\psi_n\})^* & \text{if } \Delta(\sigma) \cup \{\psi_n\} \not\vdash \chi_n \\ \Delta(\sigma) & \text{otherwise} \end{cases}$

Here by $(\Delta(\sigma) \cup \{\psi_n\})^*$ we mean the prime set of **fórmulas** which exists by **Lema 60.4** applied to the set $\Delta(\sigma) \cup \{\psi_n\}$ and the **fórmula** χ_n . Note that by this definition, if $\Delta(\sigma) \cup \{\psi_n\} \not\vdash \chi_n$, then $\Delta(\sigma.n) \vdash \psi_n$ and $\Delta(\sigma.n) \not\vdash \chi_n$. Note also that $\Delta(\sigma) \subseteq \Delta(\sigma.n)$ for any n . If Δ is prime, then $\Delta(\sigma)$ is prime for all σ .

int:sc:mod: **Definição 60.5.** Suppose Δ is prime. Then the *canonical model* $\mathfrak{M}(\Delta)$ for
defn:canonical-model Δ is defined by:

1. $W = \mathbb{N}^*$, the set of finite sequences of natural numbers.
2. R is the partial order according to which $R\sigma\sigma'$ iff σ is an initial segment of σ' (O.s., $\sigma' = \sigma \frown \sigma''$ for some sequence σ'').
3. $V(p) = \{\sigma : p \in \Delta(\sigma)\}$.

It is easy to verify that R is indeed a partial order. Also, the monotonicity condition on V is satisfied. Since $\Delta(\sigma) \subseteq \Delta(\sigma.n)$ we get $\Delta(\sigma) \subseteq \Delta(\sigma')$ whenever $R\sigma\sigma'$ by induction on σ .

-NoValue-

content/intuitionistic-logic/soundness-completeness/truth-lemma.tex

60.5 The Truth Lemma

The following lemma connects satisfaction in the canonical model with which **fórmulas** are **membros** of the prime set Δ . int:sc:tru:
sec

Lema 60.6. *If Δ is prime, then $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$ iff $\Delta(\sigma) \vdash \varphi$.* int:sc:tru:
lem:truth

Prova. By induction on φ .

1. $\varphi \equiv \perp$: Since $\Delta(\sigma)$ is prime, it is consistent, so $\Delta(\sigma) \not\vdash \varphi$. By definition, $\mathfrak{M}(\Delta), \sigma \not\Vdash \varphi$.
2. $\varphi \equiv p$: By definition of $\Vdash$, $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$ iff $\sigma \in V(p)$, O.s., $\Delta(\sigma) \vdash \varphi$.
3. $\varphi \equiv \neg\psi$: exercise.
4. $\varphi \equiv \psi \wedge \chi$: $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$ iff $\mathfrak{M}(\Delta), \sigma \Vdash \psi$ and $\mathfrak{M}(\Delta), \sigma \Vdash \chi$. By induction hypothesis, $\mathfrak{M}(\Delta), \sigma \Vdash \psi$ iff $\Delta(\sigma) \vdash \psi$, and similarly for χ . But $\Delta(\sigma) \vdash \psi$ and $\Delta(\sigma) \vdash \chi$ iff $\Delta(\sigma) \vdash \varphi$.
5. $\varphi \equiv \psi \vee \chi$: $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$ iff $\mathfrak{M}(\Delta), \sigma \Vdash \psi$ or $\mathfrak{M}(\Delta), \sigma \Vdash \chi$. By induction hypothesis, this holds iff $\Delta(\sigma) \vdash \psi$ or $\Delta(\sigma) \vdash \chi$. We have to show that this in turn holds iff $\Delta(\sigma) \vdash \varphi$. The left-to-right direction is clear. The right-to-left direction follows since $\Delta(\sigma)$ is prime.
6. $\varphi \equiv \psi \rightarrow \chi$: First the contrapositive of the left-to-right direction: Assume $\Delta(\sigma) \not\vdash \psi \rightarrow \chi$. Then also $\Delta(\sigma) \cup \{\psi\} \not\vdash \chi$. Since $\langle \psi, \chi \rangle$ is $\langle \psi_n, \chi_n \rangle$ for some n , we have $\Delta(\sigma.n) = (\Delta(\sigma) \cup \{\psi\})^*$, and $\Delta(\sigma.n) \vdash \psi$ but $\Delta(\sigma.n) \not\vdash \chi$. By inductive hypothesis, $\mathfrak{M}(\Delta), \sigma.n \Vdash \psi$ and $\mathfrak{M}(\Delta), \sigma.n \not\Vdash \chi$. Since $R\sigma(\sigma.n)$, this means that $\mathfrak{M}(\Delta), \sigma \not\Vdash \varphi$.
Now assume $\Delta(\sigma) \vdash \psi \rightarrow \chi$, and let $R\sigma\sigma'$. Since $\Delta(\sigma) \subseteq \Delta(\sigma')$, we have: if $\Delta(\sigma') \vdash \psi$, then $\Delta(\sigma') \vdash \chi$. In other words, for every σ' such that $R\sigma\sigma'$, either $\Delta(\sigma') \not\vdash \psi$ or $\Delta(\sigma') \vdash \chi$. By induction hypothesis, this means that whenever $R\sigma\sigma'$, either $\mathfrak{M}(\Delta), \sigma' \not\Vdash \psi$ or $\mathfrak{M}(\Delta), \sigma' \Vdash \chi$, O.s., $\mathfrak{M}(\Delta), \sigma \Vdash \varphi$. □

-NoValue-

content/intuitionistic-logic/soundness-completeness/completeness-thm.tex

60.6 The Completeness Theorem

Teorema 60.7. *If $\Gamma \models \varphi$ then $\Gamma \vdash \varphi$.* int:sc:cpl:
sec
int:sc:cpl:
thm:completeness

Prova. We prove the contrapositive: Suppose $\Gamma \not\models \varphi$. Then by **Lema 60.4**, there is a prime set $\Gamma^* \supseteq \Gamma$ such that $\Gamma^* \not\models \varphi$. Consider the canonical model $\mathfrak{M}(\Gamma^*)$ for Γ^* as defined in **Definição 60.5**. For any $\psi \in \Gamma$, $\Gamma^* \vdash \psi$. Note that $\Gamma^*(\Lambda) = \Gamma^*$. By the Truth Lemma (**Lema 60.6**), we have $\mathfrak{M}(\Gamma^*), \Lambda \Vdash \psi$ for all $\psi \in \Gamma$ and $\mathfrak{M}(\Gamma^*), \Lambda \not\Vdash \varphi$. This shows that $\Gamma \not\models \varphi$. □

Problema 60.4. Show that if φ only contains **variáveis proposicionais**, $\vee$, and $\wedge$, then $\not\models \varphi$. Use this to conclude that $\rightarrow$ is not definable in intuitionistic logic from $\vee$ and $\wedge$.

Problema 60.5. By using the completeness theorem prove that if $\vdash \varphi \vee \psi$ then $\vdash \varphi$ or $\vdash \psi$. (Hint: Assume $\mathfrak{M}_1 \not\models \varphi$ and $\mathfrak{M}_2 \not\models \psi$ and construct a new model $\mathfrak{M}$ such that $\mathfrak{M} \not\models \varphi \vee \psi$.)

Problema 60.6. Show that if $\mathfrak{M}$ is a relational model using a linear order then $\mathfrak{M} \Vdash (\varphi \rightarrow \psi) \vee (\psi \rightarrow \varphi)$.

-NoValue-

`content/intuitionistic-logic/soundness-completeness/decidability.tex`

60.7 Decidability

int:sc:dec: sec Observe that the proof of the completeness theorem gives us for every $\Gamma \not\models \varphi$ a model with an infinite number of worlds witnessing the fact that $\Gamma \not\models \varphi$. The following proposition shows that to prove $\models \varphi$ it is enough to prove that $\mathfrak{M} \Vdash \varphi$ for all finite models (O.s., models with a finite set of worlds).

int:sc:dec: thm:decidability **Teorema 60.8.** *If $\not\models \varphi$ then there is a finite model $\mathfrak{M}' \not\models \varphi$.*

Prova. Assume $\mathfrak{M} = \langle W, R, V \rangle$ is such that $\mathfrak{M} \not\models \varphi$ and P is the set of **variáveis proposicionais** occurring in φ . Define $\mathfrak{M}' = \langle W', R', V' \rangle$ by letting $W' = \{[w] : w \in W\}$ where $[w] = \{p \in P : w \in V(p)\}$, R' be the subset relation, and $V'(p) = \{[w] : p \in [w]\}$. It should be clear that W' is a finite set and that $\mathfrak{M}'$ is a **modelo relacional**.

It can be shown, by induction on φ , that

$$\mathfrak{M}, w \Vdash \varphi \text{ iff } \mathfrak{M}', [w] \Vdash \varphi$$

for all **fórmulas** φ with only **variáveis proposicionais** from P . This is left as an exercise for the reader. $\square$

Problema 60.7. Finish the proof of **Teorema 60.8** by showing that $\mathfrak{M}, w \Vdash \varphi$ iff $\mathfrak{M}', [w] \Vdash \varphi$ for all **fórmulas** φ with only propositional variables from P .

From [Teorema 60.8](#) it follows that there is an algorithm to decide whether $\models \varphi$.

Capítulo 61

Intuitionistic Tableaux

Draft chapter on prefixed tableaux for intuitionistic logic. Needs more examples, completeness proofs, and discussion of how one can find countermodels from unsuccessful searches for closed tableaux.

-NoValue-

[content/intuitionistic-logic/tableaux/introduction.tex](#)

61.1 Introduction

Tableaux are certain (downward-branching) trees of **fórmulas assinadas**, O.s., [int:tab:int:sec](#) pairs consisting of a truth value sign ($\mathbb{T}$ or $\mathbb{F}$) and a **sentença**

$\mathbb{T}\varphi$ or $\mathbb{F}\varphi$.

A **tableau** begins with a number of *assumptions*. Each further **fórmula assinada** is generated by applying one of the inference rules. Some inference rules add one or more **fórmulas assinadas** to a tip of the tree; others add two new tips, resulting in two branches. Rules result in **fórmulas assinadas** where the **fórmula** is less complex than that of the **fórmula assinada** to which it was applied. When a branch contains both $\mathbb{T}\varphi$ and $\mathbb{F}\varphi$, we say the branch is *closed*. If every branch in a **tableau** is closed, the entire **tableau** is closed. A closed **tableau** constitutes a **derivação** that shows that the set of **fórmulas assinadas** which were used to begin the **tableau** are unsatisfiable. This can be used to define a $\vdash$ relation: $\Gamma \vdash \varphi$ iff there is some finite set $\Gamma_0 = \{\psi_1, \dots, \psi_n\} \subseteq \Gamma$ such that there is a closed **tableau** for the assumptions

$$\{\mathbb{F}\varphi, \mathbb{T}\psi_1, \dots, \mathbb{T}\psi_n\}.$$

61.2. RULES FOR INTUITIONISTIC LOGIC

$\frac{\sigma \mathbb{T} \varphi \wedge \psi}{\sigma \mathbb{T} \varphi \quad \sigma \mathbb{T} \psi} \wedge \mathbb{T}$	$\frac{\sigma \mathbb{F} \varphi \wedge \psi}{\sigma \mathbb{F} \varphi \quad \quad \sigma \mathbb{F} \psi} \wedge \mathbb{F}$
$\frac{\sigma \mathbb{T} \varphi \vee \psi}{\sigma \mathbb{T} \varphi \quad \quad \sigma \mathbb{T} \psi} \vee \mathbb{T}$	$\frac{\sigma \mathbb{F} \varphi \vee \psi}{\sigma \mathbb{F} \varphi \quad \sigma \mathbb{F} \psi} \vee \mathbb{F}$

Tabela 61.1: Prefixed **tableau** rules for $\wedge$ and $\vee$

int:tab:rul:
tab:prop-rules

For intuitionistic logic, we have to both extend the notion of **fórmula assinada** and adjust the rules for the connectives. In addition to a sign($\mathbb{T}$ or $\mathbb{F}$), **fórmulas** in modal **tableaux** also have *prefixes* σ . The prefixes are non-empty sequences of positive integers, O.s., $\sigma \in (\mathbb{Z}^+)^* \setminus \{\Lambda\}$. When we write such prefixes without the surrounding $\langle \rangle$, and separate the individual **membros** by .'s instead of , 's. If σ is a prefix, then $\sigma.n$ is $\sigma \frown \langle n \rangle$; p.ex., if $\sigma = 1.2.1$, then $\sigma.3$ is $1.2.1.3$. So for instance,

$$1.2 \mathbb{T} \varphi \rightarrow (\psi \rightarrow \chi)$$

is a *prefixed fórmula assinada* (or just a *prefixed fórmula* for short).

Intuitively, the prefix names a world in a model that might satisfy the **fórmulas** on a branch of a **tableau**, and if σ names some world, then $\sigma.n$ names a world accessible from (the world named by) σ .

In intuitionistic models, the accessibility relation is reflexive and transitive. In terms of prefixes, this means that σ is accessible from σ itself, and so is any prefix that extends σ , O.s., any prefix of the form $\sigma.n_1 \dots .n_k$. Let's introduce the notation $\sigma.*$ to indicate σ itself and any extension of it. In other words, the prefixes $\sigma.*$ are all and only the prefixes accessible from σ .

-NoValue-

content/intuitionistic-logic/tableaux/rules.tex

61.2 Rules for Intuitionistic Logic

int:tab:rul:
sec

The rules for the connectives $\wedge$ and $\vee$ are the same as for regular propositional signed **tableaux**, just with prefixes added. In each case, the rule applied to a signed **fórmula** $\sigma S \varphi$ produces new **fórmulas** that are also prefixed by σ . This should be intuitively clear: p.ex., if $\varphi \wedge \psi$ is true at (a world named by) σ , then φ and ψ are true at σ (and not at any other world). We collect the rules for $\wedge$ and $\vee$ in **Tabela 61.1**.

The closure condition is similar to that for ordinary **tableaux**, although we require that not just the **fórmulas**, but also that the prefixes must match. In

fact, we can be somewhat more liberal: Since in intuitionistic models, **fórmulas**, once true, remain true, it is impossible that φ is true at σ but false at any accessible prefix $\sigma.*$. So a branch is closed if it contains both

$$\sigma \mathbb{T} \varphi \quad \text{and} \quad \sigma.* \mathbb{F} \varphi$$

for some prefix σ and **fórmula** φ . Note that if the signs are reversed, O.s., if it contains

$$\sigma \mathbb{F} \varphi \quad \text{and} \quad \sigma.* \mathbb{T} \varphi$$

the branch is closed only if $*$ is the empty sequence.

In addition, a branch is closed if it contains $\sigma \mathbb{T} \perp$.

The rules for setting up assumptions is also as for ordinary **tableaux**, except that for assumptions we always use the prefix 1. (It does not matter which prefix we use, as long as it's the same for all assumptions.) So, p.ex., we say that

$$\psi_1, \dots, \psi_n \vdash \varphi$$

iff there is a closed tableau for the assumptions

$$1 \mathbb{T} \psi_1, \dots, 1 \mathbb{T} \psi_n, 1 \mathbb{F} \varphi.$$

For the conditional $\rightarrow$, the rules differ from the classical and modal cases. The $\mathbb{T} \rightarrow$ rule extends a branch containing $\sigma \mathbb{T} \varphi \rightarrow \psi$ by $\sigma.* \mathbb{T} \varphi$ and $\sigma.* \mathbb{F} \psi$ on two different branches. It can only be applied for a prefix $\sigma.*$ which *already* occurs on the branch in which it is applied. Let's call such a prefix “used” (on the branch). (Since $\sigma.*$ includes σ itself, the rule can always be applied by adding the prefixed signed formulas $\sigma \mathbb{T} \varphi$ and $\sigma \mathbb{F} \psi$ on separate branches.)

The $\mathbb{F} \rightarrow$ rule extends a branch containing $\sigma \mathbb{F} \varphi \rightarrow \psi$ by both $\sigma.n \mathbb{T} \varphi$ and $\sigma.n \mathbb{F} \psi$ on the same branch, with $\sigma.n$ a prefix new to the branch.

The rules for $\neg$ are defined analogously (using the definition of $\neg \varphi$ as $\varphi \rightarrow \perp$).

The rules are given in **Tabela 61.2**.

-NoValue-

`content/intuitionistic-logic/tableaux/proofs.tex`

61.3 Tableaux for Intuitionistic Logic

Exemplo 61.1. We give a closed tableau that shows $(\varphi \wedge \psi) \rightarrow \chi \vdash \varphi \rightarrow (\psi \rightarrow \chi)$.

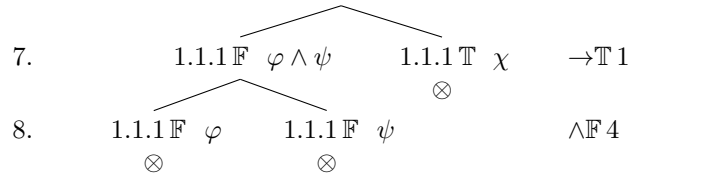
int:tab:prf:
sec

61.3. TABLEAUX FOR INTUITIONISTIC LOGIC

$\frac{\sigma \mathbb{T} \neg \varphi}{\sigma.* \mathbb{F} \varphi} \neg \mathbb{T}$ $\sigma.*$ is used	$\frac{\sigma \mathbb{F} \neg \varphi}{\sigma.n \mathbb{T} \varphi} \neg \mathbb{F}$ $\sigma.n$ is new
$\frac{\sigma \mathbb{T} \varphi \rightarrow \psi}{\sigma.* \mathbb{F} \varphi \mid \sigma.* \mathbb{T} \psi} \rightarrow \mathbb{T}$ $\sigma.*$ is used	$\frac{\sigma \mathbb{F} \varphi \rightarrow \psi}{\sigma.n \mathbb{T} \varphi \mid \sigma.n \mathbb{F} \psi} \rightarrow \mathbb{F}$ $\sigma.n$ is new

Tabela 61.2: Prefixed **tableau** rules for $\neg$ and $\rightarrow$

int:tab:rul:
tab:rules-lif-lnot

1.	$1 \mathbb{T} (\varphi \wedge \psi) \rightarrow \chi$	Assumption
2.	$1 \mathbb{F} \varphi \rightarrow (\psi \rightarrow \chi)$	Assumption
3.	$1.1 \mathbb{T} \varphi$	$\rightarrow \mathbb{F} 2$
4.	$1.1 \mathbb{F} \psi \rightarrow \chi$	$\rightarrow \mathbb{F} 2$
5.	$1.1.1 \mathbb{T} \psi$	$\rightarrow \mathbb{F} 4$
6.	$1.1.1 \mathbb{F} \chi$	$\rightarrow \mathbb{F} 4$
		
7.	$1.1.1 \mathbb{F} \varphi \wedge \psi$	$\rightarrow \mathbb{T} 1$
8.	$1.1.1 \mathbb{T} \chi$	$\wedge \mathbb{F} 4$

Problema 61.1. Find closed intuitionistic **tableaux** to show the following:

- $\vdash \varphi \rightarrow (\psi \rightarrow \varphi)$
- $\vdash \neg(\varphi \wedge \neg \varphi)$
- $\varphi \rightarrow (\psi \rightarrow \chi) \vdash (\varphi \wedge \psi) \rightarrow \chi$
- $\neg \varphi \vee \neg \psi \vdash \neg(\varphi \wedge \psi)$

-NoValue-

content/intuitionistic-logic/tableaux/soundness.tex

61.4 Soundness for Intuitionistic Tableaux

explanation In order to show that intuitionistic **tableaux** are sound, we have to show that int:tab:sou:sec if

$$1 \mathbb{T} \psi_1, \dots, 1 \mathbb{T} \psi_n, 1 \mathbb{F} \varphi$$

has a closed **tableau** then $\psi_1, \dots, \psi_n \models \varphi$. It is easier to prove the contrapositive: if for some $\mathfrak{M}$ and world w , $\mathfrak{M}, w \Vdash \psi_i$ for all $i = 1, \dots, n$ but $\mathfrak{M}, w \nVdash \varphi$, then no **tableau** can close. Such a countermodel shows that the initial assumptions of the **tableau** are satisfiable. The strategy of the proof is to show that whenever all the prefixed **fórmulas** on a **tableau** branch are satisfiable, any application of a rule results in at least one extended branch that is also satisfiable. Since closed branches are unsatisfiable, any **tableau** for a satisfiable set of prefixed **fórmulas** must have at least one open branch.

In order to apply this strategy in the modal case, we have to extend our definition of “satisfiable” to relational and prefixes. With that in hand, however, the proof is straightforward.

Definição 61.2. Let P be some set of prefixes, O.s., $P \subseteq (\mathbb{Z}^+)^* \setminus \{\Lambda\}$ and let $\mathfrak{M}$ be a model. A function $f: P \rightarrow W$ is an *interpretation of P in $\mathfrak{M}$* if, whenever σ and $\sigma.n$ are both in P , then $Rf(\sigma)f(\sigma.n)$.

Relative to an interpretation of prefixes P we can define:

1. $\mathfrak{M}$ satisfies $\sigma \mathbb{T} \varphi$ iff $\mathfrak{M}, f(\sigma) \Vdash \varphi$.
2. $\mathfrak{M}$ satisfies $\sigma \mathbb{F} \varphi$ iff $\mathfrak{M}, f(\sigma) \nVdash \varphi$.

Note that since R is reflexive and transitive and $\sigma.*$ denotes, $\sigma, \sigma.n_1, \sigma.n_1.n_2, \dots$, we also have that $Rf(\sigma)f(\sigma.*)$.

Definição 61.3. Let Γ be a set of prefixed **fórmulas**, and let $P(\Gamma)$ be the set of prefixes that occur in it. If f is an interpretation of $P(\Gamma)$ in $\mathfrak{M}$, we say that $\mathfrak{M}$ satisfies Γ with respect to f , $\mathfrak{M}, f \Vdash \Gamma$, if $\mathfrak{M}$ satisfies every prefixed **fórmula** in Γ with respect to f . Γ is *satisfiable* iff there is a model $\mathfrak{M}$ and interpretation f of $P(\Gamma)$ such that $\mathfrak{M}, f \Vdash \Gamma$.

Proposição 61.4. If Γ contains both $\sigma \mathbb{T} \varphi$ and $\sigma.* \mathbb{F} \varphi$ for some **fórmula** φ and prefix σ , or it contains $\sigma \mathbb{T} \perp$, then Γ is unsatisfiable.

Prova. Since always $\mathfrak{M}, f(\sigma) \nVdash \perp$, a Γ that contains $\mathbb{T} \perp$ is unsatisfiable.

There also cannot be a model $\mathfrak{M}$ and interpretation f of $P(\Gamma)$ such that both If $\mathfrak{M}, f(\sigma) \Vdash \varphi$, then by **Proposição 59.3**, since $Rf(\sigma)(\sigma.*)$, $\mathfrak{M}, f(\sigma) \Vdash \varphi$. So we cannot have both $\mathfrak{M}, f(\sigma) \Vdash \varphi$ and $\mathfrak{M}, f(\sigma.*) \nVdash \varphi$. $\square$

Teorema 61.5 (Soundness). If Γ has a closed **tableau**, Γ is unsatisfiable.

int:tab:sou:thm:tableau-soundness

61.4. SOUNDNESS FOR INTUITIONISTIC TABLEAUX

Prova. We call a branch of a **tableau** satisfiable iff the set of **fórmulas assinadas** on it is satisfiable, and let's call a **tableau** satisfiable if it contains at least one satisfiable branch.

We show the following: Extending a satisfiable **tableau** by one of the rules of inference always results in a satisfiable **tableau**. This will prove the theorem: any closed **tableau** results by applying rules of inference to the **tableau** consisting only of assumptions from Γ . So if Γ were satisfiable, any **tableau** for it would be satisfiable. A closed **tableau**, however, is clearly not satisfiable, since all its branches are closed and closed branches are unsatisfiable.

Suppose we have a satisfiable **tableau**, O.s., a **tableau** with at least one satisfiable branch. Applying a rule of inference either adds **fórmulas assinadas** to a branch, or splits a branch in two. If the **tableau** has a satisfiable branch which is not extended by the rule application in question, it remains a satisfiable branch in the extended **tableau**, so the extended tableau is satisfiable. So we only have to consider the case where a rule is applied to a satisfiable branch.

Let Γ be the set of **fórmulas assinadas** on that branch, and let $\sigma S\varphi \in \Gamma$ be the **fórmula assinada** to which the rule is applied. If the rule does not result in a split branch, we have to show that the extended branch, O.s., Γ together with the conclusions of the rule, is still satisfiable. If the rule results in split branch, we have to show that at least one of the two resulting branches is satisfiable.

1. The branch is expanded by applying $\neg\mathbb{T}$ to $\sigma\mathbb{T}\neg\psi \in \Gamma$. Then the extended branch contains the **fórmulas assinadas** $\Gamma \cup \{\sigma.*\mathbb{F}\psi\}$. Suppose $\mathfrak{M}, f \Vdash \Gamma$. In particular, $\mathfrak{M}, f(\sigma) \Vdash \neg\psi$. Thus, $\mathfrak{M}, w \not\Vdash \psi$ for any w such that $Rf(\sigma)w$, and that includes $f(\sigma.*)$. So, $\mathfrak{M}$ satisfies $\sigma.*\mathbb{F}\psi$ with respect to f .
2. The branch is expanded by applying $\neg\mathbb{F}$ to $\sigma\mathbb{F}\neg\psi \in \Gamma$: Exercise.
3. The branch is expanded by applying $\wedge\mathbb{T}$ to $\sigma\mathbb{T}\psi \wedge \chi \in \Gamma$, which results in two new **fórmulas assinadas** on the branch: $\sigma\mathbb{T}\psi$ and $\sigma\mathbb{T}\chi$. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular $\mathfrak{M}, f(\sigma) \Vdash \psi \wedge \chi$. Then $\mathfrak{M}, f(\sigma) \Vdash \psi$ and $\mathfrak{M}, f(\sigma) \Vdash \chi$. This means that $\mathfrak{M}$ satisfies both $\sigma\mathbb{T}\psi$ and $\sigma\mathbb{T}\chi$ with respect to f .
4. The branch is expanded by applying $\vee\mathbb{F}$ to $\mathbb{F}\psi \vee \chi \in \Gamma$: Exercise.
5. The branch is expanded by applying $\rightarrow\mathbb{F}$ to $\sigma\mathbb{F}\psi \rightarrow \chi \in \Gamma$: This results in two new **fórmulas assinadas** on the branch: $\sigma.n\mathbb{T}\psi$ and $\sigma.n\mathbb{F}\chi$, where $\sigma.n$ is a new prefix on the branch, O.s., $\sigma.n \notin P(\Gamma)$.

Since Γ is satisfiable, there is a $\mathfrak{M}$ and interpretation f of $P(\Gamma)$ such that $\mathfrak{M}, f \Vdash \Gamma$, in particular $\mathfrak{M}, f(\sigma) \not\Vdash \psi \rightarrow \chi$. We have to show that $\Gamma \cup \{\sigma.n\mathbb{F}\psi \rightarrow \chi\}$ is satisfiable. To do this, we define an interpretation of $P(\Gamma) \cup \{\sigma.n\}$ as follows:

Since $\mathfrak{M}, f(\sigma) \not\Vdash \psi \rightarrow \chi$, there is a $w \in W$ such that $Rf(\sigma)w$ such that $\mathfrak{M}, w \Vdash \psi$ and $\mathfrak{M}, w \not\Vdash \chi$. Let f' be like f , except that $f'(\sigma.n) = w$. Since

$f'(\sigma) = f(\sigma)$ and $Rf(\sigma)w$, we have $Rf'(\sigma)f'(\sigma.n)$, so f' is an interpretation of $P(\Gamma) \cup \{\sigma.n\}$. Obviously $\mathfrak{M}, f'(\sigma.n) \Vdash \psi$ and $\mathfrak{M}, f'(\sigma.n) \nVdash \chi$. Since $f(\sigma') = f'(\sigma')$ for all prefixes $\sigma' \in P(\Gamma)$, $\mathfrak{M}, f' \Vdash \Gamma$. So, $\mathfrak{M}, f'$ satisfies $\Gamma \cup \{\sigma.n \Vdash \psi \rightarrow \chi\}$

Now let's consider the possible inferences with two premises.

1. The branch is expanded by applying $\wedge\mathbb{F}$ to $\sigma \mathbb{F} \psi \wedge \chi \in \Gamma$, which results in two branches, a left one continuing through $\sigma \mathbb{F} \psi$ and a right one through $\sigma \mathbb{F} \chi$. Suppose $\mathfrak{M}, f \Vdash \Gamma$, in particular $\mathfrak{M}, f(\sigma) \nVdash \psi \wedge \chi$. Then $\mathfrak{M}, f(\sigma) \nVdash \psi$ or $\mathfrak{M}, f(\sigma) \nVdash \chi$. In the former case, $\mathfrak{M}, f$ satisfies $\sigma \mathbb{F} \psi$, O.s., the left branch is satisfiable. In the latter, $\mathfrak{M}, f$ satisfies $\sigma \mathbb{F} \chi$, O.s., the right branch is satisfiable.
2. The branch is expanded by applying $\vee\mathbb{T}$ to $\sigma \mathbb{T} \psi \vee \chi \in \Gamma$: Exercise.
3. The branch is expanded by applying $\rightarrow\mathbb{T}$ to $\sigma \mathbb{T} \psi \rightarrow \chi \in \Gamma$: Exercise. $\square$

Problema 61.2. Complete the proof of [Teorema 61.5](#).

Corolário 61.6. *If $\Gamma \vdash \varphi$ then $\Gamma \models \varphi$.*

*int:tab:sou:
cor:entailment-soundness*

Prova. If $\Gamma \vdash \varphi$ then for some $\psi_1, \dots, \psi_n \in \Gamma$, $\Delta = \{1 \mathbb{F} \varphi, 1 \mathbb{T} \psi_1, \dots, 1 \mathbb{T} \psi_n\}$ has a closed [tableau](#). We want to show that $\Gamma \models \varphi$. Suppose not, so for some $\mathfrak{M}$ and w , $\mathfrak{M}, w \Vdash \psi_i$ for $i = 1, \dots, n$, but $\mathfrak{M}, w \nVdash \varphi$. Let $f(1) = w$; then f is an interpretation of $P(\Delta)$ into $\mathfrak{M}$, and $\mathfrak{M}$ satisfies Δ with respect to f . But by [Teorema 61.5](#), Δ is unsatisfiable since it has a closed [tableau](#), a contradiction. So we must have $\Gamma \vdash \varphi$ after all. $\square$

Corolário 61.7. *If $\vdash \varphi$ then φ is true in all models.*

*int:tab:sou:
cor:weak-soundness*

Capítulo 62

Propositions as Types

62.1. INTRODUCTION

This is a *very experimental* draft of a chapter on the Curry–Howard correspondence. It needs more explanation and motivation, and there are probably errors and omissions. The proof of normalization should be reviewed and expanded. There are no examples for the product type. Permutation and simplification conversions are not covered. It will make a lot more sense once there is also material on the (typed) lambda calculus which is basically presupposed here. Use with extreme caution.

-NoValue-

`content/intuitionistic-logic/propositions-as-types/introduction.tex`

62.1 Introduction

`int:pty:int:`
`sec`

Historically the lambda calculus and intuitionistic logic were developed separately. Haskell Curry and William Howard independently discovered a close similarity: types in a typed lambda calculus correspond to formulas in intuitionistic logic in such a way that a *derivação* of a *fórmula* corresponds directly to a typed lambda term with that *fórmula* as its type. Moreover, beta reduction in the typed lambda calculus corresponds to certain transformations of *derivações*.

For instance, a *derivação* of $\varphi \rightarrow \psi$ corresponds to a term $\lambda x^\varphi. N^\psi$, which has the function type $\varphi \rightarrow \psi$. The inference rules of natural deduction correspond to typing rules in the typed lambda calculus, p.ex.,

$$\begin{array}{c} [\varphi]^x \\ \vdots \\ \psi \\ x \frac{}{\varphi \rightarrow \psi} \rightarrow\text{Intro} \end{array} \quad \text{corresponds to} \quad \frac{x : \varphi \Rightarrow N : \psi}{\Rightarrow \lambda x^\varphi. N^\psi : \varphi \rightarrow \psi} \lambda$$

where the rule on the right means that if x is of type φ and N is of type ψ , then $\lambda x^\varphi. N$ is of type $\varphi \rightarrow \psi$.

The $\rightarrow\text{Elim}$ rule corresponds to the typing rule for composition terms, O.s.,

$$\frac{\frac{\varphi \rightarrow \psi \quad \varphi}{\psi} \rightarrow\text{Elim} \quad \frac{}{\Rightarrow P : \varphi \rightarrow \psi} \quad \frac{}{\Rightarrow Q : \varphi} \text{app}}{\Rightarrow P^{\varphi \rightarrow \psi} Q^\varphi : \psi} \text{corresponds to}$$

If a $\rightarrow$ Intro rule is followed immediately by a $\rightarrow$ Elim rule, the *derivação* can be simplified:

$$\begin{array}{ccc}
 \begin{array}{c} [\varphi]^x \\ \vdots \\ \psi \\ \hline \varphi \rightarrow \psi \end{array} & \xrightarrow{\rightarrow\text{Intro}} & \begin{array}{c} \vdots \\ \varphi \end{array} \\
 \hline \psi & \xrightarrow{\rightarrow\text{Elim}} & \begin{array}{c} \vdots \\ \varphi \\ \vdots \\ \psi \end{array}
 \end{array}$$

which corresponds to the beta reduction of lambda terms

$$(\lambda x^\varphi. P^\psi)Q \rightarrow P[Q/x].$$

Similar correspondences hold between the rules for $\wedge$ and “product” types, and between the rules for $\vee$ and “sum” types.

This correspondence between terms in the simply typed lambda calculus and natural deduction *derivações* is called the “Curry–Howard”, or “propositions as types” correspondence. In addition to *fórmulas* (propositions) corresponding to types, and proofs to terms, we can summarize the correspondences as follows:

logic	program
proposition	type
proof	term
assumption	variable
discharged assumption	bind variable
not discharged assumption	free variable
implication	function type
conjunction	product type
disjunction	sum type
absurdity	bottom type

The Curry–Howard correspondence is one of the cornerstones of automated proof assistants and type checkers for programs, since checking a proof witnessing a proposition (as we did above) amounts to checking if a program (term) has the declared type.

-NoValue-

`content/intuitionistic-logic/propositions-as-types/sequent-natural-deduction.tex`

62.2 Sequent Natural Deduction

Let us write $\Gamma \Rightarrow \varphi$ if there is a natural deduction *derivação* with Γ as *não descarregada* assumptions and φ as conclusion; or $\Rightarrow \varphi$ if Γ is empty.

We write $\Gamma, \varphi_1, \dots, \varphi_n$ for $\Gamma \cup \{\varphi_1, \dots, \varphi_n\}$, and Γ, Δ for $\Gamma \cup \Delta$.

62.2. SEQUENT NATURAL DEDUCTION

Observe that when we have $\Gamma \Rightarrow \varphi \wedge \psi$, meaning we have a *derivação* with Γ as *não descarregada* assumptions and $\varphi \wedge \psi$ as end-fórmula, then by applying $\wedge$ Elim at the bottom, we can get a *derivação* with the same *não descarregada* assumptions and φ as conclusion. In other words, if $\Gamma \Rightarrow \varphi \wedge \psi$, then $\Gamma \Rightarrow \varphi$.

$$\frac{\Gamma \Rightarrow \varphi \wedge \psi}{\Gamma \Rightarrow \varphi} \wedge\text{Elim} \qquad \frac{\Gamma \Rightarrow \varphi \wedge \psi}{\Gamma \Rightarrow \psi} \wedge\text{Elim}$$

The label $\wedge$ Elim hints at the relation with the rule of the same name in natural deduction.

Likewise, suppose we have $\Gamma, \varphi \Rightarrow \psi$, meaning we have a *derivação* with *não descarregada* assumptions Γ, φ and end-fórmula ψ . If we apply the $\rightarrow$ Intro rule, we have a *derivação* with Γ as *não descarregada* assumptions and $\varphi \rightarrow \psi$ as the end-fórmula, O.s., $\Gamma \Rightarrow \varphi \rightarrow \psi$. Note how this has made the *descarrega* of assumptions more explicit.

$$\frac{\Gamma, \varphi \Rightarrow \psi}{\Gamma \Rightarrow \varphi \rightarrow \psi} \rightarrow\text{Intro}$$

We can draw conclusions from other rules in the same fashion, which is spelled out as follows:

$$\begin{array}{c} \frac{\Gamma \Rightarrow \varphi \quad \Delta \Rightarrow \psi}{\Gamma, \Delta \Rightarrow \varphi \wedge \psi} \wedge\text{Intro} \\ \frac{\Gamma \Rightarrow \varphi \wedge \psi}{\Gamma \Rightarrow \varphi} \wedge\text{Elim}_1 \qquad \frac{\Gamma \Rightarrow \varphi \wedge \psi}{\Gamma \Rightarrow \psi} \wedge\text{Elim}_2 \\ \frac{\Gamma \Rightarrow \varphi}{\Gamma \Rightarrow \varphi \vee \psi} \vee\text{Intro}_1 \qquad \frac{\Gamma \Rightarrow \psi}{\Gamma \Rightarrow \varphi \vee \psi} \vee\text{Intro}_2 \\ \frac{\Gamma \Rightarrow \varphi \vee \psi \quad \Delta, \varphi \Rightarrow \chi \quad \Delta', \psi \Rightarrow \chi}{\Gamma, \Delta, \Delta' \Rightarrow \chi} \vee\text{Elim} \\ \frac{\Gamma, \varphi \Rightarrow \psi}{\Gamma \Rightarrow \varphi \rightarrow \psi} \rightarrow\text{Intro} \qquad \frac{\Delta \Rightarrow \varphi \rightarrow \psi \quad \Gamma \Rightarrow \varphi}{\Gamma, \Delta \Rightarrow \psi} \rightarrow\text{Elim} \\ \frac{\Gamma \Rightarrow \perp}{\Gamma \Rightarrow \varphi} \perp_I \end{array}$$

Any assumption by itself is a *derivação* of φ from φ , O.s., we always have $\varphi \Rightarrow \varphi$.

$$\overline{\varphi \Rightarrow \varphi}$$

Together, these rules can be taken as a calculus about what natural deduction *derivações* exist. They can also be taken as a notational variant of natural deduction, in which each step records not only the *fórmula derivad* but also the *não descarregada* assumptions from which it was *derivad*.

$$\begin{array}{c}
\frac{\varphi \Rightarrow \varphi}{\varphi \Rightarrow \varphi \vee (\varphi \rightarrow \perp)} \quad \psi \Rightarrow \psi \\
\hline
\varphi, \psi \rightarrow \Rightarrow \perp \\
\hline
\frac{(\psi \Rightarrow \varphi \rightarrow \perp)}{(\psi \Rightarrow \varphi \vee (\varphi \rightarrow \perp))} \quad (\psi \Rightarrow \psi) \\
\hline
\frac{(\psi \Rightarrow \perp)}{\Rightarrow \psi \rightarrow \perp}
\end{array}$$

where ψ is short for $(\varphi \vee (\varphi \rightarrow \perp)) \rightarrow \perp$.

-NoValue-

[content/intuitionistic-logic/propositions-as-types/proof-terms.tex](#)

62.3 Proof Terms

We give the definition of proof terms, and then establish its relation with [int:pty:ter:sec](#) natural deduction [derivações](#).

Definição 62.1 (Proof terms). Proof terms are inductively generated by the following rules:

1. A single variable x is a proof term.
2. If P and Q are proof terms, then PQ is also a proof term.
3. If x is a variable, φ is a [fórmula](#), and N is a proof term, then $\lambda x^\varphi. N$ is also a proof term.
4. If P and Q are proof terms, then $\langle P, Q \rangle$ is a proof term.
5. If M is a proof term, then $p_i(M)$ is also a proof term, where i is 1 or 2.
6. If M is a proof term, and φ is a formula, then $\text{in}_i^\varphi(M)$ is a proof term, where i is 1 or 2.
7. If M, N_1, N_2 is proof terms, and x_1, x_2 are variables, then $\text{case}(M, x_1.N_1, x_2.N_2)$ is a proof term.
8. If M is a proof term and φ is a formula, then $\text{contr}_\varphi(M)$ is proof term.

Each of the above rules corresponds to an inference rule in natural deduction. Thus we can inductively assign proof terms to the [fórmulas](#) in a [derivação](#). To make this assignment unique, we must distinguish between the two versions of $\wedge$ Elim and of $\vee$ Intro. For instance, the proof terms assigned to the conclusion of $\vee$ Intro must carry the information whether $\varphi \vee \psi$ is inferred from φ or from ψ . Suppose M is the term assigned to φ from which $\varphi \vee \psi$ is inferred. Then the proof term assigned to $\varphi \vee \psi$ is $\text{in}_1^\varphi(M)$. If we instead infer $\psi \vee \varphi$ then the proof term assigned is $\text{in}_2^\varphi(M)$.

62.4. CONVERTING DERIVAÇÕES TO PROOF TERMS

The term $\lambda x^\varphi. N$ is assigned to the conclusion of $\rightarrow$ Intro. The φ represents the assumption being discharged; only have we included it can we infer the formula of $\lambda x^\varphi. N$ based on the formula of N .

Definição 62.2 (Typing context). A *typing context* is a mapping from variables to formulas. We will call it simply the “context” if there is no confusion. We write a context Γ as a set of pairs $\langle x, \varphi \rangle$.

A pair $\Gamma \Rightarrow M$ where M is a proof term represents a *derivação* of a formula with context Γ .

Definição 62.3 (Typing pair). A *typing pair* is a pair $\langle \Gamma, M \rangle$, where Γ is a typing context and M is a proof term.

Since in general terms only make sense with specific contexts, we will speak simply of “terms” from now on instead of “typing pair”; and it will be apparent when we are talking about the literal term M .

-NoValue-

content/intuitionistic-logic/propositions-as-types/proofs-to-terms.tex

62.4 Converting Derivações to Proof Terms

int:pty:pt:
sec

We will describe the process of converting natural deduction *derivações* to pairs. We will write a proof term to the left of each formula in the *derivação*, resulting in expressions of the form $M : \varphi$. We’ll then say that, M *witnesses* φ . Let’s call such an expression a *judgment*.

First let us assign to each assumption a variable, with the following constraints:

1. Assumptions *descarregada* in the same step (that is, with the same number on the square bracket) must be assigned the same variable.
2. For assumptions not *descarregada*, assumptions of different *fórmulas* should be assigned different variables.

Such an assignment translates all assumptions of the form

$$\varphi \quad \text{into} \quad x : \varphi.$$

With assumptions all associated with variables (which are terms), we can now inductively translate the rest of the deduction tree. The modified natural deduction rules taking into account context and proof terms are given below. Given the proof terms for the premise(s), we obtain the corresponding proof term for conclusion.

$$\frac{\frac{M_1 : \varphi_1 \quad M_2 : \varphi_2}{\langle M_1, M_2 \rangle : \varphi_1 \wedge \varphi_2} \wedge \text{Intro}}{\frac{M : \varphi_1 \wedge \varphi_2}{p_i(M) : \varphi_1} \wedge \text{Elim}_1 \quad \frac{M : \varphi_1 \wedge \varphi_2}{p_i(M) : \varphi_2} \wedge \text{Elim}_2}$$

In $\wedge \text{Intro}$ we assume we have φ_1 witnessed by term M_1 and φ_2 witnessed by term M_2 . We pack up the two terms into a pair $\langle M_1, M_2 \rangle$ which witnesses $\varphi_1 \wedge \varphi_2$.

In $\wedge \text{Elim}_i$ we assume that M witnesses $\varphi_1 \wedge \varphi_2$. The term witnessing φ_i is $p_i(M)$. Note that M is not necessary of the form $\langle M_1, M_2 \rangle$, so we cannot simply assign M_1 to the conclusion φ_i .

Note how this coincides with the BHK interpretation. What the BHK interpretation does not specify is how the function used as proof for $\varphi \rightarrow \psi$ is supposed to be obtained. If we think of proof terms as proofs or functions of proofs, we can be more explicit.

$$\frac{\frac{\begin{array}{c} [x : \varphi] \\ \vdots \\ N : \psi \end{array}}{\lambda x^\varphi. N : \varphi \rightarrow \psi} \rightarrow \text{Intro}}{\frac{P : \varphi \rightarrow \psi \quad Q : \varphi}{PQ : \psi} \rightarrow \text{Elim}}$$

The λ notation should be understood as the same as in the lambda calculus, and PQ means applying P to Q .

$$\frac{\frac{M_1 : \varphi_1}{\text{in}_1^{\varphi_1}(M_1) : \varphi_1 \vee \varphi_2} \vee \text{Intro}_1 \quad \frac{M_2 : \varphi_2}{\text{in}_2^{\varphi_2}(M_2) : \varphi_1 \vee \varphi_2} \vee \text{Intro}_2}{\frac{\begin{array}{ccc} [x_1 : \varphi_1] & & [x_2 : \varphi_2] \\ \vdots & & \vdots \\ M : A_1 \vee \varphi_2 & N_1 : \chi & N_2 : \chi \end{array}}{\text{case}(M, x_1.N_1, x_2.N_2) : \chi} \vee \text{Elim}}$$

The proof term $\text{in}_1^{\varphi_1}(M_1)$ is a term witnessing $\varphi_1 \vee \varphi_2$, where M_1 witnesses φ_1 .

The term $\text{case}(M, x_1.N_1, x_2.N_2)$ mimics the case clause in programming languages: we already have the *derivação* of $\varphi \vee \psi$, a *derivação* of χ assuming φ , and a *derivação* of χ assuming ψ . The *case* operator thus select the appropriate proof depending on M ; either way it's a proof of χ .

$$\frac{N : \perp}{\text{contr}_\varphi(N) : \varphi} \perp_I$$

$\text{contr}_\varphi(N)$ is a term witnessing φ , whenever N is a term witnessing $\perp$.

62.4. CONVERTING DERIVAÇÕES TO PROOF TERMS

Now we have a natural deduction *derivação* with all formulas associated with a term. At each step, the relevant typing context Γ is given by the list of assumptions remaining *não descarregada* at that step. Note that Γ is well defined: since we have forbidden assumptions of different *não descarregada* assumptions to be assigned the same variable, there won't be any disagreement about the formulas mapped to which a variable is mapped.

We now give some examples of such translations:

Consider the *derivação* of $\neg\neg(\varphi \vee \neg\varphi)$, O.s., $((\varphi \vee (\varphi \rightarrow \perp)) \rightarrow \perp) \rightarrow \perp$. Its translation is:

$$\begin{array}{c}
 \frac{[x : \varphi]^1}{\frac{[y : (\varphi \vee (\varphi \rightarrow \perp)) \rightarrow \perp]^2}{\frac{1}{\frac{y(\text{in}_1^{\varphi \rightarrow \perp}(x)) : \perp}{\lambda x^\varphi. y(\text{in}_1^{\varphi \rightarrow \perp}(x)) : \varphi \rightarrow \perp}} \text{in}_1^{\varphi \rightarrow \perp}(x) : \varphi \vee (\varphi \rightarrow \perp)} \\
 \frac{[y : (\varphi \vee (\varphi \rightarrow \perp)) \rightarrow \perp]^2}{\frac{2}{\frac{y(\text{in}_2^\varphi(\lambda x^\varphi. y(\text{in}_1^{\varphi \rightarrow \perp}(x))) : \perp}{\lambda y^{(\varphi \vee (\varphi \rightarrow \perp)) \rightarrow \perp}. y(\text{in}_2^\varphi(\lambda x^\varphi. y(\text{in}_1^{\varphi \rightarrow \perp}(x))) : ((\varphi \vee (\varphi \rightarrow \perp)) \rightarrow \perp) \rightarrow \perp}} \text{in}_2^\varphi(\lambda x^\varphi. y(\text{in}_1^{\varphi \rightarrow \perp}(x))) : \varphi \vee (\varphi \rightarrow \perp)}
 \end{array}$$

The tree has no assumptions, so the context is empty; we get:

$$\vdash \lambda y^{(\varphi \vee (\varphi \rightarrow \perp)) \rightarrow \perp}. y(\text{in}_2^\varphi(\lambda x^\varphi. y(\text{in}_1^{\varphi \rightarrow \perp}(x))) : ((\varphi \vee (\varphi \rightarrow \perp)) \rightarrow \perp) \rightarrow \perp$$

If we leave out the last $\rightarrow$ Intro, the assumptions denoted by y would be in the context and we would get:

$$y : ((\varphi \vee (\varphi \rightarrow \perp)) \rightarrow \perp) \vdash y(\text{in}_2^\varphi(\lambda x^\varphi. y(\text{in}_1^{\varphi \rightarrow \perp}(x))) : \perp$$

Another example: $\vdash \varphi \rightarrow (\varphi \rightarrow \perp) \rightarrow \perp$

$$\begin{array}{c}
 \frac{[x : \varphi]^2}{\frac{1}{\frac{yx : \perp}{\lambda y^{\varphi \rightarrow \perp}. yx : (\varphi \rightarrow \perp) \rightarrow \perp}} [y : \varphi \rightarrow \perp]^1 \\
 \frac{2}{\lambda x^\varphi. \lambda y^{\varphi \rightarrow \perp}. yx : \varphi \rightarrow (\varphi \rightarrow \perp) \rightarrow \perp}
 \end{array}$$

Again all assumptions are *descarregada* and thus the context is empty, the resulting term is

$$\vdash \lambda x^\varphi. \lambda y^{\varphi \rightarrow \perp}. yx : \varphi \rightarrow (\varphi \rightarrow \perp) \rightarrow \perp$$

If we leave out the last two $\rightarrow$ Intro inferences, the assumptions denoted by both x and y would be in context and we would get

$$x : \varphi, y : \varphi \rightarrow \perp \vdash yx : \perp$$

-NoValue-

[content/intuitionistic-logic/propositions-as-types/terms-to-proofs.tex](#)

62.5 Recovering Derivações from Proof Terms

Now let us consider the other direction: translating terms back to natural deduction trees. We will still use the double refutation of the excluded middle as example, and let S denote this term, O.s., int:pty:tp:
sec

$$\lambda y^{(\varphi \vee (\varphi \rightarrow \perp)) \rightarrow \perp}. y(\text{in}_2^\varphi(\lambda x^\varphi. y \text{in}_1^{\varphi \rightarrow \perp}(x))) : ((\varphi \vee (\varphi \rightarrow \perp)) \rightarrow \perp) \rightarrow \perp$$

For each natural deduction rule, the term in the conclusion is always formed by wrapping some operator around the terms assigned to the premise(s). Rules correspond uniquely to such operators. For example, from the structure of the S we infer that the last rule applied must be $\rightarrow$ Intro, since it is of the form $\lambda y \dots$. $\dots$, and the λ operator corresponds to $\rightarrow$ Intro. In general we can recover the skeleton of the **derivação** solely by the structure of the term, p.ex.,

$$\begin{array}{c} \frac{[x]^1}{\text{in}_1^{\varphi \rightarrow \perp}(x) :} \text{VIntro}_1 \\ \frac{[y :]^2 \quad \text{in}_1^{\varphi \rightarrow \perp}(x) :}{y(\text{in}_1^{\varphi \rightarrow \perp}(x)) :} \rightarrow \text{Elim} \\ \frac{1 \quad y(\text{in}_1^{\varphi \rightarrow \perp}(x)) :}{\lambda x^\varphi. y(\text{in}_1^{\varphi \rightarrow \perp}(x)) :} \rightarrow \text{Intro} \\ \frac{[y :]^2 \quad \text{in}_2^\varphi(\lambda x^\varphi. y \text{in}_1^{\varphi \rightarrow \perp}(x)) :}{y(\text{in}_2^\varphi(\lambda x^\varphi. y \text{in}_1^{\varphi \rightarrow \perp}(x))) :} \text{VIntro}_2 \\ \frac{2 \quad y(\text{in}_2^\varphi(\lambda x^\varphi. y \text{in}_1^{\varphi \rightarrow \perp}(x))) :}{\lambda y^{(\varphi \vee (\varphi \rightarrow \perp)) \rightarrow \perp}. y(\text{in}_2^\varphi(\lambda x^\varphi. y(\text{in}_1^{\varphi \rightarrow \perp}(x)))) :} \rightarrow \text{Intro} \end{array}$$

Our next step is to recover the **fórmulas** these terms witness. We define a function $F(\Gamma, M)$ which denotes the **fórmula** witnessed by M in context Γ , by induction on M as follows:

$$\begin{aligned} F(\Gamma, x) &= \Gamma(x) \\ F(\Gamma, \langle N_1, N_2 \rangle) &= F(\Gamma, N_1) \wedge F(\Gamma, N_2) \\ F(\Gamma, \text{p}_i(N)) &= \varphi_i \text{ if } F(\Gamma, N) = \varphi_1 \wedge \varphi_2 \\ F(\Gamma, \text{in}_i^\varphi(N)) &= \begin{cases} F(N) \vee \varphi & \text{if } i = 1 \\ \varphi \vee F(N) & \text{if } i = 2 \end{cases} \\ F(\Gamma, \text{case}(M, x_1.N_1, x_2.N_2)) &= F(\Gamma \cup \{x_i : F(\Gamma, M)\}, N_i) \\ F(\Gamma, \lambda x^\varphi. N) &= \varphi \rightarrow F(\Gamma \cup \{x : \varphi\}, N) \\ F(\Gamma, NM) &= \psi \text{ if } F(\Gamma, N) = \varphi \rightarrow \psi \end{aligned}$$

where $\Gamma(x)$ means the formula mapped to by x in Γ and $\Gamma \cup \{x : \varphi\}$ is a context exactly as Γ except mapping x to φ , whether or not x is already in Γ .

Note there are cases where $F(\Gamma, M)$ is not defined, for example:

1. In the first line, it is possible that x is not in Γ .
2. In recursive cases, the inner invocation may be undefined, making the outer one undefined too.

62.5. RECOVERING DERIVAÇÕES FROM PROOF TERMS

3. In the third line, its only defined when $F(\Gamma, M)$ is of the form $\varphi_1 \vee \varphi_2$, and the right hand is independent on i .

As we recursively compute $F(\Gamma, M)$, we work our way up the natural deduction *derivação*. The every step in the computation of $F(\Gamma, M)$ corresponds to a term in the *derivação* to which the *derivação*-to-term translation assigns M , and the formula computed is the end-*fórmula* of the *derivação*. However, the result may not be defined for some choices of Γ . We say that such pairs $\langle \Gamma, M \rangle$ are *ill-typed*, and otherwise *well-typed*. However, if the term M results from translating a *derivação*, and the *fórmulas* in Γ correspond to the *não descarregada* assumptions of the *derivação*, the pair $\langle \Gamma, M \rangle$ will be well-typed.

Proposição 62.4. *If D is a *derivação* with *não descarregada* assumptions $\varphi_1, \dots, \varphi_n$, M is the proof term associated with D and $\Gamma = \{x_1 : \varphi_1, \dots, x_n : \varphi_n\}$, then the result of recovering *derivação* from M in context Γ is D .*

In the other direction, if we first translate a typing pair to natural deduction and then translate it back, we won't get the same pair back since the choice of variables for the *não descarregada* assumptions is underdetermined. For example, consider the pair $\langle \{x : \varphi, y : \varphi \rightarrow \psi\}, yx \rangle$. The corresponding *derivação* is

$$\frac{\varphi \rightarrow \psi \quad \varphi}{\psi} \rightarrow\text{Elim}$$

By assigning different variables to the *não descarregada* assumptions, say, u to $\varphi \rightarrow \psi$ and v to φ , we would get the term uv rather than yx . There is a connection, though: the terms will be the same up to renaming of variables.

Now we have established the correspondence between typing pairs and natural deduction, we can prove theorems for typing pairs and transfer the result to natural deduction *derivações*.

Similar to what we did in the natural deduction section, we can make some observations here too. Let $\Gamma \vdash M : \varphi$ denote that there is a pair (Γ, M) witnessing the formula φ . Then always $\Gamma \vdash x : \varphi$ if $x : \varphi \in \Gamma$, and the following rules are valid:

$$\begin{array}{c} \frac{\Gamma \vdash M_1 : \varphi_1 \quad \Delta \vdash M_2 : \varphi_2}{\Gamma, \Delta \vdash \langle M_1, M_2 \rangle : \varphi_1 \wedge \varphi_2} \wedge\text{Intro} \quad \frac{\Gamma \vdash M : \varphi_1 \wedge \varphi_2}{\Gamma \vdash p_i(M) : \varphi_i} \wedge\text{Elim}_i \\ \frac{\Gamma \vdash M_1 : \varphi_1}{\Gamma \vdash \text{in}_1^{\varphi_2}(M) : \varphi_1 \vee \varphi_2} \vee\text{Intro}_1 \quad \frac{\Gamma \vdash M_2 : \varphi_2}{\Gamma \vdash \text{in}_2^{\varphi_1}(M) : \varphi_1 \vee \varphi_2} \vee\text{Intro}_2 \\ \frac{\Gamma \vdash M : \varphi \vee \psi \quad \Delta_1, x_1 : \varphi_1 \vdash N_1 : \chi \quad \Delta_2, x_2 : \varphi_2 \vdash N_2 : \chi}{\Gamma, \Delta, \Delta' \vdash \text{case}(M, x_1.N_1, x_2.N_2) : \chi} \vee\text{Elim} \\ \frac{\Gamma, x : \varphi \vdash N : \psi}{\Gamma \vdash \lambda x^\varphi. N : \varphi \rightarrow \psi} \rightarrow\text{Intro} \quad \frac{\Gamma \vdash Q : \varphi \quad \Delta \vdash P : \varphi \rightarrow \psi}{\Gamma, \Delta \vdash PQ : \psi} \rightarrow\text{Elim} \\ \frac{\Gamma \vdash M : \perp}{\Gamma \vdash \text{contr}_\varphi(M) : \varphi} \perp\text{Elim} \end{array}$$

These are the typing rules of the simply typed lambda calculus extended with product, sum and bottom.

In addition, the $F(\Gamma, M)$ is actually a type checking algorithm; it returns the type of the term with respect to the context, or is undefined if the term is ill-typed with respect to the context.

-NoValue-

`content/intuitionistic-logic/propositions-as-types/reduction.tex`

62.6 Reduction

In natural deduction `derivações`, an introduction rule that is followed by an `int:pty:red:sec` elimination rule is redundant. For instance, the `derivação`

$$\frac{\varphi \quad \varphi \rightarrow \psi}{\psi} \rightarrow\text{Elim} \quad \frac{[\chi]}{\psi \wedge \chi} \wedge\text{Intro} \quad \frac{\psi \wedge \chi}{\psi} \wedge\text{Elim} \quad \frac{\psi}{\chi \rightarrow \psi} \rightarrow\text{Intro}$$

can be replaced with the simpler `derivação`:

$$\frac{\varphi \quad \varphi \rightarrow \psi}{\psi} \rightarrow\text{Elim} \quad \frac{\psi}{\chi \rightarrow \psi} \rightarrow\text{Intro}$$

As we see, an $\wedge\text{Intro}$ followed by $\wedge\text{Elim}$ “cancel out.” In general, we see that the conclusion of $\wedge\text{Elim}$ is always the formula on one side of the conjunction, and the premises of $\wedge\text{Intro}$ requires both sides of the conjunction, thus if we need a `derivação` of either side, we can simply use that `derivação` without introducing the conjunction followed by eliminating it.

Thus in general we have

$$\frac{\frac{\frac{\vdots D_1}{\varphi_1} \quad \frac{\vdots D_2}{\varphi_2}}{\varphi_1 \wedge \varphi_2} \wedge\text{Intro} \quad \frac{\varphi_1 \wedge \varphi_2}{\varphi_i} \wedge\text{Elim}_i}{\varphi_i} \rightarrow \frac{\vdots D_i}{\varphi_i}$$

The $\rightarrow$ symbol has a similar meaning as in the lambda calculus, O.s., a single step of a reduction. In the proof term syntax for `derivações`, the above reduction rule thus becomes:

$$(\Gamma, p_i \langle M_1^{\varphi_1}, M_2^{\varphi_2} \rangle) \rightarrow (\Gamma, M_i)$$

In the typed lambda calculus, this is the beta reduction rule for the product type.

62.6. REDUCTION

Note the type annotation on M_1 and M_2 : while in the standard term syntax only $\lambda x^\varphi. N$ has such notion, we reuse the notation here to remind us of the formula the term is associated with in the corresponding natural deduction *derivação*, to reveal the correspondence between the two kinds of syntax.

In natural deduction, a pair of inferences such as those on the left, O.s., a pair that is subject to cancelling is called a *cut*. In the typed lambda calculus the term on the left of $\rightarrow$ is called a *redex*, and the term to the right is called the *reductum*. Unlike untyped lambda calculus, where only $(\lambda x. N)Q$ is considered to be redex, in the typed lambda calculus the syntax is extended to terms involving $\langle N, M \rangle$, $p_i(N)$, $\text{in}_i^\varphi(N)$, $\text{case}(N, x_1.M_1, x_2.M_2)$, and $\text{contr}_N()$, with corresponding redexes.

Similarly we have reduction for disjunction:

$$\begin{array}{c}
 \begin{array}{c} \vdots \\ D \end{array} \\
 \hline
 \varphi_i \\
 \hline
 \varphi_1 \vee \varphi_2 \quad \vee\text{Intro} \\
 \hline
 \chi
 \end{array}
 \quad
 \begin{array}{c}
 [\varphi_1]^u \\
 \vdots \\
 D_1 \\
 \vdots \\
 \chi
 \end{array}
 \quad
 \begin{array}{c}
 [\varphi_2]^u \\
 \vdots \\
 D_2 \\
 \vdots \\
 \chi
 \end{array}
 \quad \vee\text{Elim} \rightarrow
 \begin{array}{c}
 \vdots \\
 D \\
 \vdots \\
 \varphi_i \\
 \vdots \\
 D_i \\
 \vdots \\
 \chi
 \end{array}$$

This corresponds to a reduction on proof terms:

$$(\Gamma, \text{case}(\text{in}_i^{\varphi_i}(M^{\varphi_i}), x_1^{\varphi_1}.N_1^\chi, x_2^{\varphi_2}.N_2^\chi)) \rightarrow (\Gamma, N_i^\chi[M^{\varphi_i}/x_i^{\varphi_i}])$$

This is the beta reduction rule of for sum types. Here, $M[N/x]$ means replacing all assumptions denoted by variable x in M with N ,

It would be nice if we pass the context Γ to the substitution function so that it can check if the substitution makes sense. For example, $xy[ab/y]$ does not make sense under the context $\{x : \varphi \rightarrow \theta, y : \varphi, a : \psi \rightarrow \chi, b : \psi\}$ since then we would be substituting a *derivação* of χ where a *derivação* of φ is expected. However, as long as our usage of substitution is careful enough to avoid such errors, we won't have to worry about such conflicts. Thus we can define it recursively as we did for untyped lambda calculus as if we are dealing with untyped terms.

Finally, the reduction of the function type corresponds to removal of a detour of a $\rightarrow$ Intro followed by a $\rightarrow$ Elim.

$$\begin{array}{c}
 [\varphi]^u \\
 \vdots \\
 D \\
 \vdots \\
 \psi \\
 \hline
 \varphi \rightarrow \psi \quad \rightarrow\text{Intro} \\
 \hline
 \psi
 \end{array}
 \quad
 \begin{array}{c}
 \vdots \\
 D' \\
 \vdots \\
 \varphi
 \end{array}
 \quad \rightarrow\text{Elim} \rightarrow
 \begin{array}{c}
 \vdots \\
 D' \\
 \vdots \\
 \varphi \\
 \vdots \\
 D \\
 \vdots \\
 \psi
 \end{array}$$

For proof terms, this amounts to ordinary beta reduction:

$$(\Gamma, (\lambda x^\varphi. N^\psi)Q^\varphi) \rightarrow (\Gamma, N^\psi[Q^\varphi/x^\varphi])$$

Absurdity has only an elimination rule and no introduction rule, thus there is no such reduction for it.

Note that the above notion of reduction concerns only deductions with a cut at the end of a *derivação*. We would of course like to extend it to reduction of cuts anywhere in a *derivação*, or reductions of subterms of proof terms which constitute redexes. Note that, however, the conclusion of the reduction does not change after reduction, thus we are free to continue applying rules to both sides of $\rightarrow$. The resulting pairs of trees constitutes an extended notion of reduction; it is analogous to compatibility in the untyped lambda calculus.

It's easy to see that the context Γ does not change during the reduction (both the original and the extended version), thus it's unnecessary to mention the context when we are discussing reductions. In what follows we will assume that every term is accompanied by a context which does no change during reduction. We then say “proof term” when we mean a proof term accompanied by a context which makes it well-typed.

As in lambda calculus, the notion of normal-form term and normal deduction is given:

Definição 62.5. A proof term with no redex is said to be in *normal form*; likewise, a *derivação* without cuts is a *normal derivação*. A proof term is in normal form if and only if its counterpart *derivação* is normal.

-NoValue-

content/intuitionistic-logic/propositions-as-types/normalization.tex

62.7 Normalization

In this section we prove that, via some reduction order, any deduction can be reduced to a normal deduction, which is called the *normalization property*. We will make use of the propositions-as-types correspondence: we show that every proof term can be reduced to a normal form; normalization for natural deduction *derivações* then follows. int:pty:nor:
sec

Firstly we define some functions that measure the complexity of terms. The *length* $\text{len}(\varphi)$ of a *fórmulas* is defined by

$$\begin{aligned}\text{len}(p) &= 0 \\ \text{len}(\varphi \wedge \psi) &= \text{len}(\varphi) + \text{len}(\psi) + 1 \\ \text{len}(\varphi \vee \psi) &= \text{len}(\varphi) + \text{len}(\psi) + 1 \\ \text{len}(\varphi \rightarrow \psi) &= \text{len}(\varphi) + \text{len}(\psi) + 1.\end{aligned}$$

The complexity of a redex M is measured by its *cut rank* $\text{cr}(M)$:

$$\begin{aligned}\text{cr}((\lambda x^\varphi. N^\psi)Q) &= \text{len}(\varphi) + \text{len}(\psi) + 1 \\ \text{cr}(\text{p}_i(\langle M^\varphi, N^\psi \rangle)) &= \text{len}(\varphi) + \text{len}(\psi) + 1 \\ \text{cr}(\text{case}(\text{in}_i^{\varphi_i}(M^{\varphi_i}), x_1^{\varphi_1}.N_1^X, x_2^{\varphi_2}.N_2^X)) &= \text{len}(\varphi) + \text{len}(\psi) + 1\end{aligned}$$

62.7. NORMALIZATION

The complexity of a proof term is measured by the most complex redex in it, and 0 if it is normal:

$$\text{mr}(M) = \max\{\text{cr}(N) \mid N \text{ is a sub term of } M \text{ and is redex}\}$$

int:pty:nor: **Lema 62.6.** *lem:subst* If $M[N^\varphi/x^\varphi]$ is a redex and $M \not\equiv x$, then one of the following cases holds:

1. M is itself a redex, or
2. M is of the form $\text{p}_i(x)$, and N is of the form $\langle P_1, P_2 \rangle$
3. M is of the form $\text{case}(i, x_1.P_1, x_2.P_2)$, and N is of the form $\text{in}_i(Q)$
4. M is of the form xQ , and N is of the form $\lambda x.P$

In the first case, $\text{cr}(M[N/x]) = \text{cr}(M)$; in the other cases, $\text{cr}(M[N/x]) = \text{len}(\varphi)$.

Prova. Proof by induction on M .

1. If M is a single variable y and $y \not\equiv x$, then $y[N/x]$ is y , hence not a redex.
2. If M is of the form $\langle N_1, N_2 \rangle$, or $\lambda x.N$, or $\text{in}_i(N)$, then $M[N^\varphi/x^\varphi]$ is also of that form, and so is not a redex.
3. If M is of the form $\text{p}_i(P)$, we consider two cases.

- a) If P is of the form $\langle P_1, P_2 \rangle$, then $M \equiv \text{p}_i(\langle P_1, P_2 \rangle)$ is a redex, and clearly

$$M[N/x] \equiv \text{p}_i(\langle P_1[N/x], P_2[N/x] \rangle)$$

is also a redex. The cut ranks are equal.

- b) If P is a single variable, it must be x to make the substitution a redex, and N must be of the form $\langle P_1, P_2 \rangle$. Now consider

$$M[N/x] \equiv \text{p}_i(x)[\langle P_1, P_2 \rangle/x],$$

which is $\text{p}_i(\langle P_1, P_2 \rangle)$. Its cut rank is equal to $\text{cr}(x)$, which is $\text{len}(\varphi)$.

The cases of $\text{case}(N, x_1.N_1, x_2.N_2)$ and PQ are similar. $\square$

Lema 62.7. If M contracts to M' , and $\text{cr}(M) > \text{cr}(N)$ for all proper redex sub-terms N of M , then $\text{cr}(M) > \text{mr}(M')$.

Prova. Proof by cases.

1. If M is of the form $\text{p}_i(\langle M_1, M_2 \rangle)$, then M' is M_i ; since any sub-term of M_i is also proper sub-term of M , the claim holds.

2. If M is of the form $(\lambda x^\varphi. N)Q^\varphi$, then M' is $N[Q^\varphi/x^\varphi]$. Consider a redex in M' . Either there is corresponding redex in N with equal cut rank, which is less than $\text{cr}(M)$ by assumption, or the cut rank equals $\text{len}(\varphi)$, which by definition is less than $\text{cr}((\lambda x^\varphi. N)Q)$.
3. If M is of the form

$$\text{case}(\text{in}_i(N^{\varphi_i}), x_1^{\varphi_1}.N_1^\chi, x_2^{\varphi_2}.N_2^\chi),$$

then $M' \equiv N_i[N/x_i^{\varphi_i}]$. Consider a redex in M' . Either there is corresponding redex in N_i with equal cut rank, which is less than $\text{cr}(M)$ by assumption; or the cut rank equals $\text{len}(\varphi_i)$, which by definition is less than $\text{cr}(\text{case}(\text{in}_i(N^{\varphi_i}), x_1^{\varphi_1}.N_1^\chi, x_2^{\varphi_2}.N_2^\chi))$. $\square$

Teorema 62.8. *All proof terms reduce to normal form; all *derivações* reduce to normal *derivações*.*

Prova. The second follows from the first. We prove the first by complete induction on $m = \text{mr}(M)$, where M is a proof term.

1. If $m = 0$, M is already normal.
2. Otherwise, we proceed by induction on n , the number of redexes in M with cut rank equal to m .
 - a) If $n = 1$, select any redex N such that $m = \text{cr}(N) > \text{cr}(P)$ for any proper sub-term P which is also a redex of course. Such a redex must exist, since any term only has finitely many subterms. Let N' denote the reductum of N . Now by the lemma $\text{mr}(N') < \text{mr}(N)$, thus we can see that n , the number of redexes with $\text{cr}(=)m$ is decreased. So m is decreased (by 1 or more), and we can apply the inductive hypothesis for m .
 - b) For the induction step, assume $n > 1$. the process is similar, except that n is only decreased to a positive number and thus m does not change. We simply apply the induction hypothesis for n . $\square$

The normalization of terms is actually not specific to the reduction order we chose. In fact, one can prove that regardless of the order in which redexes are reduced, the term always reduces to a normal form. This property is called *strong normalization*.

Parte XIV

Counterfactuals

Capítulo 63

Introduction

-NoValue-

`content/counterfactuals/introduction/material-conditional.tex`

63.1 The Material Conditional

cnt:int:mat:
sec In its simplest form in English, a conditional is a sentence of the form “If ... then ...,” where the ... are themselves sentences, such as “If the butler did it, then the gardener is innocent.” In introductory logic courses, we learn to symbolize conditionals using the $\rightarrow$ connective: symbolize the parts indicated by ..., p.ex., by *fórmulas* φ and ψ , and the entire conditional is symbolized by $\varphi \rightarrow \psi$.

The connective $\rightarrow$ is *truth-functional*, O.s., the truth value— $\mathbb{T}$ or $\mathbb{F}$ —of $\varphi \rightarrow \psi$ is determined by the truth values of φ and ψ : $\varphi \rightarrow \psi$ is true iff φ is false or ψ is true, and false otherwise. Relative to a truth value assignment $\mathbf{v}$, we define $\mathbf{v} \models \varphi \rightarrow \psi$ iff $\mathbf{v} \not\models \varphi$ or $\mathbf{v} \models \psi$. The connective $\rightarrow$ with this semantics is called the *material conditional*.

This definition results in a number of elementary logical facts. First of all, the deduction theorem holds for the material conditional:

$$\text{If } \Gamma, \varphi \models \psi \text{ then } \Gamma \models \varphi \rightarrow \psi \quad (63.1)$$

It is truth-functional: $\varphi \rightarrow \psi$ and $\neg\varphi \vee \psi$ are equivalent:

$$\varphi \rightarrow \psi \models \neg\varphi \vee \psi \quad (63.2)$$

$$\neg\varphi \vee \psi \models \varphi \rightarrow \psi \quad (63.3)$$

CAPÍTULO 63. INTRODUCTION

A material conditional is entailed by its consequent and by the negation of its antecedent:

$$\psi \models \varphi \rightarrow \psi \quad (63.4)$$

$$\neg\varphi \models \varphi \rightarrow \psi \quad (63.5)$$

A false material conditional is equivalent to the conjunction of its antecedent and the negation of its consequent: if $\varphi \rightarrow \psi$ is false, $\varphi \wedge \neg\psi$ is true, and vice versa:

$$\neg(\varphi \rightarrow \psi) \models \varphi \wedge \neg\psi \quad (63.6)$$

$$\varphi \wedge \neg\psi \models \neg(\varphi \rightarrow \psi) \quad (63.7)$$

The material conditional supports modus ponens:

$$\varphi, \varphi \rightarrow \psi \models \psi \quad (63.8)$$

The material conditional agglomerates:

$$\varphi \rightarrow \psi, \varphi \rightarrow \chi \models \varphi \rightarrow (\psi \wedge \chi) \quad (63.9)$$

We can always strengthen the antecedent, O.s., the conditional is *monotonic*:

$$\varphi \rightarrow \psi \models (\varphi \wedge \chi) \rightarrow \psi \quad (63.10)$$

The material conditional is transitive, O.s., the chain rule is valid:

$$\varphi \rightarrow \psi, \psi \rightarrow \chi \models \varphi \rightarrow \chi \quad (63.11)$$

The material conditional is equivalent to its contrapositive:

$$\varphi \rightarrow \psi \models \neg\psi \rightarrow \neg\varphi \quad (63.12)$$

$$\neg\psi \rightarrow \neg\varphi \models \varphi \rightarrow \psi \quad (63.13)$$

These are all useful and unproblematic inferences in mathematical reasoning. However, the philosophical and linguistic literature is replete with purported counterexamples to the equivalent inferences in non-mathematical contexts. These suggest that the material conditional $\rightarrow$ is not—or at least not always—the appropriate connective to use when symbolizing English “if ... then ...” statements.

-NoValue-

content/counterfactuals/introduction/paradoxes-material.tex

63.2 Paradoxes of the Material Conditional

One of the first to criticize the use of $\varphi \rightarrow \psi$ as a way to symbolize “if ... then ...” statements of English was C. I. Lewis. Lewis was criticizing the use of the material condition in Whitehead and Russell’s *Principia Mathematica*, who pronounced $\rightarrow$ as “implies.” Lewis rightly complained that if $\rightarrow$ meant “implies,” then any false proposition p implies that p implies q , since $p \rightarrow (p \rightarrow q)$ is true if p is false, and that any true proposition q implies that p implies q , since $q \rightarrow (p \rightarrow q)$ is true if q is true.

Logicians of course know that *implication*, O.s., logical entailment, is not a connective but a relation between *fórmulas* or statements. So we should just not read $\rightarrow$ as “implies” to avoid confusion.¹ As long as we don’t, the particular worry that Lewis had simply does not arise: p does not “imply” q even if we think of p as standing for a false English sentence. To determine if $p \models q$ we must consider *all* *valorações*, and $p \not\models q$ even when we use p to symbolize a sentence which happens to be false.

But there is still something odd about “if ... then ...” statements such as Lewis’s

If the moon is made of green cheese, then $2 + 2 = 4$.

and about the inferences

The moon is not made of green cheese. Therefore, if the moon is made of green cheese, then $2 + 2 = 4$.

$2 + 2 = 4$. Therefore, if the moon is made of green cheese, then $2 + 2 = 4$.

Yet, if “if ... then ...” were just $\rightarrow$, the sentence would be unproblematically true, and the inferences unproblematically valid.

Another example of concerns the tautology $(\varphi \rightarrow \psi) \vee (\psi \rightarrow \varphi)$. This would suggest that if you take two indicative sentences S and T from the newspaper at random, the sentence “If S then T , or if T then S ” should be true.

-NoValue-

content/counterfactuals/introduction/strict-conditional.tex

63.3 The Strict Conditional

Lewis introduced the *strict conditional* $\rightarrow$ and argued that it, not the material conditional, corresponds to implication. In alethic modal logic, $\varphi \rightarrow \psi$ can be defined as $\Box(\varphi \rightarrow \psi)$. A strict conditional is thus true (at a world) iff the corresponding material conditional is necessary.

¹Reading “ $\rightarrow$ ” as “implies” is still widely practised by mathematicians and computer scientists, although philosophers try to avoid the confusions Lewis highlighted by pronouncing it as “only if.”

How does the strict conditional fare vis-a-vis the paradoxes of the material conditional? A strict conditional with a false antecedent and one with a true consequent, may be true, or it may be false. Moreover, $(\varphi \rightarrow \psi) \vee (\psi \rightarrow \varphi)$ is not valid. The strict conditional $\varphi \rightarrow \psi$ is also not equivalent to $\neg\varphi \vee \psi$, so it is not truth functional.

We have:

$$\varphi \rightarrow \psi \models \neg\varphi \vee \psi \text{ but:} \quad (63.14)$$

$$\neg\varphi \vee \psi \not\models \varphi \rightarrow \psi \quad (63.15)$$

$$\psi \not\models \varphi \rightarrow \psi \quad (63.16)$$

$$\neg\varphi \not\models \varphi \rightarrow \psi \quad (63.17)$$

$$\neg(\varphi \rightarrow \psi) \not\models \varphi \wedge \neg\psi \text{ but:} \quad (63.18)$$

$$\varphi \wedge \neg\psi \models \neg(\varphi \rightarrow \psi) \quad (63.19)$$

However, the strict conditional still supports modus ponens:

$$\varphi, \varphi \rightarrow \psi \models \psi \quad (63.20)$$

The strict conditional agglomerates:

$$\varphi \rightarrow \psi, \varphi \rightarrow \chi \models \varphi \rightarrow (\psi \wedge \chi) \quad (63.21)$$

Antecedent strengthening holds for the strict conditional:

$$\varphi \rightarrow \psi \models (\varphi \wedge \chi) \rightarrow \psi \quad (63.22)$$

The strict conditional is also transitive:

$$\varphi \rightarrow \psi, \psi \rightarrow \chi \models \varphi \rightarrow \chi \quad (63.23)$$

Finally, the strict conditional is equivalent to its contrapositive:

$$\varphi \rightarrow \psi \models \neg\psi \rightarrow \neg\varphi \quad (63.24)$$

$$\neg\psi \rightarrow \neg\varphi \models \varphi \rightarrow \psi \quad (63.25)$$

Problema 63.1. Give **S5**-counterexamples to the entailment relations which do not hold for the strict conditional, O.s., for:

1. $\neg p \not\models \Box(p \rightarrow q)$
2. $q \not\models \Box(p \rightarrow q)$
3. $\neg\Box(p \rightarrow q) \not\models p \wedge \neg q$
4. $\not\models \Box(p \rightarrow q) \vee \Box(q \rightarrow p)$

63.3. THE STRICT CONDITIONAL

Problem 63.2. Show that the valid entailment relations hold for the strict conditional by giving **S5**-proofs of:

1. $\Box(\varphi \rightarrow \psi) \models \neg\varphi \vee \psi$
2. $\varphi \wedge \neg\psi \models \neg\Box(\varphi \rightarrow \psi)$
3. $\varphi, \Box(\varphi \rightarrow \psi) \models \psi$
4. $\Box(\varphi \rightarrow \psi), \Box(\varphi \rightarrow \chi) \models \Box(\varphi \rightarrow (\psi \wedge \chi))$
5. $\Box(\varphi \rightarrow \psi) \models \Box((\varphi \wedge \chi) \rightarrow \psi)$
6. $\Box(\varphi \rightarrow \psi), \Box(\psi \rightarrow \chi) \models \Box(\varphi \rightarrow \chi)$
7. $\Box(\varphi \rightarrow \psi) \models \Box(\neg\psi \rightarrow \neg\varphi)$
8. $\Box(\neg\psi \rightarrow \neg\varphi) \models \Box(\varphi \rightarrow \psi)$

However, the strict conditional still has its own “paradoxes.” Just as a material conditional with a false antecedent or a true consequent is true, a strict conditional with a *necessarily* false antecedent or a necessarily true consequent is true. Moreover, any true strict conditional is necessarily true, and any false strict conditional is necessarily false. In other words, we have

$$\Box\neg\varphi \models \varphi \rightarrow \psi \quad (63.26)$$

$$\Box\psi \models \varphi \rightarrow \psi \quad (63.27)$$

$$\varphi \rightarrow \psi \models \Box(\varphi \rightarrow \psi) \quad (63.28)$$

$$\neg(\varphi \rightarrow \psi) \models \Box\neg(\varphi \rightarrow \psi) \quad (63.29)$$

These are not problems if you think of $\rightarrow$ as “implies.” Logical entailment relationships are, after all, mathematical facts and so can’t be contingent. But they do raise issues if you want to use $\rightarrow$ as a logical connective that is supposed to capture “if ... then ...,” especially the last two. For surely there are “if ... then ...” statements that are contingently true or contingently false—in fact, they generally are neither necessary nor impossible.

Problem 63.3. Give proofs in **S5** of:

1. $\Box\neg\varphi \models \varphi \rightarrow \psi$
2. $\varphi \rightarrow \psi \models \Box(\varphi \rightarrow \psi)$
3. $\neg(\varphi \rightarrow \psi) \models \Box\neg(\varphi \rightarrow \psi)$

Use the definition of $\rightarrow$ to do so.

-NoValue-

content/counterfactuals/introduction/counterfactuals.tex

63.4 Counterfactuals

A very common and important form of “if ... then ...” constructions in English are built using the past subjunctive form of *to be*: “if it were the case that ... then it would be the case that ...” Because usually the antecedent of such a conditional is false, O.s., counter to fact, they are called *counterfactual conditionals* (and because they use the subjunctive form of *to be*, also *subjunctive conditionals*. They are distinguished from *indicative conditionals* which take the form of “if it is the case that ... then it is the case that ...” Counterfactual and indicative conditionals differ in truth conditions. Consider Adams’s famous example:

cnt:int:cnt:
sec

If Oswald didn’t kill Kennedy, someone else did.

If Oswald hadn’t killed Kennedy, someone else would have.

The first is indicative, the second counterfactual. The first is clearly true: we know President John F. Kennedy was killed by *someone*, and if that someone wasn’t (contrary to the Warren Report) Lee Harvey Oswald, then someone else killed Kennedy. The second one says something different. It claims that if Oswald hadn’t killed Kennedy, O.s., if the Dallas shooting had been avoided or had been unsuccessful, history would have subsequently unfolded in such a way that another assassination would have been successful. In order for it to be true, it would have to be the case that powerful forces had conspired to ensure JFK’s death (as many JFK conspiracy theorists believe).

It is a live debate whether the *indicative* conditional is correctly captured by the material conditional, in particular, whether the paradoxes of the material conditional can be “explained” in a way that is compatible with it giving the truth conditions for English indicative conditionals. By contrast, it is uncontroversial that counterfactual conditionals cannot be symbolized correctly by the material conditionals. That is clear because, even though generally the antecedents of counterfactuals are false, not all counterfactuals with false antecedents are true—for instance, if you believe the Warren Report, and there was no conspiracy to assassinate JFK, then Adams’s counterfactual conditional is an example.

Counterfactual conditionals play an important role in causal reasoning: a prime example of the use of counterfactuals is to express causal relationships. P.ex., striking a match causes it to light, and you can express this by saying “if this match were struck, it would light.” Material, and generally indicative conditionals, cannot be used to express this: “the match is struck $\rightarrow$ the match lights” is true if the match is never struck, regardless of what would happen if it were. Even worse, “the match is struck $\rightarrow$ the match turns into a bouquet of flowers” is also true if it is never struck, but the match would certainly not turn into a bouquet of flowers if it were struck.

It is still debated What exactly the correct logic of counterfactuals is. An influential analysis of counterfactuals was given by Stalnaker and Lewis. According to them, a counterfactual “if it were the case that *S* then it would be

the case that T is true iff T is true in the counterfactual situation (“possible world”) that is closest to the way the actual world is and where S is true. This is called an “ontic” analysis, since it makes reference to an ontology of possible worlds. Other analyses make use of conditional probabilities or theories of belief revision. There is a proliferation of different proposed logics of counterfactuals. There isn’t even a single Lewis–Stalnaker logic of counterfactuals: even though Stalnaker and Lewis proposed accounts along similar lines with reference to closest possible worlds, the assumptions they made result in different valid inferences.

Capítulo 64

Minimal Change Semantics

-NoValue-

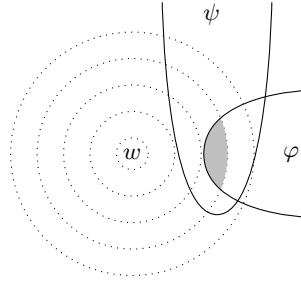
`content/counterfactuals/minimal-change-semantics/introduction.tex`

64.1 Introduction

cnt:min:int:
sec Stalnaker and Lewis proposed accounts of counterfactual conditionals such as “If the match were struck, it would light.” Their accounts were proposals for how to properly understand the truth conditions for such sentences. The idea behind both proposals is this: to evaluate whether a counterfactual conditional is true, we have to consider those possible worlds which are minimally different from the way the world actually is to make the antecedent true. If the consequent is true in these possible worlds, then the counterfactual is true. For instance, suppose I hold a match and a matchbook in my hand. In the actual world I only look at them and ponder what would happen if I were to strike the match. The minimal change from the actual world where I strike the match is that where I decide to act and strike the match. It is minimal in that nothing else changes: I don’t also jump in the air, striking the match doesn’t also light my hair on fire, I don’t suddenly lose all strength in my fingers, I am not simultaneously doused with water in a SuperSoaker ambush, etc. In that alternative possibility, the match lights. Hence, it’s true that if I were to strike the match, it would light.

This intuitive account can be paired with formal semantics for logics of counterfactuals. Lewis introduced the symbol “ $\Box\rightarrow$ ” for the counterfactual while Stalnaker used the symbol “ $>$ ”. We’ll use $\Box\rightarrow$, and add it as a binary connective to propositional logic. So, we have, in addition to *fórmulas* of the form $\varphi \rightarrow \psi$ also *fórmulas* of the form $\varphi \Box\rightarrow \psi$. The formal semantics, like the relational semantics for modal logic, is based on models in which *fórmulas* are evaluated at worlds, and the satisfaction condition defining $\mathfrak{M}, w \Vdash \varphi \Box\rightarrow \psi$ is given in terms of $\mathfrak{M}, w' \Vdash \varphi$ and $\mathfrak{M}, w' \Vdash \psi$ for some (other) worlds w' . Which w' ? Intuitively, the one(s) closest to w for which it holds that $\mathfrak{M}, w' \Vdash \varphi$. This requires that a relation of “closeness” has to be included in the model as well.

Lewis introduced an instructive way of representing counterfactual situations graphically. Each possible world is at the center of a set of nested spheres containing other worlds—we draw these spheres as concentric circles. The worlds between two spheres are equally close to the world at the center as each other, those contained in a nested sphere are closer, and those in a surrounding sphere further away.



The closest φ -worlds are those worlds w' where φ is satisfied which lie in the smallest sphere around the center world w (the gray area). Intuitively, $\varphi \Box\rightarrow \psi$ is satisfied at w if ψ is true at all closest φ -worlds.

-NoValue-

[content/counterfactuals/minimal-change-semantics/sphere-models.tex](#)

64.2 Sphere Models

One way of providing a formal semantics for counterfactuals is to turn Lewis’s informal account into a mathematical structure. The spheres around a world w then are sets of worlds. Since the spheres are nested, the sets of worlds around w have to be linearly ordered by the subset relation.

con:min:sph:
sec

Definição 64.1. A *sphere model* is a triple $\mathfrak{M} = \langle W, O, V \rangle$ where W is a non-empty set of worlds, $V: \text{At}_0 \rightarrow \wp(W)$ is a valuation, and $O: W \rightarrow \wp(\wp(W))$ assigns to each world w a *system of spheres* O_w . For each w , O_w is a set of sets of worlds, and must satisfy:

1. O_w is *centered* on w : $\{w\} \in O_w$.

64.2. SPHERE MODELS

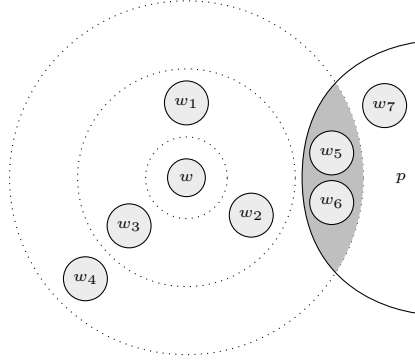


Figura 64.1: Diagram of a sphere model

con:min:sph:
fig:sphere-model

2. O_w is *nested*: whenever $S_1, S_2 \in O_w$, $S_1 \subseteq S_2$ or $S_2 \subseteq S_1$, O.s., O_w is linearly ordered by $\subseteq$.
3. O_w is closed under non-empty unions.
4. O_w is closed under non-empty intersections.

The intuition behind O_w is that the worlds “around” w are stratified according to how far away they are from w . The innermost sphere is just w by itself, O.s., the set $\{w\}$: w is closer to w than the worlds in any other sphere. If $S \subsetneq S'$, then the worlds in $S' \setminus S$ are further way from w than the worlds in S : $S' \setminus S$ is the “layer” between the S and the worlds outside of S' . In particular, we have to think of the spheres as containing all the worlds within their outer surface; they are not just the individual layers.

The diagram in **Figura 64.1** corresponds to the sphere model with $W = \{w, w_1, \dots, w_7\}$, $V(p) = \{w_5, w_6, w_7\}$. The innermost sphere $S_1 = \{w\}$. The closest worlds to w are w_1, w_2, w_3 , so the next larger sphere is $S_2 = \{w, w_1, w_2, w_3\}$. The worlds further out are w_4, w_5, w_6 , so the outermost sphere is $S_3 = \{w, w_1, \dots, w_6\}$. The system of spheres around w is $O_w = \{S_1, S_2, S_3\}$. The world w_7 is not in any sphere around w . The closest worlds in which p is true are w_5 and w_6 , and so the smallest p -admitting sphere is S_3 .

To define satisfaction of a formula φ at world w in a sphere model $\mathfrak{M}$, $\mathfrak{M}, w \models \varphi$, we expand the definition for modal **formulas** to include a clause for $\psi \Box \rightarrow \chi$:

Definição 64.2. $\mathfrak{M}, w \models \psi \Box \rightarrow \chi$ iff either

1. For all $u \in \bigcup O_w$, $\mathfrak{M}, u \not\models \psi$, or
2. For some $S \in O_w$,
 - a) $\mathfrak{M}, u \models \psi$ for some $u \in S$, and

con:min:sph:
sphere-vac
con:min:sph:
sphere-nonvac

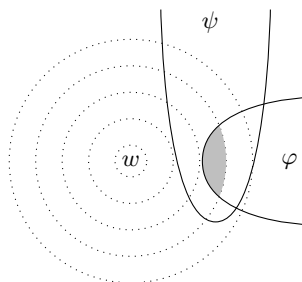


Figura 64.2: Non-vacuously true counterfactual

con:min:tf:
fig:true

b) for all $v \in S$, either $\mathfrak{M}, v \not\models \psi$ or $\mathfrak{M}, v \models \chi$.

According to this definition, $\mathfrak{M}, w \models \psi \Box \rightarrow \chi$ iff either the antecedent ψ is false everywhere in the spheres around w , or there is a sphere S where ψ is true, and the material conditional $\psi \rightarrow \chi$ is true at all worlds in that “ ψ -admitting” sphere. Note that we didn’t require in the definition that S is the *innermost* ψ -admitting sphere, contrary to what one might expect from the intuitive explanation. But if the condition in (2) is satisfied for some sphere S , then it is also satisfied for all spheres S contains, and hence in particular for the innermost sphere.

Note also that the definition of sphere models does not require that there *is* an innermost ψ -admitting sphere: we may have an infinite sequence $S_1 \supsetneq S_2 \supsetneq \dots \supsetneq \{w\}$ of ψ -admitting spheres, and hence no innermost ψ -admitting spheres. In that case, $\mathfrak{M}, w \models \psi \Box \rightarrow \chi$ iff $\psi \rightarrow \chi$ holds throughout the spheres $S_i, S_{i+1}, \dots$, for some i .

-NoValue-

content/counterfactuals/minimal-change-semantics/true-false.tex

64.3 Truth and Falsity of Counterfactuals

A counterfactual $\varphi \Box \rightarrow \psi$ is (non-vacuously) true if the closest φ -worlds are all ψ -worlds, as depicted in [Figura 64.2](#). A counterfactual is also true at w if the system of spheres around w has no φ -admitting spheres at all. In that case it is *vacuously* true (see [Figura 64.3](#)).

con:min:tf:
sec

It can be false in two ways. One way is if the closest φ -worlds are not all ψ -worlds, but some of them are. In this case, $\varphi \Box \rightarrow \neg\psi$ is also false (see [Figura 64.4](#)). If the closest φ -worlds do not overlap with the ψ -worlds at all, then $\varphi \Box \rightarrow \psi$ is false. But, in this case all the closest φ -worlds are $\neg\psi$ -worlds, and so $\varphi \Box \rightarrow \neg\psi$ is true (see [Figura 64.5](#)).

64.3. TRUTH AND FALSITY OF COUNTERFACTUALS

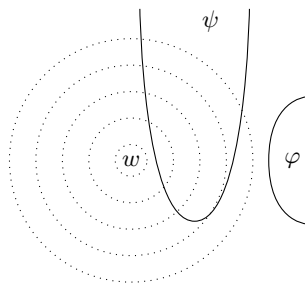


Figura 64.3: Vacuously true counterfactual

con:min:tf:
fig:vacuous

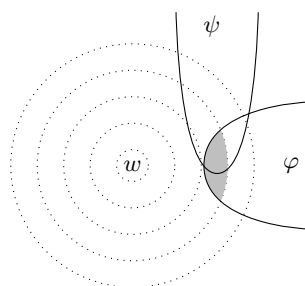


Figura 64.4: False counterfactual, false opposite

con:min:tf:
fig:false

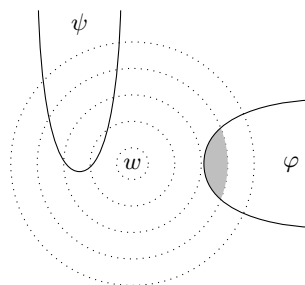


Figura 64.5: False counterfactual, true opposite

con:min:tf:
fig:false-opposite

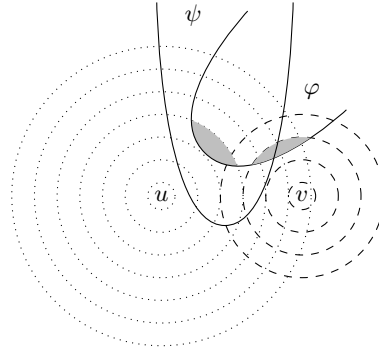


Figura 64.6: Contingent counterfactual

In contrast to the strict conditional, counterfactuals may be contingent. Consider the sphere model in [Figura 64.6](#). The φ -worlds closest to u are all ψ -worlds, so $\mathfrak{M}, u \Vdash \varphi \Box \rightarrow \psi$. But there are φ -worlds closest to v which are not ψ -worlds, so $\mathfrak{M}, v \nVdash \varphi \Box \rightarrow \psi$.

con:min:tf:
fig:contingent

content/counterfactuals/minimal-change-semantics/antecedent-strengthening.tex

64.4 Antecedent Strengthening

“Strengthening the antecedent” refers to the inference $\varphi \rightarrow \chi \models (\varphi \wedge \psi) \rightarrow \chi$. It is valid for the material conditional, but invalid for counterfactuals. Suppose it is true that if I were to strike this match, it would light. (That means, there is nothing wrong with the match or the matchbook surface, I will not break the match, etc.) But it is not true that if I were to light this match in outer space, it would light. So the following inference is invalid:

cnt:min:agg:
sec

If the match were struck, it would light.

Therefore, if the match were struck in outer space, it would light.

The Lewis–Stalnaker account of conditionals explains this: the closest world where I light the match and I do so in outer space is much further removed from the actual world than the closest world where I light the match is. So although it’s true that the match lights in the latter, it is not in the former. And that is as it should be.

Exemplo 64.3. The sphere semantics invalidates the inference, O.s., we have $p \Box \rightarrow r \nVdash (p \wedge q) \Box \rightarrow r$. Consider the model $\mathfrak{M} = \langle W, O, V \rangle$ where $W = \{w, w_1, w_2\}$, $O_w = \{\{w\}, \{w, w_1\}, \{w, w_1, w_2\}\}$, $V(p) = \{w_1, w_2\}$, $V(q) = \{w_2\}$, and $V(r) = \{w_1\}$. There is a p -admitting sphere $S = \{w, w_1\}$ and $p \rightarrow r$ is true at all worlds in it, so $\mathfrak{M}, w \Vdash p \Box \rightarrow r$. There is also a $(p \wedge q)$ -admitting

64.5. TRANSITIVITY

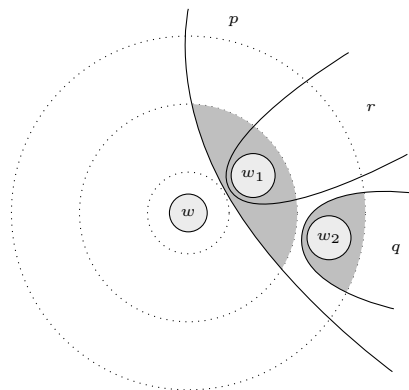


Figura 64.7: Counterexample to antecedent strengthening

sphere $S' = \{w, w_1, w_2\}$ but $\mathfrak{M}, w_2 \not\models (p \wedge q) \rightarrow r$, so $\mathfrak{M}, w \not\models (p \wedge q) \Box \rightarrow r$ (see [Figura 64.7](#)).

-NoValue-

[content/counterfactuals/minimal-change-semantics/transitivity.tex](#)

64.5 Transitivity

For the material conditional, the chain rule holds: $\varphi \rightarrow \psi, \psi \rightarrow \chi \models \varphi \rightarrow \chi$. In other words, the material conditional is transitive. Is the same true for counterfactuals? Consider the following example due to Stalnaker.

If J. Edgar Hoover had been born a Russian, he would have been a Communist.

If J. Edgar Hoover were a Communist, he would have been be a traitor.

Therefore, If J. Edgar Hoover had been born a Russian, he would have been be a traitor.

If Hoover had been born (at the same time he actually did), not in the United States, but in Russia, he would have grown up in the Soviet Union and become a Communist (let's assume). So the first premise is true. Likewise, the second premise, considered in isolation is true. The conclusion, however, is false: in all likelihood, Hoover would have been a fervent Communist if he had been born in the USSR, and not been a traitor (to his country). The intuitive assignment of truth values is borne out by the Stalnaker–Lewis account. The closest possible world to ours with the only change being Hoover's place of birth is the one

where Hoover grows up to be a good citizen of the USSR. This is the closest possible world where the antecedent of the first premise and of the conclusion is true, and in that world Hoover is a loyal member of the Communist party, and so not a traitor. To evaluate the second premise, we have to look at a different world, however: the closest world where Hoover is a Communist, which is one where he was born in the United States, turned, and thus became a traitor.¹

Problema 64.1. Find a convincing, intuitive example for the failure of transitivity of counterfactuals.

Exemplo 64.4. The sphere semantics invalidates the inference, O.s., we have $p \Box \rightarrow q, q \Box \rightarrow r \not\models p \Box \rightarrow r$. Consider the model $\mathfrak{M} = \langle W, O, V \rangle$ where $W = \{w, w_1, w_2\}$, $O_w = \{\{w\}, \{w, w_1\}, \{w, w_1, w_2\}\}$, $V(p) = \{w_2\}$, $V(q) = \{w_1, w_2\}$, and $V(r) = \{w_1\}$. There is a p -admitting sphere $S = \{w, w_1, w_2\}$ and $p \rightarrow q$ is true at all worlds in it, so $\mathfrak{M}, w \Vdash p \Box \rightarrow q$. There is also a q -admitting sphere $S' = \{w, w_1\}$ and $\mathfrak{M} \not\models q \rightarrow r$ is true at all worlds in it, so $\mathfrak{M}, w \Vdash q \Box \rightarrow r$. However, the p -admitting sphere $\{w, w_1, w_2\}$ contains a world, namely w_2 , where $\mathfrak{M}, w_2 \not\models p \rightarrow r$.

cnt:min:tra:
ex:trans-counterex

Problema 64.2. Draw the sphere diagram corresponding to the counterexample in [Exemplo 64.4](#).

Problema 64.3. In [Exemplo 64.4](#), world w_2 is where Hoover is born in Russia, is a communist, and not a traitor, and w_1 is the world where Hoover is born in the US, is a communist, and a traitor. In this model, w_1 is closer to w than w_2 is. Is this necessary? Can you give a counterexample that does not assume that Hoover's being born in Russia is a more remote possibility than him being a Communist?

-NoValue-

[content/counterfactuals/minimal-change-semantics/contraposition.tex](#)

64.6 Contraposition

Material and strict conditionals are equivalent to their contrapositives. Counterfactuals are not. Here is an example due to Kratzer:

cnt:min:cpo:
sec

If Goethe hadn't died in 1832, he would (still) be dead now.

If Goethe weren't dead now, he would have died in 1832.

The first sentence is true: humans don't live hundreds of years. The second is clearly false: if Goethe weren't dead now, he would be still alive, and so couldn't have died in 1832.

¹Of course, to appreciate the force of the example we have to take on board some metaphysical and political assumptions, p.ex., that it is possible that Hoover could have been born to Russian parents, or that Communists in the US of the 1950s were traitors to their country.

64.6. CONTRAPOSITION

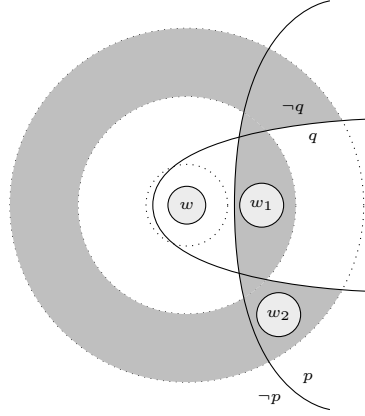


Figura 64.8: Counterexample to contraposition

cnt:min:cpo:
fig:contraposition

cnt:min:cpo:
ex:contraposition-counterex

Exemplo 64.5. The sphere semantics invalidates contraposition, O.s., we have $p \Box \rightarrow q \not\models \neg q \Box \rightarrow \neg p$. Think of p as “Goethe didn’t die in 1832” and q as “Goethe is dead now.” We can capture this in a model $\mathfrak{M}_1 = \langle W, O, V \rangle$ with $W = \{w, w_1, w_2\}$, $O = \{\{w\}, \{w, w_1\}, \{w, w_1, w_2\}\}$, $V(p) = \{w_1, w_2\}$ and $V(q) = \{w, w_1\}$. So w is the actual world where Goethe died in 1832 and is still dead; w_1 is the (close) world where Goethe died in, say, 1833, and is still dead; and w_2 is a (remote) world where Goethe is still alive. There is a p -admitting sphere $S = \{w, w_1\}$ and $p \rightarrow q$ is true at all worlds in it, so $\mathfrak{M}, w \Vdash p \Box \rightarrow q$. However, the $\neg q$ -admitting sphere $\{w, w_1, w_2\}$ contains a world, namely w_2 , where q is false and p is true, so $\mathfrak{M}, w_2 \not\models \neg q \rightarrow \neg p$.

Parte XV

Set Theory

Capítulo 65

A Concepção Iterativa

`content/set-theory/story/extensionality.tex`

65.1 Extensionalidade

O primeiro ponto a notar é que os conjuntos são individuados pelos seus **mem-** sth:story:extensionality:sec
bros. Mais precisamente:

Extensionalidade (Extensionalidade). Se os conjuntos A e B têm os mesmos **membros**, então A e B são o mesmo conjunto.

$$\forall A \forall B (\forall x (x \in A \leftrightarrow x \in B) \rightarrow A = B)$$

Assumimos isto ao longo de **Parte I**. Mas vale a pena repetir. O axioma de extensionalidade exprime a ideia fundamental de que um conjunto é determinado pelos seus **membros**. (Assim, os conjuntos podem contrastar-se com os *conceitos*, nos quais precisamente os mesmos objetos podem cair sob muitos conceitos diferentes.)

Por que aceitar este princípio? Ora, é plausível dizer que qualquer recusa da extensionalidade é uma decisão de abandonar tudo o que possa ser chamado *teoria dos conjuntos*. A teoria dos conjuntos não é mais nem menos do que a teoria de coleções extensionais.

O verdadeiro desafio na **Parte XV**, contudo, é estabelecer princípios que nos digam *que conjuntos existem*. E acaba por se verificar que a única resposta verdadeiramente “óbvia” a esta questão é provadamente errada.

`content/set-theory/story/russells-paradox-again.tex`

65.2 Paradoxo de Russell (de novo)

sth:story:rus:
sec

Na **Parte I**, trabalhamos com uma teoria ingênua dos conjuntos. Mas, segundo uma concepção *muito* ingênua, os conjuntos são apenas as extensões dos predicados. Este pensamento ingênuo imporia o seguinte princípio:

Compreensão ingênua. $\{x : \varphi(x)\}$ existe para qualquer fórmula φ .

Por tentador que seja este princípio, é provadamente inconsistente. Vimo-lo na **Seção 1.6**, mas o resultado é tão importante, e tão direto, que vale a pena repetir. *Ipsis verbis*.

Teorema 65.1 (Paradoxo de Russell). *Não existe conjunto $R = \{x : x \notin x\}$*

Prova. Se $R = \{x : x \notin x\}$ existisse, então $R \in R$ sse $R \notin R$, o que é uma contradição. $\square$

Russell descobriu este resultado em junho de 1901. (Não colocou, porém, o paradoxo exatamente na forma em que o acabamos de apresentar, pois estava a considerar a teoria dos conjuntos de Frege, tal como delineada em *Grundgesetze*. Voltaremos a isto na **Seção 65.6**.) Russell escreveu a Frege a 16 de junho de 1902, explicando a inconsistência no sistema de Frege. Para a correspondência, e um pouco de contexto, ver **Heijenoort (1967)**, pp. 124–8).

Vale a pena enfatizar que esta prova de duas linhas é um resultado de *lógica pura*. De fato, usamos implicitamente um axioma (não lógico?) de extensionalidade na nossa notação $\{x : x \notin x\}$; pois $\{x : \varphi(x)\}$ deve ser o *único* (pela extensionalidade) conjunto dos φ s, se algum existir. Mas podemos evitar até o vestígio da extensionalidade, bastando enunciar o resultado assim: *não existe conjunto cujos membros sejam exatamente os conjuntos que não são membros de si próprios*. E isto pouco tem a ver com conjuntos. Como o próprio Russell observou, raciocínio rigorosamente análogo leva-nos a concluir: *nenhum homem barbeia exatamente os homens que não se barbeiam a si próprios*. Ou: *nenhum pug cheira exatamente os pugs que não se cheiram a si próprios*. E assim por diante. Esquemáticamente, a forma do resultado é apenas:

$$\neg \exists x \forall z (Rzx \leftrightarrow \neg Rzz).$$

E isto não passa de um teorema (esquema) da lógica de primeira ordem. Consequentemente, não podemos evitar o paradoxo de Russell apenas mexendo na nossa teoria dos conjuntos; ele surge antes sequer de *chegarmos* à teoria dos conjuntos. Se formos usar lógica (clássica) de primeira ordem, temos simplesmente de *aceitar* que não existe conjunto $R = \{x : x \notin x\}$.

A consequência é esta. Se quisermos aceitar a compreensão ingênua enquanto *evitamos* a inconsistência, não basta mexer na *teoria dos conjuntos*. Teríamos, em vez disso, de reformular a nossa *lógica*.

É claro que já foram apresentadas teorias dos conjuntos com lógicas não clássicas. Mas são—no mínimo—não standard. A abordagem usual ao paradoxo de Russell é tratá-lo como uma prova direta de não existência e depois tentar aprender a viver com ela. É essa a abordagem que seguiremos.

`content/set-theory/story/predicativity.tex`

65.3 Predicativo e Impredicativo

O conjunto de Russell, R , foi definido via $\{x : x \notin x\}$. Dito com mais detalhe, R seria o conjunto que contém todos e apenas aqueles conjuntos que não são membros de si próprios. Assim, ao definir R , quantificamos sobre o domínio que conteria R (se este existisse).

sth:story:predicative:
sec

Isto é uma definição *impredicativa*. Mais em geral, podemos dizer que uma definição é impredicativa sse quantifica sobre um domínio que contém o objeto que está a ser definido.

Na sequência dos paradoxos, Whitehead, Russell, Poincaré e Weyl rejeitaram tais definições impredicativas como “viciosamente circulares”:

Uma análise dos paradoxos a evitar mostra que todos eles resultam de uma espécie de círculo vicioso. Os círculos viciosos em questão surgem ao supor que uma coleção de objetos pode conter membros que só podem ser definidos por meio da coleção como um todo[... ¶]

O princípio que nos permite evitar totalidades ilegítimas pode enunciar-se do seguinte modo: “Tudo o que envolve *todos* os membros de uma coleção não pode ser um dos membros da coleção”; ou, inversamente: “Se, supondo que uma certa coleção tivesse um todo, ela teria membros só definíveis em termos desse todo, então a dita coleção não tem todo.” Chamaremos a isto o “princípio do círculo vicioso”, porque nos permite evitar os círculos viciosos envolvidos na suposição de totalidades ilegítimas. (Whitehead and Russell, 1910, p. 37)

Se os acompanharmos na rejeição das definições impredicativas motivada pelo *princípio do círculo vicioso*, então podemos tentar substituir o desastroso esquema de compreensão ingênua (de [Seção 65.2](#)) por algo deste gênero:

Compreensão predicativa. Para toda a fórmula φ que quantifica apenas sobre conjuntos: existe o conjunto' $\{x : \varphi(x)\}$.

Enquanto os conjuntos' não forem conjuntos, não se seguirá contradição.

Infelizmente, a compreensão predicativa não é muito *abrangente*. Afinal, introduz-nos novas entidades, conjuntos'. Logo teremos de considerar fórmulas que quantificam sobre conjuntos'. Se estas derem sempre um conjunto', então o

65.3. PREDICATIVO E IMPREDICATIVO

paradoxo de Russell surgirá de novo, bastando considerar o conjunto' de todos os conjuntos' que não são membros de si próprios. Assim, a seguir o mesmo fio de ideias, temos de dizer que uma fórmula que quantifica sobre conjuntos' produz um conjunto'' correspondente. E então precisaremos de conjuntos''', conjuntos''', etc. Para evitar uma enxurrada de primos, será mais fácil pensar nestes como conjuntos₀, conjuntos₁, conjuntos₂, conjuntos₃, conjuntos₄, ... E isto dar-nos-ia entrada na teoria (simples) dos tipos.

Há algumas objeções óbvias a tal teoria (embora não seja óbvio que sejam objeções *esmagadoras*). Em resumo: a teoria resultante é pouco manejável; é prolixa ao postular diferentes tipos de objetos; e não está claro, no fim, que as definições impredicativas sejam sequer *assim tão más*.

Para ilustrar o último ponto, consideremos esta observação de Ramsey:

podemos referir-nos a um homem como o mais alto de um grupo, identificando-o por meio de uma totalidade da qual ele próprio é membro sem que haja círculo vicioso. (Ramsey, 1925)

O ponto de Ramsey é que “o homem mais alto do grupo” é uma definição impredicativa; mas é obviamente perfeitamente legítima.

Alguém poderá responder que, neste caso, poderíamos identificar a pessoa mais alta por meios *predicativos*. Por exemplo, talvez pudéssemos simplesmente apontar para o homem em questão. A objeção às definições impredicativas, então, teria claramente de se limitar a entidades que *só* possam ser identificadas impredicativamente. Mas mesmo assim, seria preciso ouvir mais, sobre por que tal “impredicatividade essencial” seria tão grave.¹

Admitidamente, as definições impredicativas são péssimas notícias, se quisermos que as definições nos forneçam algo como uma receita para *criar* um objeto. Pois, dada uma definição impredicativa, estar-se-ia genuinamente preso a um círculo vicioso: para criar o objeto especificado impredicativamente, seria preciso *primeiro* criar todos os objetos (incluindo o objeto especificado impredicativamente), uma vez que este é especificado em termos de todos os objetos; logo seria preciso criar o objeto especificado impredicativamente antes de o ter criado. Mas, de novo, isto só é uma objeção séria a conjuntos especificados de modo “essencialmente impredicativo”, se pensarmos nos conjuntos como coisas que *criamos*. E nós (provavelmente) não os criamos.

Assim—para o bem ou para o mal—a abordagem que se tornou comum não envolve tomar uma posição rígida sobre (im)predicatividade. Envolve antes o que hoje se considera a abordagem cumulativa-iterativa. No fim, isto permitir-nos-á estratificar os conjuntos em “estágios”—um *pouco* como a abordagem predicativa estratifica entidades em conjuntos₀, conjuntos₁, conjuntos₂, ...—mas não postularemos diferença de natureza entre eles.

content/set-theory/story/cumulative-approach.tex

¹Para mais, ver Linnebo (2010).

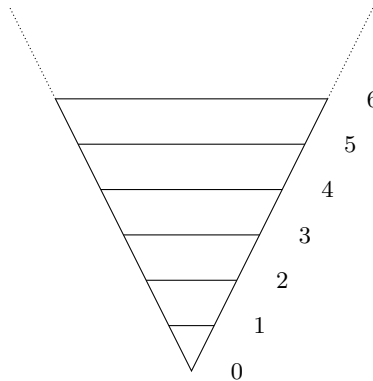
65.4 A Abordagem Cumulativa-Iterativa

Eis um enunciado um pouco mais completo de como estratificaremos os conjuntos em estágios: sth:story:approach:sec

Os conjuntos formam-se em *estágios*. Para cada estágio S , há certos estágios que são *anteriores* a S . No estágio S , cada coleção constituída por conjuntos formados em estágios anteriores a S é formada num conjunto. Não há conjuntos além dos que são formados nos estágios. (Shoenfield, 1977, p. 323)

Isto é um esboço da *concepção cumulativa-iterativa do conjunto*. Servirá de base à teoria formal dos conjuntos que apresentamos em **Parte XV**.

Exploremos isto com um pouco mais de detalhe. Como Shoenfield descreve o processo, em cada estágio formamos novos conjuntos a partir dos conjuntos que tínhamos disponíveis nos estágios anteriores. Assim, na imagem de Shoenfield, no estágio inicial, estágio 0, não há estágios *anteriores*, e portanto *a fortiori* não há conjuntos disponíveis a partir de estágios anteriores.² Logo formamos apenas um conjunto: o conjunto sem **membros** $\emptyset$. No estágio 1, está exatamente um conjunto disponível a partir de estágios anteriores, logo apenas um conjunto novo é $\{\emptyset\}$. No estágio 2, dois conjuntos estão disponíveis a partir de estágios anteriores, e formamos dois conjuntos novos $\{\{\emptyset\}\}$ e $\{\emptyset, \{\emptyset\}\}$. No estágio 3, quatro conjuntos estão disponíveis a partir de estágios anteriores, logo formamos doze conjuntos novos. . . . Assim, a imagem cumulativa-iterativa dos conjuntos ficará algo assim (com números a indicar estágios):



Então: por que deveríamos aceitar esta narrativa?

Uma razão é que é uma narrativa agradável e tratável. Dado o fim da narrativa mais óbvia, isto é, a compreensão ingênua, precisamos de algo de substância.

²Por que deveríamos supor que *existe* um primeiro estágio? Ver a nota de rodapé a *Estágios-são-ordenados* em **Seção 66.1**.

65.5. URELEMENTOS OU NÃO?

Mas a narrativa não é *apenas* agradável. Temos boa razão para acreditar que qualquer teoria dos conjuntos baseada nesta narrativa será *consistente*. Eis a razão.

Dada a concepção cumulativa-iterativa do conjunto, formamos conjuntos em estágios; e os seus **membros** têm de ser objetos que já estavam disponíveis. Assim, para qualquer estágio S , podemos formar o conjunto

$$R_S = \{x : x \notin x \text{ e } x \text{ já estava disponível antes de } S\}$$

O raciocínio envolvido na prova do paradoxo de Russell estabelece agora que o próprio R_S não está disponível antes do estágio S . E isso não é uma contradição. Além disso, se aceitarmos a concepção cumulativa-iterativa do conjunto, nem sequer deveríamos ter *esperado* poder formar o próprio conjunto de Russell. Pois esse seria o conjunto de todos os conjuntos que não são membros de si próprios que “alguma vez estarão disponíveis”. Em resumo: o fato de, provadamente, não podermos formar o conjunto de Russell não é *surpreendente*, dada a narrativa cumulativa-iterativa; é antes *previsível*.

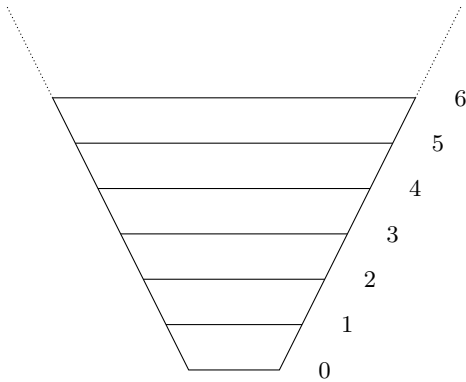
`content/set-theory/story/urelements.tex`

65.5 Urelementos ou Não?

`sth:story:urelements:`
`sec`

Nos próximos capítulos, tentaremos extrair axiomas da concepção cumulativa-iterativa do conjunto. Mas, antes de avançar mais, precisamos de dizer algo mais sobre os *urelementos*.

A imagem de **Seção 65.4** só nos permitia formar novos conjuntos a partir de *conjuntos* antigos. Contudo, podemos querer admitir que certos *não-conjuntos*—vacas, porcos, grãos de areia, ou o que seja—possam ser **membros** de conjuntos. Nesse caso, começaríamos com certos elementos básicos, *urelementos*, e diríamos que em cada estágio S formaríamos “todos os possíveis” conjuntos constituídos por urelementos ou conjuntos formados em estágios anteriores a S (em qualquer combinação). A imagem resultante parecer-se-ia mais com isto:



Temos pois uma decisão a tomar: *devemos admitir urelementos?*

Do ponto de vista filosófico, faz sentido incluir urelementos na nossa teorização. A principal razão é tornar a teoria dos conjuntos *aplicável*. Para ilustrar, recordemos de [Capítulo 4](#) que dizemos que dois conjuntos A e B têm o mesmo tamanho, isto é, $A \approx B$, sse existe uma bijeção entre eles. Agora, se as vacas no campo e os porcos no chiqueiro constituem ambos conjuntos, podemos oferecer um tratamento conjunto-teórico da afirmação “há tantas vacas como porcos”. Mas se proibirmos urelementos, de modo que as vacas e os porcos *não* constituem conjuntos, esse tratamento conjunto-teórico deixará de estar disponível. De fato, não teremos capacidade direta de aplicar a teoria dos conjuntos a mais nada além dos próprios conjuntos. (Para mais razões para incluir urelementos, ver [Potter 2004](#), pp. vi, 24, 50–1.)

Matematicamente, contudo, é bastante raro admitir urelementos. Em parte, porque é *ligeiramente* mais fácil formular a teoria dos conjuntos sem urelementos. Mas, ocasionalmente, encontram-se justificações mais interessantes para excluir urelementos da teoria dos conjuntos:

De acordo com a convicção de que a teoria dos conjuntos é o fundamento da matemática, devemos ser capazes de captar toda a matemática apenas falando de conjuntos, pelo que as nossas variáveis não devem quantificar sobre objetos como vacas e porcos. ([Kunen, 1980](#), p. 8)

Assim: um foco na aplicabilidade sugeriria *incluir* urelementos; um foco num objetivo fundacional redutivo (reduzir a matemática à teoria pura dos conjuntos) poderia sugerir *excluí-los*. Uma ligeira preguiça também aponta no sentido de excluir urelementos.

Seguiremos o caminho mais preguiçoso. Em parte, há também uma justificação pedagógica. O nosso objetivo é introduzir os elementos da teoria dos conjuntos de que precisa para começar na filosofia da teoria dos conjuntos. E a maior parte dessa literatura filosófica discute teorias dos conjuntos formuladas *sem* urelementos. Logo este livro será, talvez, mais útil se se mantiver bastante próximo dessa literatura.

[content/set-theory/story/grundgesetze.tex](#)

65.6 Apêndice: Lei Básica V de Frege

Na [Seção 65.2](#), explicamos que Russell formulou o paradoxo como um problema para o sistema que Frege delineou nos seus *Grundgesetze*. O sistema de Frege não incluía uma formulação direta da compreensão ingênua. Logo, neste apêndice, explicaremos muito sumariamente o que o sistema de Frege *incluía*, como se relaciona com a compreensão ingênua e como se relaciona com o paradoxo de Russell.

[sth:story:blv:sec](#)

O sistema de Frege é de *segunda ordem* e foi concebido para formular a noção de *extensão de um conceito*.³ Usando notação inspirada em Frege, escreveremos $\epsilon x F(x)$ para a *extensão do conceito* F . Este é um dispositivo que toma um *predicado*, “ F ”, e transforma-o num *termo* (de primeira ordem), “ $\epsilon x F(x)$ ”. Com este dispositivo, Frege propôs a seguinte *definição* de pertença:

$$a \in b =_{\text{df}} \exists G(b = \epsilon x G(x) \wedge Ga)$$

grosso modo: $a \in b$ sse a cai sob um conceito cuja extensão é b . (Note que o quantificador “ $\exists G$ ” é de segunda ordem.) Frege também sustentava o seguinte princípio, conhecido como *Lei básica V*:

$$\epsilon x F(x) = \epsilon x G(x) \leftrightarrow \forall x(Fx \leftrightarrow Gx)$$

grosso modo: conceitos têm extensões idênticas sse são coextensivos. (De novo, tanto “ F ” como “ G ” estão em posição de predicado.) Um princípio simples liga agora a pertença à satisfação de propriedades:

Lema 65.2 (em *Grundgesetze*). $\forall F \forall a(a \in \epsilon x F(x) \leftrightarrow Fa)$

sth:story:blv:
lem:Fregeextensions

Prova. Fixe F e a . Ora $a \in \epsilon x F(x)$ sse $\exists G(\epsilon x F(x) = \epsilon x G(x) \wedge Ga)$ (pela definição de pertença) sse $\exists G(\forall x(Fx \leftrightarrow Gx) \wedge Ga)$ (pela lei básica V) sse Fa (por lógica elementar de segunda ordem). $\square$

E disto resulta quase de imediato a compreensão ingênua:

Lema 65.3 (em *Grundgesetze*). $\forall F \exists s \forall a(a \in s \leftrightarrow Fa)$

Prova. Fixe F ; agora **Lema 65.2** dá $\forall a(a \in \epsilon x F(x) \leftrightarrow Fa)$; logo $\exists s \forall a(a \in s \leftrightarrow Fa)$ por generalização existencial. O resultado segue-se, pois F era arbitrário. $\square$

O paradoxo de Russell segue tomando F dado por $\forall x(Fx \leftrightarrow x \notin x)$.

Capítulo 66

Passos rumo a Z

[content/set-theory/z/story.tex](#)

³Estritamente falando, Frege tenta formalizar uma noção mais geral: o “valor de curso” de uma função. As extensões de conceitos são um caso especial da noção mais geral. Ver Heck (2012, pp. 8–9) para pormenores.

66.1 A história com mais pormenor

Na [Seção 65.4](#), citamos a descrição de Shoenfield do processo de formação de conjuntos. Queremos agora escrever mais alguns princípios, para tornar esta história um pouco mais precisa. Eis-os: sth:z:story:
sec

Estágios-são-chave. Cada conjunto forma-se por algum estágio.

Estágios-são-ordenados. Os estágios estão ordenados: alguns vêm *antes* de outros.¹

Estágios-acumulam. Para qualquer estágio S , e para quaisquer conjuntos que se formaram *antes* do estágio S : forma-se, no estágio S , um conjunto cujos membros são exatamente esses conjuntos. Mais nada se forma no estágio S .

Estes são princípios informais, mas poderemos usá-los para justificar vários dos axiomas da teoria de conjuntos de Zermelo.

(Convém uma palavra de cautela. Embora vamos apresentar alguns axiomas completamente standard, com nomes completamente standard, os princípios em *itálico* que acabamos de apresentar não têm nomes particulares na literatura. São apenas alcunhas que esperamos serem úteis.)

content/set-theory/z/separation.tex

66.2 Separação

Começamos com um princípio que substitui a compreensão ingênua: sth:z:sep:
sec

Esquema de Separação (Esquema de Separação). Para cada fórmula $\varphi(x)$, isto é um axioma: para qualquer A , o conjunto $\{x \in A : \varphi(x)\}$ existe.

Note que isto não é um único axioma. É um *esquema* de axiomas. Há *infinitamente muitos* axiomas de Separação; um para cada fórmula $\varphi(x)$. O esquema pode igualmente ser (e normalmente é) escrito do seguinte modo:

Para qualquer fórmula $\varphi(x)$ que não contenha “ S ”, isto é um axioma:

$$\forall A \exists S \forall x (x \in S \leftrightarrow (\varphi(x) \wedge x \in A)).$$

¹Assumiremos de fato—tacitamente—que os estágios estão *bem ordenados*. O que isto significa explica-se no [Capítulo 67](#). É um pressuposto substancial. De fato, usando uma técnica muito engenhosa devida a [Scott \(1974\)](#), este pressuposto pode ser *evitado* e depois *deduzido*. (Isto também explicará por que devemos pensar que existe um estágio inicial.) Não podemos aprofundar isso aqui; para mais, veja [Button \(2021\)](#).

66.2. SEPARAÇÃO

Em conformidade com a convenção assinalada no início da **Parte XV**, as fórmulas φ nos axiomas de Separação podem ter parâmetros.²

A Separação justifica-se de imediato pela concepção cumulativa iterativa de conjuntos que vamos expondo. Para ver por que, seja A um conjunto. Então A forma-se por algum estágio S (por *Estágios-são-chave*). Uma vez que A se formou no estágio S , todos os membros de A formaram-se antes do estágio S (por *Estágios-acumulam*). Consideremos em particular todos os conjuntos que são membros de A e que também satisfazem φ ; claramente todos estes conjuntos também se formaram antes do estágio S . Logo formam-se num conjunto $\{x \in A : \varphi(x)\}$ também no estágio S (por *Estágios-acumulam*).

Ao contrário da compreensão ingênua, isto evita o paradoxo de Russell. Pois não podemos simplesmente afirmar a existência do conjunto $\{x : x \notin x\}$. Pelo contrário, *dado* algum conjunto A , podemos afirmar a existência do conjunto $R_A = \{x \in A : x \notin x\}$. Mas isto só prova que $R_A \notin R_A$ e $R_A \notin A$, o que não é preocupante.

Contudo, a Separação tem uma consequência imediata e marcante:

Teorema 66.1. *Não existe conjunto universal, isto é, $\{x : x = x\}$ não existe.*

sth::z:sep:
thm::NoUniversalSet

Prova. Por redução ao absurdo, suponhamos que V é um conjunto universal. Então, por Separação, $R = \{x \in V : x \notin x\} = \{x : x \notin x\}$ existe, o que contradiz o paradoxo de Russell. $\square$

A ausência de um conjunto universal—de fato, a abertura da hierarquia de conjuntos—é uma das ideias mais fundamentais por detrás da concepção cumulativa iterativa. Por isso vale a pena ver que, intuitivamente, poderíamos chegar lá por outro caminho. Um conjunto universal teria de ser **um membro** de si próprio. Mas, na nossa concepção cumulativa iterativa, cada conjunto aparece (pela primeira vez) na hierarquia no primeiro estágio imediatamente a seguir a todos os seus **membros**. Isto implica que *nenhum* conjunto pertence a si próprio. Pois qualquer conjunto que pertencesse a si próprio teria de ocorrer pela primeira vez imediatamente depois do estágio em que ocorreu pela primeira vez, o que é absurdo. (Veremos na **Definição 68.15** como tornar esta explicação mais rigorosa, usando a noção de “posto” de um conjunto. Contudo, precisaremos de mais alguns axiomas para o fazer.)

Eis mais algumas consequências da Separação e da Extensionalidade.

Proposição 66.2. *Se existe algum conjunto, então $\emptyset$ existe.*

sth::z:sep:
prop::emptyexists

Prova. Se A é um conjunto, $\emptyset = \{x \in A : x \neq x\}$ existe por Separação. $\square$

Proposição 66.3. *$A \setminus B$ existe quaisquer que sejam os conjuntos A e B*

Prova. $A \setminus B = \{x \in A : x \notin B\}$ existe por Separação. $\square$

²Para uma explicação do que isto significa, veja a discussão imediatamente a seguir a **Corolário 6.7**.

Também se verifica que existem (quase) interseções arbitrárias:

Proposição 66.4. *Se $A \neq \emptyset$, então $\bigcap A = \{x : (\forall y \in A)x \in y\}$ existe.*

*sth::z:sep:
prop::intersectionexist*

Prova. Seja $A \neq \emptyset$, logo existe algum $c \in A$. Então $\bigcap A = \{x : (\forall y \in A)x \in y\} = \{x \in c : (\forall y \in A)x \in y\}$, que existe por Separação. $\square$

Note, porém, a condição de que $A \neq \emptyset$; pois $\bigcap \emptyset$ seria o conjunto universal, vacuamente, o que contradiz **Teorema 66.1**.

content/set-theory/z/union.tex

66.3 União

Proposição 66.4 deu-nos interseções. Mas, se quisermos que existam uniões arbitrárias, precisamos de enunciar outro axioma:

*sth::z:union:
sec*

União (União). Para qualquer conjunto A , o conjunto $\bigcup A = \{x : (\exists b \in A)x \in b\}$ existe.

$$\forall A \exists U \forall x (x \in U \leftrightarrow (\exists b \in A)x \in b)$$

Este axioma também se justifica pela concepção cumulativa iterativa. Seja A um conjunto, logo A forma-se por algum estágio S (por *Estágios-são-chave*). Cada membro de A formou-se *antes* de S (por *Estágios-acumulam*); logo, raciocinando de modo análogo, cada membro de cada membro de A formou-se antes de S . Assim, todos *esses* conjuntos estão disponíveis antes de S , para serem reunidos num conjunto em S . E esse conjunto é precisamente $\bigcup A$.

content/set-theory/z/pairs.tex

66.4 Pares

O próximo axioma a considerar é o seguinte:

*sth::z:pairs:
sec*

Pares (Pares). Quaisquer que sejam os conjuntos a, b , o conjunto $\{a, b\}$ existe.

$$\forall a \forall b \exists P \forall x (x \in P \leftrightarrow (x = a \vee x = b))$$

Eis como justificar este axioma à luz da concepção iterativa. Suponhamos que a está disponível no estágio S , e b no estágio T . Seja M o que quer que seja dos estágios S e T que ocorre mais tarde. Então, uma vez que a e b estão ambos disponíveis no estágio M , o conjunto $\{a, b\}$ é uma coleção possível disponível em qualquer estágio depois de M (o que quer que seja o posterior).

Mas espera! Porquê assumir que *existem* estágios depois de M ? Se não existir nenhum, a nossa justificação falha. Por isso, para justificar Pares, teremos de acrescentar outro princípio à história que contamos na **Seção 66.1**, nomeadamente:

66.5. CONJUNTOS POTÊNCIA

Estágios-prosseguem. Não há último estágio.

Este princípio está justificado? Nada na história de Shoenfield afirmava *explicitamente* que não há último estágio. Ainda assim, mesmo que seja (estritamente falando) um acréscimo à nossa história, encaixa bem na ideia básica de que os conjuntos se formam em estágios. Aceitá-lo-emos simplesmente no que se segue. E, assim, aceitaremos também o Axioma dos Pares.

Armados com este novo axioma, podemos provar a existência de muitos mais conjuntos. Por exemplo:

*sth::z:pairs:
prop:pairsconsequences*

Proposição 66.5. *Quaisquer que sejam os conjuntos a e b , existem os seguintes conjuntos:*

*sth::z:pairs:
singleton
sth::z:pairs:
binunion
sth::z:pairs:
tuples*

1. $\{a\}$

2. $a \cup b$

3. $\langle a, b \rangle$

Prova. (1). Por Pares, $\{a, a\}$ existe, o que é $\{a\}$ por Extensionalidade.

(2). Por Pares, $\{a, b\}$ existe. Ora, $a \cup b = \bigcup \{a, b\}$ existe por União.

(3). Por (1), $\{a\}$ existe. Por Pares, $\{a, b\}$ existe. Ora, $\{\{a\}, \{a, b\}\} = \langle a, b \rangle$ existe, de novo por Pares. $\square$

Problema 66.1. Mostre que, quaisquer que sejam os conjuntos a, b, c , o conjunto $\{a, b, c\}$ existe.

Problema 66.2. Mostre que, quaisquer que sejam os conjuntos $a_1, \dots, a_n$, o conjunto $\{a_1, \dots, a_n\}$ existe.

content/set-theory/z/powerset.tex

66.5 Conjuntos potência

*sth::z:power:
sec*

Prosseguiremos com outro axioma:

Conjuntos potência (Conjuntos potência). Para qualquer conjunto A , existe o conjunto $\wp(A) = \{x : x \subseteq A\}$.

$$\forall A \exists P \forall x (x \in P \leftrightarrow (\forall z \in x) z \in A)$$

A nossa justificação é bastante direta. Suponhamos que A se forma no estágio S . Então todos os membros de A estavam disponíveis antes de S (por *Estágios-acumulam*). Logo, raciocinando como na justificação da Separação, cada subconjunto de A forma-se até ao estágio S . Estão todos disponíveis, para serem reunidos num único conjunto, em qualquer estágio depois de S . E sabemos que existe algum tal estágio, pois S não é o último estágio (por *Estágios-prosseguem*). Logo $\wp(A)$ existe.

Eis uma boa consequência dos conjuntos potência:

Proposição 66.6. *Dados quaisquer conjuntos A, B , o seu produto cartesiano $A \times B$ existe.*

Prova. O conjunto $\wp(\wp(A \cup B))$ existe por Conjuntos potência e **Proposição 66.5**. Logo, por Separação, existe este conjunto:

$$C = \{z \in \wp(\wp(A \cup B)) : (\exists x \in A)(\exists y \in B)z = \langle x, y \rangle\}.$$

Agora, para quaisquer $x \in A$ e $y \in B$, o conjunto $\langle x, y \rangle$ existe por **Proposição 66.5**. Além disso, uma vez que $x, y \in A \cup B$, temos $\{x\}, \{x, y\} \in \wp(A \cup B)$, e $\langle x, y \rangle \in \wp(\wp(A \cup B))$. Logo $A \times B = C$. $\square$

Nesta prova, o axioma dos conjuntos potência interage com a Separação. E isso não surpreende. Sem Separação, o princípio dos conjuntos potência não seria muito *poderoso*. Afinal, a Separação nos diz que subconjuntos de um conjunto existem e, por conseguinte, determina quão “gordo” é cada conjunto potência.

Problema 66.3. Mostre que, quaisquer que sejam os conjuntos A, B : (i) existe o conjunto de todas as relações com domínio A e contradomínio B ; e (ii) existe o conjunto de todas as funções de A para B .

Problema 66.4. Seja A um conjunto, e seja $\sim$ uma relação de equivalência em A . Prove que existe o conjunto das classes de equivalência sob $\sim$ em A , isto é, $A/\sim$.

content/set-theory/z/infinity-again.tex

66.6 Infinito

Já temos axiomas suficientes para garantir que há infinitamente muitos conjuntos (se é que existe algum). De fato, suponhamos que algum conjunto existe; então $\emptyset$ existe (por **Proposição 66.2**). Agora, para qualquer conjunto x , o conjunto $x \cup \{x\}$ existe por **Proposição 66.5**. Logo, aplicando isto algumas vezes, obtemos conjuntos como se segue:

0. $\emptyset$
1. $\{\emptyset\}$
2. $\{\emptyset, \{\emptyset\}\}$
3. $\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$
4. $\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}\}$

sth:z:infinity-again:
sec

66.6. INFINITO

e podemos verificar que cada um destes conjuntos é distinto.

Começamos a numeração em 0, por várias razões. Mas uma delas é esta. Não é difícil verificar que o conjunto que rotulamos “ n ” tem exatamente n membros e que (intuitivamente) se forma no n -ésimo estágio.

Contudo. Isto nos dá *infinitamente muitos* conjuntos, mas não garante que exista um conjunto *infinito*, isto é, um conjunto com infinitamente muitos membros. E isto importa deveras: a menos que encontremos um conjunto infinito de Dedekind, não podemos construir uma álgebra de Dedekind. Mas queremos uma álgebra de Dedekind, para a podermos tratar como o conjunto dos números naturais. (Compare-se com [Seção 6.4.](#))

Importa que os axiomas que estabelecemos até aqui *não* garantem a existência de nenhum conjunto infinito. Por isso temos de enunciar um novo axioma:

Infinito (Infinito). Existe um conjunto, I , tal que $\emptyset \in I$ e $x \cup \{x\} \in I$ sempre que $x \in I$.

$$\exists I((\exists o \in I)\forall x (x \notin o \wedge (\forall x \in I)(\exists s \in I)\forall z(z \in s \leftrightarrow (z \in x \vee z = x))))$$

É fácil ver que o conjunto I que nos é dado pelo Axioma do Infinito é infinito de Dedekind. O seu elemento distinguido é $\emptyset$, e a injeção em I é dada por $s(x) = x \cup \{x\}$. Agora, [Teorema 6.5](#) mostrou como extrair uma álgebra de Dedekind de um conjunto infinito de Dedekind; e trataremos isso como o nosso conjunto de números naturais. Mais precisamente:

[sth:z:infinity-again:](#)
[defnomega](#)

Definição 66.7. Seja I qualquer conjunto que nos seja dado pelo Axioma do Infinito. Seja s a função $s(x) = x \cup \{x\}$. Seja $\omega = \text{clo}_s(\emptyset)$. Chamamos *números naturais* aos membros de ω , e dizemos que n é o resultado de n aplicações de s a $\emptyset$.

Pode agora voltar atrás e verificar que o conjunto rotulado “ n ”, há uns parágrafos, será tratado *como* o número n .

Discutiremos o significado desta estipulação na [Seção 66.8](#). Por ora, permitenos provar um resultado intuitivo:

[sth:z:infinity-again:](#)
[naturalnumbersarentinfinite](#)

Proposição 66.8. *Nenhum número natural é infinito de Dedekind.*

Prova. A prova é por indução, isto é, [Teorema 6.6](#). Claramente $0 = \emptyset$ não é infinito de Dedekind. Para o passo indutivo, estabeleceremos a contrapositiva: se (absurdamente) $s(n)$ for infinito de Dedekind, então n é infinito de Dedekind.

Suponhamos, pois, que $s(n)$ é infinito de Dedekind, isto é, que existe alguma *injeção* f com $\text{Im}(f) \subsetneq \text{dom}(f) = s(n) = n \cup \{n\}$. Há dois casos a considerar.

Caso 1: $n \notin \text{Im}(f)$. Então $\text{Im}(f) \subseteq n$, e $f(n) \in n$. Seja $g = f|_n$; agora $\text{Im}(g) = \text{Im}(f) \setminus \{f(n)\} \subsetneq n = \text{dom}(g)$. Logo n é infinito de Dedekind.

Caso 2: $n \in \text{Im}(f)$. Fixemos $m \in \text{dom}(f) \setminus \text{Im}(f)$, e definamos uma função h com domínio $s(n) = n \cup \{n\}$:

$$h(x) = \begin{cases} f(x) & \text{se } f(x) \neq n \\ m & \text{se } f(x) = n \end{cases}$$

Assim h e f coincidem em toda a parte, exceto que $h(f^{-1}(n)) = m \neq n = f(f^{-1}(n))$. Uma vez que f é uma injeção, $n \notin \text{Im}(h)$; e $\text{Im}(h) \subsetneq \text{dom}(h) = s(n)$. Logo n é infinito de Dedekind, pelo argumento do Caso 1. $\square$

A questão que permanece é como poderíamos *justificar* o Axioma do Infinito. A resposta curta é que precisaremos de acrescentar outro princípio à história que vamos contando. Esse princípio é o seguinte:

Estágios-atingem-o-infinito. Há um estágio infinito. Isto é, há um estágio que (a) não é o primeiro estágio, e que (b) tem alguns estágios antes de si, mas que (c) não tem predecessor imediato.

O Axioma do Infinito segue-se diretamente deste princípio. Sabemos que o número natural n se forma no estágio n . Logo o conjunto ω forma-se no primeiro estágio infinito. E o próprio ω testemunha o Axioma do Infinito.

Isto, porém, apenas nos leva de volta à questão de como justificar *Estágios-atingem-o-infinito*. Tal como *Estágios-prosseguem*, não fazia parte explícita da história que contamos sobre a hierarquia cumulativa iterativa. Mas mais do que isso: nada na própria ideia de uma hierarquia iterativa, em que os conjuntos se formam estágio a estágio, nos obriga a pensar que o processo envolve um estágio *infinito*. Parece perfeitamente coerente pensar que os estágios estão ordenados como os números naturais.

Isto, contudo, levanta um problema óbvio. Na [Seção 6.4](#), consideramos a “prova” de Dedekind de que existe um conjunto infinito de Dedekind (de pensamentos). Isto poderá não lhe ter parecido muito satisfatório. Mas se *Estágios-atingem-o-infinito* não nos é “imposto” pela concepção iterativa de conjunto (nem pelas “leis do pensamento”), continuamos sem uma justificação intrínseca para a afirmação de que existe um conjunto infinito de Dedekind.

Há muito mais a dizer aqui, naturalmente. Mas esperamos que esteja agora num ponto em que possa começar a refletir sobre o que poderá *custar* justificar um axioma (ou um princípio). No que se segue tomaremos simplesmente *Estágios-atingem-o-infinito* como concedido.

`content/set-theory/z/milestone.tex`

66.7 $\mathbf{Z}^-$: um marco

Voltaremos a *Estágios-atingem-o-infinito* na parte seguinte. Contudo, com o Axioma do Infinito, atingimos um marco importante. Temos agora todos os axiomas necessários para a teoria $\mathbf{Z}^-$. Em detalhe:

Definição 66.9. A teoria $\mathbf{Z}^-$ tem estes axiomas: Extensionalidade, União, Pares, Conjuntos potência, Infinito, e todas as instâncias do esquema de Separação.

sth:z:milestone:
sec

66.8. A ESCOLHER OS NÚMEROS NATURAIS

O nome abrevia *Zermelo* (*menos* algo que abordaremos mais tarde). Zermelo merece a honra, pois formulou essencialmente esta teoria no seu trabalho de 1908a.³

Esta teoria é bastante poderosa para permitir-nos fazer uma quantidade enorme de matemática. Em particular, *deverá* voltar atrás na **Parte I** e convencer-se de que tudo o que fizemos, ingenuamente, pode ser feito de modo mais formal dentro de $\mathbf{Z}^-$. (Depois de o fazer durante algum tempo, poderá querer avançar e ler a **Seção 66.9**.) Por isso, daqui em diante, e sem mais comentários, assumiremos estar a trabalhar em $\mathbf{Z}^-$ (pelo menos).

content/set-theory/z/nat.tex

66.8 A escolher os números naturais

sth:z:nat:
sec

Na **Definição 66.7**, definimos explicitamente a expressão “números naturais”. Como deve entender esta estipulação? Não é uma afirmação metafísica, mas apenas uma decisão de *tratar* certos conjuntos como os números naturais. Já abordamos razões para pensar assim na **Seção 2.2**, na **Seção 5.5** e na **Seção 6.4**. Mas podemos tornar essas razões ainda mais nítidas.

O nosso Axioma do Infinito segue von Neumann (1925). Mas eis outro axioma, que poderíamos ter adotado em alternativa:

Axioma do Infinito de Zermelo 1908a. Existe um conjunto A tal que $\emptyset \in A$ e $(\forall x \in A)\{x\} \in A$.

Se tivéssemos usado o axioma de Zermelo em vez do nosso Axioma do Infinito (inspirado em von Neumann), ter-nos-iam sido dado igualmente um conjunto infinito de Dedekind e, portanto, uma álgebra de Dedekind. Na abordagem de Zermelo, o elemento distinguido da nossa álgebra seria de novo $\emptyset$ (o nosso sucedâneo de 0), mas a injeção seria dada pela aplicação $x \mapsto \{x\}$, em vez de $x \mapsto x \cup \{x\}$. A consequência mais simples é que Zermelo trata 2 como $\{\{\emptyset\}\}$, ao passo que nós (com von Neumann) tratamos 2 como $\{\emptyset, \{\emptyset\}\}$.

Porquê escolher um axioma do Infinito em vez do outro? A razão prática principal é que a abordagem de von Neumann “escala” bastante bem para números transfinitos. Exploraremos isto a partir do **Capítulo 67** em diante. Porém, na simples perspectiva de *fazer aritmética*, ambas as abordagens serviriam igualmente bem. Por isso, se alguém lhe disser que os números naturais *são* conjuntos, a pergunta óbvia é: *Que conjuntos são eles?*

Esta pergunta precisa foi tornada célebre por Benacerraf (1965). Mas vale a pena sublinhar que é apenas o exemplo mais famoso de um fenómeno que já encontramos muitas vezes. O ponto essencial é este. A teoria dos conjuntos nos dá uma maneira de *simular* vários tipos de entidades “intuitivas”: os reais, os

³Para comentários interessantes sobre a história e os pormenores técnicos, veja Potter (2004, Apêndice A).

racionais, os inteiros e os naturais, sim; mas também pares ordenados, funções e relações. Contudo, a teoria dos conjuntos nunca nos fornece uma escolha *única* de simulação. Há *sempre* alternativas que—de modo direto—nos teriam servido tão bem.

content/set-theory/z/arbintersections.tex

66.9 Apêndice: Fechamento, compreensão e interseção

Na [Seção 66.7](#), sugerimos que voltasse atrás no trabalho ingênuo da [Parte I](#) e verificasse que este pode ser realizado em $\mathbf{Z}^-$. Se seguiu esse conselho, *um* ponto poderá tê-lo feito tropeçar: o uso da *interseção* no tratamento por Dedekind dos *fechamentos*.

sth:z:arbintersections:
sec

Recordemos, de [Definição 6.2](#), que

$$\text{clo}_f(o) = \bigcap \{X : o \in X \text{ e } X \text{ é } f\text{-fechado}\}.$$

A forma geral disto é uma definição da forma:

$$C = \bigcap \{X : \varphi(X)\}.$$

Mas isto deve fazer soar o alarme: uma vez que a compreensão ingênua falha, não há garantia de que $\{X : \varphi(X)\}$ exista. Parece, então, perigosamente, que tais definições são *truques*.

Felizmente, não são truques; ou melhor, se *o forem* tal como estão, então podemos empenhar-nos honestamente para as tornar legítimas. Esse trabalho honesto foi antecipado na [Proposição 66.4](#), quando explicamos por que $\bigcap A$ existe para qualquer $A \neq \emptyset$. Mas vamos explicitá-lo.

Dada a Extensionalidade, se tentarmos definir C como $\bigcap \{X : \varphi(X)\}$, tudo o que realmente pedimos é um objeto C que satisfaça o seguinte:

$$\forall x(x \in C \leftrightarrow \forall X(\varphi(X) \rightarrow x \in X)) \tag{*}$$

sth:z:arbintersections:
bicondelimarbintersection

Suponhamos, agora, que existe *algum* conjunto, S , tal que $\varphi(S)$. Então, para satisfazer [Eq. \(*\)](#), podemos simplesmente definir C por *Separação*, do seguinte modo:

$$C = \{x \in S : \forall X(\varphi(X) \rightarrow x \in X)\}.$$

Deixamos como exercício verificar que esta definição produz [Eq. \(*\)](#), como desejado. E esta estratégia geral permitir-nos-á contornar qualquer uso aparente da compreensão ingênua na definição de interseções. No caso particular que nos levou a esta linha de raciocínio, nomeadamente o de $\text{clo}_f(o)$, eis como isso funcionaria. Começamos a prova de [Lema 6.3](#) ao notar que $o \in \text{Im}(f) \cup \{o\}$ e que $\text{Im}(f) \cup \{o\}$ é f -fechado. Assim, podemos definir o que queremos assim:

$$\text{clo}_f(o) = \{x \in \text{Im}(f) \cup \{o\} : (\forall X \ni o)(X \text{ é } f\text{-fechado} \rightarrow x \in X)\}.$$

Capítulo 67

Ordinais

`content/set-theory/ordinals/introduction.tex`

67.1 Introdução

`sth:ordinals:intro:`
`sec`

No **Capítulo 66**, postulamos que existe um estágio infinito da hierarquia, na forma de *Estágios-atingem-o-infinito* (ver também o nosso axioma do Infinito). Contudo, dado *Estágios-prosseguem*, não podemos parar no estágio infinito; temos de continuar. Assim: no estágio seguinte ao primeiro estágio infinito, formamos todas as coleções possíveis de conjuntos que estavam disponíveis no primeiro estágio infinito; e repetimos; e repetimos; e repetimos; ...

Implicitamente, o que aqui aconteceu foi começarmos a invocar uma noção “intuitiva” de número, segundo a qual podem existir números *depois* de todos os números naturais. Em particular, a noção em causa é a de *ordinal transfinito*. O objetivo deste capítulo é tornar esta ideia mais rigorosa. Exploraremos a noção geral de ordinal e depois definiremos explicitamente certos conjuntos para serem os nossos ordinais.

`content/set-theory/ordinals/idea.tex`

67.2 A Ideia Geral de um Ordinal

`sth:ordinals:idea:`
`sec`

Considere os números naturais, na sua ordem habitual:

$$0 < 1 < 2 < 3 < 4 < 5 < \dots$$

Chamamos a isto, na gíria, uma ω -sucessão. E, de fato, esta ordenação geral espelha-se na nossa construção inicial dos estágios da hierarquia de conjuntos. Mas suponha agora que movemos 0 para o fim desta sucessão, de modo que fica depois de todos os outros números:

$$1 < 2 < 3 < 4 < 5 < \dots < 0$$

Temos aqui as mesmas entidades, mas ordenadas de modo fundamentalmente diferente: a nossa primeira ordenação não tinha último elemento; a nova tem. De fato, a nova ordenação consiste numa ω -sucessão de entidades $(1, 2, 3, 4, 5, \dots)$, seguida de outra entidade. Será uma sucessão do tipo $\omega + 1$.

Podemos gerar ainda mais tipos de ordenação, usando apenas estas entidades. Por exemplo, considere todos os números pares (na sua ordem natural) seguidos de todos os ímpares (na sua ordem natural):

$$0 < 2 < 4 < \dots < 1 < 3 < \dots$$

Isto é uma ω -sucessão seguida de outra ω -sucessão; uma sucessão do tipo $\omega + \omega$.

Pois bem, podemos continuar. Mas o que gostaríamos é de ter uma forma geral de entender esta linguagem sobre *ordenações*.

[content/set-theory/ordinals/wo.tex](#)

67.3 Boas Ordenações

A noção fundamental é a seguinte:

[sth:ordinals:wo:
sec](#)

Definição 67.1. A relação $<$ *bem ordena* A sse cumpre estas duas condições:

1. $<$ é conexa, isto é, para todos $a, b \in A$, ou $a < b$ ou $a = b$ ou $b < a$;
2. todo o subconjunto não vazio de A tem um **membro** $<$ -minimal, isto é, se $\emptyset \neq X \subseteq A$ então $(\exists m \in X)(\forall z \in X)z \not< m$

É fácil ver que os três exemplos que acabamos de considerar eram de fato relações de boa ordenação.

Problema 67.1. [Seção 67.2](#) apresentou três ordenações exemplo sobre os números naturais. Verifique que cada uma é uma boa ordenação.

Eis algumas observações elementares mas extremamente importantes sobre boa ordenação.

Proposição 67.2. *Se $<$ bem ordena A , então todo o subconjunto não vazio de A tem um único membro $<$ -mínimo, e $<$ é irreflexiva, assimétrica e transitiva.*

[sth:ordinals:wo:
wo:strictorder](#)

Prova. Se X é um subconjunto não vazio de A , tem um **membro** $<$ -minimal m , isto é, $(\forall z \in X)z \not< m$. Como $<$ é conexa, $(\forall z \in X)m \leq z$. Logo m é o **membro** $<$ -mínimo de X .

Para a irreflexividade, fixe $a \in A$; o **membro** $<$ -mínimo de $\{a\}$ é a , logo $a \not< a$. Para a transitividade, se $a < b < c$, então como $\{a, b, c\}$ tem um **membro** $<$ -mínimo, $a < c$. A assimetria segue da irreflexividade e da transitividade $\square$

Proposição 67.3. *Se $<$ bem ordena A , então para qualquer fórmula $\varphi(x)$:*

[sth:ordinals:wo:
propwoinduction](#)

$$\text{se } (\forall a \in A)((\forall b < a)\varphi(b) \rightarrow \varphi(a)), \text{ então } (\forall a \in A)\varphi(a).$$

67.4. ISOMORFISMOS DE ORDEM

Prova. Provaremos a contrapositiva. Suponha $\neg(\forall a \in A)\varphi(a)$, isto é, que $X = \{x \in A : \neg\varphi(x)\} \neq \emptyset$. Então X tem um **membro** $<$ -minimal, a . Logo $(\forall b < a)\varphi(b)$ mas $\neg\varphi(a)$. $\square$

Esta última propriedade deve recordar-lhe o princípio de indução forte nos naturais, isto é: se $(\forall n \in \omega)((\forall m < n)\varphi(m) \rightarrow \varphi(n))$, então $(\forall n \in \omega)\varphi(n)$. E esta propriedade torna a boa ordenação numa noção muito *robusta*.¹

content/set-theory/ordinals/iso.tex

67.4 Isomorfismos de Ordem

sth:ordinals:iso:
sec Para explicar *quão* robusta é a boa ordenação, começaremos por introduzir um método para comparar boas ordenações.

Definição 67.4. Uma *boa ordenação* é um par $\langle A, < \rangle$, tal que $<$ bem ordena A . As boas ordenações $\langle A, < \rangle$ e $\langle B, \leq \rangle$ são *isomorfas por ordem* sse existe **uma bijeção** $f: A \rightarrow B$ tal que: $x < y$ sse $f(x) \leq f(y)$. Neste caso, escrevemos $\langle A, < \rangle \cong \langle B, \leq \rangle$, e dizemos que f é um *isomorfismo de ordem*.

Daqui em diante, por brevidade, falaremos em “isomorfismos” em vez de “isomorfismos de ordem”. Intuitivamente, os isomorfismos são **bijeções** que preservam estrutura. Eis alguns fatos simples sobre isomorfismos.

sth:ordinals:iso:
isoscompose **Lema 67.5.** *Composições de isomorfismos são isomorfismos, isto é: se $f: A \rightarrow B$ e $g: B \rightarrow C$ são isomorfismos, então $(g \circ f): A \rightarrow C$ é um isomorfismo.*

Problema 67.2. Prove **Lema 67.5**.

Prova. Fica como exercício. $\square$

sth:ordinals:iso:
ordisoisquiv **Corolário 67.6.** *$X \cong Y$ é uma relação de equivalência.*

sth:ordinals:iso:
ordisounique **Proposição 67.7.** *Se $\langle A, < \rangle$ e $\langle B, \leq \rangle$ são boas ordenações isomorfas, então o isomorfismo entre elas é único.*

Prova. Sejam f e g isomorfismos $A \rightarrow B$. Provaremos o resultado por indução, isto é, usando **Proposição 67.3**. Fixe $a \in A$, e suponha (para indução) que $(\forall b < a)f(b) = g(b)$. Fixe $x \in B$.

Se $x \leq f(a)$, então $f^{-1}(x) < a$, logo $g(f^{-1}(x)) \leq g(a)$, invocando o fato de f e g serem isomorfismos. Mas como $f^{-1}(x) < a$, pela suposição $x = f(f^{-1}(x)) = g(f^{-1}(x))$. Logo $x \leq g(a)$. Analogamente, se $x \leq g(a)$ então $x \leq f(a)$.

Generalizando, $(\forall x \in B)(x \leq f(a) \leftrightarrow x \leq g(a))$. Segue que $f(a) = g(a)$ por **Proposição 2.27**. Logo $(\forall a \in A)f(a) = g(a)$ por **Proposição 67.3**. $\square$

¹Lembrete: todas as fórmulas podem ter parâmetros (salvo indicação em contrário).

Isto dá alguma ideia de que as boas ordenações são robustas. Mas para continuar a explicar isto, convém introduzir mais notação.

Definição 67.8. Quando $\langle A, < \rangle$ é uma boa ordenação com $a \in A$, seja $A_a = \{x \in A : x < a\}$. Dizemos que A_a é um *segmento inicial* próprio de A (e permitimos que o próprio A seja um segmento inicial impróprio de A). Seja $<_a$ a restrição de $<$ ao segmento inicial, isto é, $< \upharpoonright_{A_a}$.

Com esta notação, podemos enunciar e provar que nenhuma boa ordenação é isomorfa a nenhum dos seus segmentos iniciais próprios.

Lema 67.9. *Se $\langle A, < \rangle$ é uma boa ordenação com $a \in A$, então $\langle A, < \rangle \not\cong \langle A_a, <_a \rangle$* sth:ordinals:iso:
wellordnotinitial

Prova. Por redução ao absurdo, suponha $f: A \rightarrow A_a$ um isomorfismo. Como f é uma bijeção e $A_a \subsetneq A$, usando **Proposição 67.2** seja $b \in A$ o **membro** $<$ -mínimo de A tal que $b \neq f(b)$. Mostraremos que $(\forall x \in A)(x < b \leftrightarrow x < f(b))$, donde seguirá, por **Proposição 2.27**, que $b = f(b)$, completando a redução ao absurdo.

Suponha $x < b$. Logo $x = f(x)$, pela escolha de b . E $f(x) < f(b)$, por f ser um isomorfismo. Logo $x < f(b)$.

Suponha $x < f(b)$. Logo $f^{-1}(x) < b$, por f ser um isomorfismo, e portanto $f^{-1}(x) = x$ pela escolha de b . Logo $x < b$. □

O nosso resultado seguinte mostra, grosso modo, que um “segmento inicial” de um isomorfismo é um isomorfismo:

Lema 67.10. *Sejam $\langle A, < \rangle$ e $\langle B, < \rangle$ boas ordenações. Se $f: A \rightarrow B$ é um isomorfismo e $a \in A$, então $f \upharpoonright_{A_a}: A_a \rightarrow B_{f(a)}$ é um isomorfismo.* sth:ordinals:iso:
wellordinitialsegment

Prova. Como f é um isomorfismo:

$$\begin{aligned} f[A_a] &= f[\{x \in A : x < a\}] \\ &= f[\{f^{-1}(y) \in A : f^{-1}(y) < a\}] \\ &= \{y \in B : y < f(a)\} \\ &= B_{f(a)} \end{aligned}$$

E $f \upharpoonright_{A_a}$ preserva a ordem porque f preserva. □

Os nossos dois resultados seguintes estabelecem que as boas ordenações são sempre comparáveis:

Lema 67.11. *Sejam $\langle A, < \rangle$ e $\langle B, < \rangle$ boas ordenações. Se $\langle A_{a_1}, <_{a_1} \rangle \cong \langle B_{b_1}, <_{b_1} \rangle$ e $\langle A_{a_2}, <_{a_2} \rangle \cong \langle B_{b_2}, <_{b_2} \rangle$, então $a_1 < a_2$ sse $b_1 < b_2$* sth:ordinals:iso:
lemordsegments

Prova. Provaremos da *esquerda para a direita*; a outra direção é análoga. Suponha ambos $\langle A_{a_1}, <_{a_1} \rangle \cong \langle B_{b_1}, <_{b_1} \rangle$ e $\langle A_{a_2}, <_{a_2} \rangle \cong \langle B_{b_2}, <_{b_2} \rangle$, com $f: A_{a_2} \rightarrow B_{b_2}$ o nosso isomorfismo. Seja $a_1 < a_2$; então $\langle A_{a_1}, <_{a_1} \rangle \cong \langle B_{f(a_1)}, <_{f(a_1)} \rangle$ por **Lema 67.10**. Logo $\langle B_{b_1}, <_{b_1} \rangle \cong \langle B_{f(a_1)}, <_{f(a_1)} \rangle$, e portanto $b_1 = f(a_1)$ por **Lema 67.9**. Ora, $b_1 < b_2$ pois o domínio de f é B_{b_2} . □

67.5. CONSTRUÇÃO DE VON NEUMANN

sth:ordinals:iso:
thm:woalwayscomparable

Teorema 67.12. *Dadas quaisquer duas boas ordenações, uma é isomorfa a um segmento inicial (não necessariamente próprio) da outra.*

Prova. Sejam $\langle A, < \rangle$ e $\langle B, \triangleleft \rangle$ boas ordenações. Usando a Separação, seja

$$f = \{ \langle a, b \rangle \in A \times B : \langle A_a, <_a \rangle \cong \langle B_b, \triangleleft_b \rangle \}.$$

Por **Lema 67.11**, $a_1 < a_2$ sse $b_1 \triangleleft b_2$ para todos $\langle a_1, b_1 \rangle, \langle a_2, b_2 \rangle \in f$. Logo $f: \text{dom}(f) \rightarrow \text{Im}(f)$ é um isomorfismo.

Se $a_2 \in \text{dom}(f)$ e $a_1 < a_2$, então $a_1 \in \text{dom}(f)$ por **Lema 67.10**; logo $\text{dom}(f)$ é um segmento inicial de A . Analogamente, $\text{Im}(f)$ é um segmento inicial de B . Por redução ao absurdo, suponha que ambos são segmentos iniciais próprios. Então seja a o **membro** $<$ -mínimo de $A \setminus \text{dom}(f)$, de modo que $\text{dom}(f) = A_a$, e seja b o **membro** $\triangleleft$ -mínimo de $B \setminus \text{Im}(f)$, de modo que $\text{Im}(f) = B_b$. Logo $f: A_a \rightarrow B_b$ é um isomorfismo, e portanto $\langle a, b \rangle \in f$, contradição. $\square$

content/set-theory/ordinals/vn.tex

67.5 Construção de Von Neumann dos Ordinais

sth:ordinals:vn:
sec

Teorema 67.12 sugere uma ideia. Poderíamos introduzir certos objetos, chamados *tipos de ordem*, para servirem de representantes das boas ordenações. Escrevendo $\text{ord}(A, <)$ para o tipo de ordem da boa ordenação $\langle A, < \rangle$, gostaríamos de garantir os dois princípios seguintes:

$$\begin{aligned} \text{ord}(A, <) &= \text{ord}(B, \triangleleft) \text{ sse } \langle A, < \rangle \cong \langle B, \triangleleft \rangle \\ \text{ord}(A, <) &< \text{ord}(B, \triangleleft) \text{ sse } \langle A, < \rangle \cong \langle B_b, \triangleleft_b \rangle \text{ para algum } b \in B \end{aligned}$$

Além disso, poderíamos esperar introduzir tipos de ordem *como certos conjuntos*, tal como podemos introduzir os números naturais como certos conjuntos.

A forma mais comum de o fazer—e a abordagem que seguiremos—é definir estes tipos de ordem via certos conjuntos bem ordenados *canônicos*. Estes conjuntos canônicos foram introduzidos pela primeira vez por von Neumann:

Definição 67.13. O conjunto A é *transitivo* sse $(\forall x \in A)x \subseteq A$. Então A é um *ordinal* sse A é transitivo e bem ordenado por $\in$.

Daqui em diante, usaremos letras gregas para ordinais. Segue-se imediatamente da definição que, se α é um ordinal, então $\langle \alpha, \in_\alpha \rangle$ é uma boa ordenação, onde $\in_\alpha = \{ \langle x, y \rangle \in \alpha^2 : x \in y \}$. Assim, abusando um pouco da notação, podemos dizer que o próprio α é uma boa ordenação.

Eis os primeiros ordinais:

$$\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}, \dots$$

Repare que estes são precisamente os primeiros ordinais que encontramos no nosso Axioma do Infinito, isto é, na definição de von Neumann de ω (ver

[Seção 66.6](#)). Isto não é coincidência. A definição de von Neumann dos ordinais trata os números naturais como ordinais, mas permite também ordinais transfinitos.

Como sempre, podemos agora perguntar: *são* estes os ordinais? Ou von Neumann deu-nos apenas alguns conjuntos que podemos *tratar* como os ordinais? O tipo de discussões que se podem ter sobre esta questão é semelhante às que tivemos na [Seção 2.2](#), [Seção 5.5](#), [Seção 6.4](#), e [Seção 66.8](#), pelo que não insistiremos no ponto. Em vez disso, daqui em diante usaremos “os ordinais” para falar dos “ordinais de von Neumann”.

content/set-theory/ordinals/basic.tex

67.6 Propriedades Básicas dos Ordinais

Observamos que os primeiros ordinais são os números naturais. A razão principal para desenvolver uma teoria dos ordinais é alargar o princípio de indução que vale nos números naturais. Iremos construir isso através duma sucessão de resultados elementares.

sth:ordinals:basic:
sec

Lema 67.14. *Todo o **membro** de um ordinal é um ordinal.*

sth:ordinals:basic:
ordmemberord

Prova. Seja α um ordinal com $b \in \alpha$. Como α é transitivo, $b \subseteq \alpha$. Logo $\in$ bem ordena b pois $\in$ bem ordena α .

Para ver que b é transitivo, suponha $x \in c \in b$. Logo $c \in \alpha$ pois $b \subseteq \alpha$. Outra vez, como α é transitivo, $c \subseteq \alpha$, de modo que $x \in \alpha$. Logo $x, c, b \in \alpha$. Mas $\in$ bem ordena α , logo $\in$ é uma relação transitiva em α por [Proposição 67.2](#). Logo, como $x \in c \in b$, temos $x \in b$. Generalizando, $c \subseteq b$ □

Corolário 67.15. $\alpha = \{\beta \in \alpha : \beta \text{ é um ordinal}\}$, para qualquer ordinal α

sth:ordinals:basic:
ordissetofsmallerord

Prova. Imediato a partir de [Lema 67.14](#). □

A ideia grosseira dos dois resultados principais seguintes, [Teorema 67.16](#) e [Teorema 67.17](#), é que os ordinais eles próprios são bem ordenados por pertença:

Teorema 67.16 (Indução Transfinita). *Para qualquer fórmula $\varphi(x)$:*

sth:ordinals:basic:
ordinductionschema

$$\text{se } \exists \alpha \varphi(\alpha), \text{ então } \exists \alpha (\varphi(\alpha) \wedge (\forall \beta \in \alpha) \neg \varphi(\beta))$$

onde os quantificadores exibidos ficam implicitamente restritos a ordinais.

Prova. Suponha $\varphi(\alpha)$, para algum ordinal α . Se $(\forall \beta \in \alpha) \neg \varphi(\beta)$, terminamos. Caso contrário, como α é um ordinal, tem um **membro** $\in$ -mínimo que verifica φ , e este é um ordinal por [Lema 67.14](#). □

67.6. PROPRIEDADES BÁSICAS DOS ORDINAIS

Repare que podemos exprimir igualmente **Teorema 67.16** como o esquema:

$$\text{se } \forall \alpha ((\forall \beta \in \alpha) \varphi(\beta) \rightarrow \varphi(\alpha)), \text{ então } \forall \alpha \varphi(\alpha)$$

tomando simplesmente $\neg \varphi(\alpha)$ no **Teorema 67.16**, e depois fazendo manipulações lógicas elementares.

*sth:ordinals:basic:
ordtrichotomy*

Teorema 67.17 (Tricotomia). $\alpha \in \beta \vee \alpha = \beta \vee \beta \in \alpha$, para quaisquer ordinais α e β .

Prova. A prova é por dupla indução, isto é, usando **Teorema 67.16** duas vezes. Diga-se que x é *comparável* com y sse $x \in y \vee x = y \vee y \in x$.

Para a indução, suponha que todo o ordinal em α é comparável com *todo* o ordinal. Para uma segunda indução, suponha que α é comparável com todo o ordinal em β . Mostraremos que α é comparável com β . Por indução em β , seguirá que α é comparável com todo o ordinal; e por indução em α , *todo* o ordinal é comparável com *todo* o ordinal, como pretendido. Basta supor que $\alpha \notin \beta$ e $\beta \notin \alpha$, e mostrar que $\alpha = \beta$.

Para mostrar que $\alpha \subseteq \beta$, fixe $\gamma \in \alpha$; este é um ordinal por **Lema 67.14**. Logo, pela primeira hipótese de indução, γ é comparável com β . Mas se $\gamma = \beta$ ou $\beta \in \gamma$ então $\beta \in \alpha$ (invocando a transitividade de α se necessário), contra a suposição; logo $\gamma \in \beta$. Generalizando, $\alpha \subseteq \beta$.

Raciocínio exatamente análogo, usando a segunda hipótese de indução, mostra que $\beta \subseteq \alpha$. Logo $\alpha = \beta$. $\square$

Assim, por vezes escreveremos $\alpha < \beta$ em vez de $\alpha \in \beta$, já que $\in$ se comporta como relação de ordem. Não há razões profundas para além da familiaridade, e porque é mais fácil escrever $\alpha \leq \beta$ do que $\alpha \in \beta \vee \alpha = \beta$.²

Eis duas consequências rápidas dos últimos resultados, a primeira das quais põe a nova notação em prática:

*sth:ordinals:basic:
ordordered*

Corolário 67.18. Se $\exists \alpha \varphi(\alpha)$, então $\exists \alpha (\varphi(\alpha) \wedge \forall \beta (\varphi(\beta) \rightarrow \alpha \leq \beta))$. Além disso, para quaisquer ordinais α, β, γ , tanto $\alpha \notin \alpha$ como $\alpha \in \beta \in \gamma \rightarrow \alpha \in \gamma$.

Prova. Tal como **Proposição 67.2**. $\square$

Problema 67.3. Complete o raciocínio “exatamente análogo” na prova de **Teorema 67.17**.

*sth:ordinals:basic:
corordtransitiveord*

Corolário 67.19. A é um ordinal sse A é um conjunto transitivo de ordinais.

Prova. Da esquerda para a direita. Por **Lema 67.14**. Da direita para a esquerda. Se A é um conjunto transitivo de ordinais, então $\in$ bem ordena A por **Teorema 67.16** e **Teorema 67.17**. $\square$

²Poderíamos escrever $\alpha \subseteq \beta$; mas isso seria totalmente não convencional.

Agora, descrevemos [Teorema 67.16](#) e [Teorema 67.17](#) como dizendo que $\in$ bem ordena os ordinais. Contudo, temos de ser *muito cautelosos* com este tipo de afirmação, graças ao resultado seguinte:

Teorema 67.20 (Paradoxo de Burali-Forti). *Não existe conjunto de todos os ordinais*

[sth:ordinals:basic:
buraliforti](#)

Prova. Por redução ao absurdo, suponha O o conjunto de todos os ordinais. Se $\alpha \in \beta \in O$, então α é um ordinal, por [Lema 67.14](#), logo $\alpha \in O$. Logo O é transitivo, e portanto O é um ordinal por [Corolário 67.19](#). Daí $O \in O$, contradizendo [Corolário 67.18](#). $\square$

Este resultado deve o nome a [Burali-Forti](#). Mas foi Cantor em 1899—numa carta a Dedekind—quem viu pela primeira vez com clareza a *contradição* em supor que existe um conjunto de todos os ordinais. Como van Heijenoort explica:

O próprio Burali-Forti considerou a contradição como estabelecendo, por *reductio ad absurdum*, o resultado de que a ordenação natural dos ordinais é apenas uma ordenação parcial. ([Heijenoort, 1967](#), p. 105)

Deixando de lado o erro de Burali-Forti, podemos resumir o precedente como segue. Os ordinais são conjuntos que individualmente são bem ordenados por pertença, e coletivamente são bem ordenados por pertença (sem coletivamente constituírem um conjunto).

Para concluir, eis mais propriedades básicas dos ordinais que seguem de [Teorema 67.16](#) e [Teorema 67.17](#).

Proposição 67.21. *Toda a sucessão estritamente decendente de ordinais é finita.*

Prova. Toda a sucessão infinita estritamente decendente de ordinais $\alpha_0 > \alpha_1 > \alpha_2 > \dots$ não tem membro $<$ -mínimo, contradizendo [Teorema 67.16](#). $\square$

Proposição 67.22. $\alpha \subseteq \beta \vee \beta \subseteq \alpha$, para quaisquer ordinais α, β .

[sth:ordinals:basic:
ordinalsaresubsets](#)

Prova. Se $\alpha \in \beta$, então $\alpha \subseteq \beta$ pois β é transitivo. Analogamente, se $\beta \in \alpha$, então $\beta \subseteq \alpha$. E se $\alpha = \beta$, então $\alpha \subseteq \beta$ e $\beta \subseteq \alpha$. Logo, por [Teorema 67.17](#) terminamos. $\square$

Proposição 67.23. $\alpha = \beta$ sse $\alpha \cong \beta$, para quaisquer ordinais α, β .

[sth:ordinals:basic:
ordisoidentity](#)

Prova. Os ordinais são boas ordenações; logo isto é imediato da Tricotomia ([Teorema 67.17](#)) e de [Lema 67.9](#). $\square$

Problema 67.4. Prove que, se cada membro de X é um ordinal, então $\bigcup X$ é um ordinal.

[sth:ordinals:basic:
probunionordinalsordinal](#)

67.7. SUBSTITUIÇÃO

67.7 Substituição

sth:ordinals:replacement:
sec

Na [Seção 67.5](#), motivamos a introdução dos ordinais ao sugerir que os poderíamos tratar como tipos de ordem, isto é, como representantes canônicos de boas ordenações. Para que isso funcione, seria preciso provar que *toda a boa ordenação é isomorfa a algum ordinal*. Isso permitiria definir $\text{ord}(A, <)$ como o ordinal α tal que $\langle A, < \rangle \cong \alpha$.

Infelizmente, *não* podemos provar o resultado desejado apenas com os axiomas introduzidos até aqui. (Veremos por que em [Seção 69.2](#), mas por agora basta: não podemos.) Precisamos de uma ideia nova, e eis-a:

Esquema de Substituição (Esquema de Substituição). Para qualquer fórmula $\varphi(x, y)$, o seguinte é um axioma:

para qualquer A , se $(\forall x \in A) \exists! y \varphi(x, y)$, então $\{y : (\exists x \in A) \varphi(x, y)\}$ existe.

Como na Separação, isto é um esquema: produz infinitamente muitos axiomas, um para cada uma das infinitas φ distintas. E pode igualmente ser (e normalmente é) escrito assim:

Para qualquer fórmula $\varphi(x, y)$ que não contenha “ B ”, o seguinte é um axioma:

$$\forall A[(\forall x \in A) \exists! y \varphi(x, y) \rightarrow \exists B \forall y (y \in B \leftrightarrow (\exists x \in A) \varphi(x, y))]$$

À primeira vista, contudo, esta é uma fórmula bastante intrincada. A consequência imediata seguinte da Substituição dá provavelmente uma expressão *mais clara* à ideia intuitiva com que trabalhamos:³

Corolário 67.24. *Para qualquer termo $\tau(x)$, e qualquer conjunto A , este conjunto existe:*

$$\{\tau(x) : x \in A\} = \{y : (\exists x \in A) y = \tau(x)\}.$$

Prova. Como τ é um termo, $\forall x \exists! y \tau(x) = y$. *A fortiori*, $(\forall x \in A) \exists! y \tau(x) = y$. Logo $\{y : (\exists x \in A) \tau(x) = y\}$ existe pela Substituição. $\square$

Isto sugere que “Substituição” é um bom nome para o axioma: dado um conjunto A , pode formar um novo conjunto, $\{\tau(x) : x \in A\}$, substituindo cada membro de A pela sua imagem sob τ . De fato, seguindo a notação para a imagem de um conjunto sob uma função, poderíamos escrever $\tau[A]$ para $\{\tau(x) : x \in A\}$.

³Um termo é uma expressão que designa exatamente um objeto (em qualquer instânciação das suas variáveis livres). Por exemplo, “ $\emptyset$ ” é um termo que designa o conjunto vazio; “ $\{x\}$ ” é um termo que designa o singular de x (seja qual for x); “ $x \cup y$ ” é um termo que designa a união de x e de y (sejam quais forem).

O essencial, porém, é que τ é um *termo*. Não tem de ser (um nome de) uma *função*, no sentido de [Seção 3.3](#), isto é, um certo conjunto de pares ordenados. Afinal, se f é uma função (nesse sentido), então o conjunto $f[A] = \{f(x) : x \in A\}$ é apenas um subconjunto particular de $\text{Im}(f)$, e isso já está garantido a existir apenas com os axiomas de $\mathbf{Z}^-$.⁴ A Substituição, em contraste, é um acréscimo *poderoso* aos nossos axiomas, como veremos no [Capítulo 69](#).

[content/set-theory/ordinals/milestone.tex](#)

67.8 $\mathbf{ZF}^-$: um marco

A questão de como justificar o Esquema de Substituição (se é que o justificamos) não é linear. Como tal, a reservamos para o [Capítulo 69](#). No entanto, com a adição da Substituição, atingimos outro marco importante. Dispomos agora de todos os axiomas necessários para a teoria $\mathbf{ZF}^-$. Em detalhe:

[sth:ordinals:zfm:](#)
[sec](#)

Definição 67.25. A teoria $\mathbf{ZF}^-$ tem estes axiomas: Extensionalidade, União, Pares, Conjuntos das Partes, Infinito, e todas as instâncias dos esquemas de Separação e Substituição. Por outras palavras, $\mathbf{ZF}^-$ acrescenta a Substituição a $\mathbf{Z}^-$.

Isto designa a teoria de conjuntos de *Zermelo–Fraenkel* (*menos* algo que veremos mais adiante). Fraenkel recebe a honra, uma vez que lhe é atribuída a formulação da Substituição em [1922](#), embora a primeira formulação precisa seja devida a [Skolem \(1922\)](#).

[content/set-theory/ordinals/ordtype.tex](#)

67.9 Ordinais como Tipos de Ordem

Armados com a Substituição, e portanto trabalhando agora em $\mathbf{ZF}^-$, podemos finalmente provar o resultado a que vimos aspirando:

[sth:ordinals:ordtype:](#)
[sec](#)

Teorema 67.26. *Toda a boa ordenação é isomorfa a um único ordinal.*

[sth:ordinals:ordtype:](#)
[thmOrdinalRepresentation](#)

Prova. Seja $\langle A, < \rangle$ uma boa ordenação. Por [Proposição 67.23](#), é isomorfa no máximo a um ordinal. Logo, por redução ao absurdo, suponha que $\langle A, < \rangle$ não é isomorfa a *nenhum* ordinal. Começaremos por “tornar $\langle A, < \rangle$ o mais pequena possível”. Em detalhe: se algum segmento inicial próprio $\langle A_a, <_a \rangle$ não for isomorfo a nenhum ordinal, existe um $a \in A$ mínimo com essa propriedade; então seja $B = A_a$ e $< = <_a$. Caso contrário, seja $B = A$ e $< = <$.

Por definição, todo o segmento inicial próprio de B é isomorfo a algum ordinal, que é único como acima. Logo, pela Substituição, o conjunto seguinte existe, e é uma função:

$$f = \{\langle \beta, b \rangle : b \in B \text{ e } \beta \cong \langle B_b, <_b \rangle\}$$

⁴Baste considerar $\{y \in \bigcup f : (\exists x \in A)y = f(x)\}$.

67.10. ORDINAIS SUCESSORES E LIMITES

Para completar a redução ao absurdo, mostraremos que f é um isomorfismo $\alpha \rightarrow B$, para algum ordinal α .

É óbvio que $\text{Im}(f) = B$. E por **Lema 67.11**, f preserva a ordenação, isto é, $\gamma \in \beta$ sse $f(\gamma) < f(\beta)$. Para mostrar que $\text{dom}(f)$ é um ordinal, por **Corolário 67.19** basta mostrar que $\text{dom}(f)$ é transitivo. Fixe então $\beta \in \text{dom}(f)$, isto é, $\beta \cong \langle B_b, <_b \rangle$ para algum $b \in B$. Se $\gamma \in \beta$, então $\gamma \in \text{dom}(f)$ por **Lema 67.10**; generalizando, $\beta \subseteq \text{dom}(f)$. $\square$

Este resultado autoriza a definição seguinte, que queríamos apresentar desde **Seção 67.5**:

Definição 67.27. Se $\langle A, < \rangle$ é uma boa ordenação, então o seu tipo de ordem, $\text{ord}(A, <)$, é o único ordinal α tal que $\langle A, < \rangle \cong \alpha$.

Além disso, esta definição autoriza dois princípios convenientes:

*sth:ordinals:ordtype:
ordtypesworklikeyouwant*

Corolário 67.28. Sendo $\langle A, < \rangle$ e $\langle B, < \rangle$ boas ordenações:

$$\begin{aligned} \text{ord}(A, <) &= \text{ord}(B, <) \text{ sse } \langle A, < \rangle \cong \langle B, < \rangle \\ \text{ord}(A, <) &\in \text{ord}(B, <) \text{ sse } \langle A, < \rangle \cong \langle B_b, <_b \rangle \text{ para algum } b \in B \end{aligned}$$

Prova. A identidade vale por **Proposição 67.23**. Para provar a segunda afirmação, seja $\text{ord}(A, <) = \alpha$ e $\text{ord}(B, <) = \beta$, e seja $f: \beta \rightarrow \langle B, < \rangle$ o nosso isomorfismo. Então:

$$\begin{aligned} \alpha \in \beta \text{ sse } f|_\alpha: \alpha \rightarrow B_{f(\alpha)} \text{ é um isomorfismo} \\ \text{sse } \langle A, < \rangle \cong \langle B_{f(\alpha)}, <_{f(\alpha)} \rangle \\ \text{sse } \langle A, < \rangle \cong \langle B_b, <_b \rangle \text{ para algum } b \in B \end{aligned}$$

por **Proposição 67.7**, **Lema 67.10**, e **Corolário 67.15**. $\square$

content/set-theory/ordinals/opps.tex

67.10 Ordinais Sucessores e Limites

*sth:ordinals:opps:
sec*

Nos próximos capítulos, usaremos muito os ordinais. Por isso convém introduzir algumas noções simples.

Definição 67.29. Para qualquer ordinal α , o seu *sucessor* é $\alpha^+ = \alpha \cup \{\alpha\}$. Dizemos que α é um ordinal *sucessor* se $\beta^+ = \alpha$ para algum ordinal β . Dizemos que α é um ordinal *limite* sse α não é nem vazio nem um ordinal sucessor.

O resultado seguinte mostra que esta é a noção certa de *sucessor*:

Proposição 67.30. Para qualquer ordinal α :

1. $\alpha \in \alpha^+$;

2. α^+ é um ordinal;

3. não existe ordinal β tal que $\alpha \in \beta \in \alpha^+$.

Prova. Trivialmente, $\alpha \in \alpha \cup \{\alpha\} = \alpha^+$. Igualmente, α^+ é um conjunto transitivo de ordinais, e portanto um ordinal por [Corolário 67.19](#). E é impossível que $\alpha \in \beta \in \alpha^+$, pois então ou $\beta \in \alpha$ ou $\beta = \alpha$, contradizendo [Corolário 67.18](#). $\square$

Isto autoriza também uma variante de prova por indução transfinita:

Teorema 67.31 (Indução Transfinita Simples). *Seja $\varphi(x)$ uma fórmula tal que:* sth:ordinals:opps:
simpletransrecursion

1. $\varphi(\emptyset)$; e

2. para qualquer ordinal α , se $\varphi(\alpha)$ então $\varphi(\alpha^+)$; e

3. se α é um ordinal limite e $(\forall \beta \in \alpha) \varphi(\beta)$, então $\varphi(\alpha)$.

Então $\forall \alpha \varphi(\alpha)$.

Prova. Provamos a contrapositiva. Logo, suponha que existe algum ordinal para o qual vale $\neg \varphi$; seja γ o menor tal ordinal. Então ou $\gamma = \emptyset$, ou $\gamma = \alpha^+$ para algum α tal que $\varphi(\alpha)$; ou γ é um ordinal limite e $(\forall \beta \in \gamma) \varphi(\beta)$. $\square$

Um último bocado de notação será útil mais adiante:

Definição 67.32. Se X é um conjunto de ordinais, então $\text{lsub}(X) = \bigcup_{\alpha \in X} \alpha^+$. sth:ordinals:opps:
defsupstrict

Aqui, “lsub” abrevia “least strict upper bound” (majorante superior estrito mínimo).⁵ O resultado seguinte explica isto:

Proposição 67.33. *Se X é um conjunto de ordinais, $\text{lsub}(X)$ é o menor ordinal estritamente maior do que todo o ordinal em X .*

Prova. Seja $Y = \{\alpha^+ : \alpha \in X\}$, de modo que $\text{lsub}(X) = \bigcup Y$. Como os ordinais são transitivos e todo o membro de um ordinal é um ordinal, $\text{lsub}(X)$ é um conjunto transitivo de ordinais, e portanto um ordinal por [Corolário 67.19](#).

Se $\alpha \in X$, então $\alpha^+ \in Y$, logo $\alpha^+ \subseteq \bigcup Y = \text{lsub}(X)$, e portanto $\alpha \in \text{lsub}(X)$. Logo $\text{lsub}(X)$ é estritamente maior do que todo o ordinal em X .

Inversamente, se $\alpha \in \text{lsub}(X)$, então $\alpha \in \beta^+ \in Y$ para algum $\beta \in X$, de modo que $\alpha \leq \beta \in X$. Logo $\text{lsub}(X)$ é o majorante superior estrito *mínimo* de X . $\square$

⁵Alguns livros usam “sup(X)” para isto. Mas outros usam “sup(X)” para o majorante superior *não estrito* mínimo, isto é, simplesmente $\bigcup X$. Se X tem um maior elemento, α , estas noções divergem: o majorante superior estrito mínimo é α^+ , enquanto o majorante superior não estrito mínimo é apenas α .

Capítulo 68

Estágios e postos

content/set-theory/spine/idea.tex

68.1 Definindo os estágios como os V_α

sth:spine:valpha: No Capítulo 67, definimos boas ordenações e os ordinais (de von Neumann).
sec Neste capítulo, usaremos isto para caracterizar a própria hierarquia de conjuntos. Para tal, recordemos que na Seção 67.10 definimos a noção de ordinal sucessor e ordinal limite. Usamos estas ideias na definição seguinte:

sth:spine:valpha: Definição 68.1.
defValphas

$$\begin{aligned} V_\emptyset &= \emptyset \\ V_{\alpha+} &= \wp(V_\alpha) && \text{para qualquer ordinal } \alpha \\ V_\alpha &= \bigcup_{\gamma < \alpha} V_\gamma && \text{quando } \alpha \text{ é um ordinal limite} \end{aligned}$$

Será uma definição por *recursão transfinita* nos ordinais. A este respeito, convém comparar com definições recursivas de funções nos números naturais.¹ Como ao tratar dos números naturais, define-se um caso base e casos sucessor; mas ao tratar de ordinais, também precisamos de descrever o comportamento nos casos *limite*.

Esta definição dos V_α será um marco importante. Motivamos informalmente a nossa hierarquia de conjuntos como formando conjuntos por *estágios*. Os V_α são, em efeito, precisamente esses estágios. Importa, contudo, que se trata de uma caracterização *interna* dos estágios. Em vez de sugerir um possível *modelo* da teoria, teremos definido os estágios *dentro* da nossa teoria dos conjuntos.

content/set-theory/spine/recursion.tex

¹Cf. as definições de adição, multiplicação e exponenciação em Seção 6.2.

68.2 Teorema(s) de recursão transfinita

O primeiro passo é, contudo, confirmar que **Definição 68.1** é uma definição bem sucedida. De modo mais geral, precisamos de provar que qualquer tentativa de definir por recursão transfinita (transfinita) terá êxito. Esse é o objetivo deste texto.

sth:spine:recursion:
sec

Aviso: este material é delicado. A lição geral, no entanto, é bem simples: a indução transfinita mais o Esquema de Substituição garantem a legitimidade de (várias versões da) recursão transfinita.²

Definição 68.2. Seja $\tau(x)$ um termo; seja f uma função; seja α um ordinal. Dizemos que f é uma *aproximação de ordem α* para τ sse $\text{dom}(f) = \alpha$ e $(\forall \beta \in \alpha) f(\beta) = \tau(f \upharpoonright_\beta)$.

Lema 68.3 (Recursão limitada). *Para qualquer termo $\tau(x)$ e qualquer ordinal α , existe uma única aproximação de ordem α para τ .*

sth:spine:recursion:
transrecursionfun

Prova. Mostraremos que, para qualquer $\gamma \leq \alpha$, existe uma única aproximação de ordem γ .

Começamos por estabelecer a unicidade. Sejam g e h (respetivamente) aproximações de ordem γ e δ . Uma indução transfinita nos seus argumentos mostra que $g(\beta) = h(\beta)$ para qualquer $\beta \in \text{dom}(g) \cap \text{dom}(h) = \gamma \cap \delta = \min(\gamma, \delta)$. Logo as aproximações são únicas (se existem) e coincidem em todos os valores.

Para estabelecer a existência, usamos agora uma indução transfinita simples (**Teorema 67.31**) nos ordinais $\delta \leq \alpha$.

A função vazia é trivialmente uma aproximação de ordem $\emptyset$.

Se g é uma aproximação de ordem γ , então $g \cup \{\langle \gamma, \tau(g) \rangle\}$ é uma aproximação de ordem γ^+ .

Se γ é um ordinal limite e g_δ é uma aproximação de ordem δ para todo o $\delta < \gamma$, seja $g = \bigcup_{\delta \in \gamma} g_\delta$. Isto é uma função, pois as várias g_δ coincidem em todos os valores. E se $\delta \in \gamma$ então $g(\delta) = g_{\delta^+}(\delta) = \tau(g_\delta \upharpoonright_\delta) = \tau(g \upharpoonright_\delta)$.

Isto completa a prova por indução transfinita. $\square$

Se nos permitirmos definir um *termo* em vez de uma função, podemos retirar a cota α do resultado anterior. No enunciado e na prova do resultado seguinte, quando σ é um termo, definimos $\sigma \upharpoonright_\alpha = \{\langle \beta, \sigma(\beta) \rangle : \beta \in \alpha\}$.

Teorema 68.4 (Recursão geral). *Para qualquer termo $\tau(x)$, podemos definir explicitamente um termo $\sigma(x)$ tal que $\sigma(\alpha) = \tau(\sigma \upharpoonright_\alpha)$ para qualquer ordinal α .*

sth:spine:recursion:
transrecursionschema

²Lembrete: todas as fórmulas e termos podem ter parâmetros (salvo indicação em contrário).

68.2. TEOREMA(S) DE RECURSÃO TRANSFINITA

Prova. Para cada α , por [Lema 68.3](#) existe uma única aproximação de ordem α , f_α , para τ . Definimos $\sigma(\alpha)$ como $f_{\alpha^+}(\alpha)$. Ora:

$$\begin{aligned}\sigma(\alpha) &= f_{\alpha^+}(\alpha) \\ &= \tau(f_{\alpha^+} \upharpoonright_\alpha) \\ &= \tau(\{\langle \beta, f_{\alpha^+}(\beta) \rangle : \beta \in \alpha\}) \\ &= \tau(\{\langle \beta, f_{\beta^+}(\beta) \rangle : \beta \in \alpha\}) \\ &= \tau(\sigma \upharpoonright_\alpha)\end{aligned}$$

notando que $f_{\beta^+}(\beta) = f_{\alpha^+}(\beta)$ para todo o $\beta < \alpha$, como no [Lema 68.3](#). $\square$

Notemos que [Teorema 68.4](#) é um *esquema*. Crucialmente, não podemos esperar que σ defina uma função, isto é, um certo tipo de *conjunto*, pois então $\text{dom}(\sigma)$ seria o conjunto de todos os ordinais, contradizendo o paradoxo de Burali-Forti ([Teorema 67.20](#)).

Ainda falta mostrar, contudo, que [Teorema 68.4](#) justifica a nossa definição dos V_α . Isto pode não ser óbvio de imediato; mas tornar-se-á claro com uma última versão simples da recursão transfinita.

*sth:spine:recursion:
simplerecursionschema*

Teorema 68.5 (Recursão simples). *Para quaisquer termos $\tau(x)$ e $\theta(x)$ e qualquer conjunto A , podemos definir explicitamente um termo $\sigma(x)$ tal que:*

$$\begin{aligned}\sigma(\emptyset) &= A \\ \sigma(\alpha^+) &= \tau(\sigma(\alpha)) && \text{para qualquer ordinal } \alpha \\ \sigma(\alpha) &= \theta(\text{Im}(\sigma \upharpoonright_\alpha)) && \text{quando } \alpha \text{ é um ordinal limite}\end{aligned}$$

Prova. Começamos por definir um termo, $\xi(x)$, do seguinte modo:

$$\xi(x) = \begin{cases} A & \text{se } x \text{ não é uma função cujo} \\ & \text{domínio é um ordinal; caso contrário:} \\ \tau(x(\alpha)) & \text{se } \text{dom}(x) = \alpha^+ \\ \theta(\text{Im}(x)) & \text{se } \text{dom}(x) \text{ é um ordinal limite} \end{cases}$$

Por [Teorema 68.4](#), existe um termo $\sigma(x)$ tal que $\sigma(\alpha) = \xi(\sigma \upharpoonright_\alpha)$ para todo o ordinal α ; além disso, $\sigma \upharpoonright_\alpha$ é uma função com domínio α . Mostramos que σ tem as propriedades pretendidas, por indução transfinita simples ([Teorema 67.31](#)).

Primeiro, $\sigma(\emptyset) = \xi(\emptyset) = A$.

Segundo, $\sigma(\alpha^+) = \xi(\sigma \upharpoonright_{\alpha^+}) = \tau(\sigma \upharpoonright_{\alpha^+}(\alpha)) = \tau(\sigma(\alpha))$.

Por fim, $\sigma(\alpha) = \xi(\sigma \upharpoonright_\alpha) = \theta(\text{Im}(\sigma \upharpoonright_\alpha))$, quando α é limite. $\square$

Ora, para justificar [Definição 68.1](#), basta tomar $A = \emptyset$, $\tau(x) = \wp(x)$ e $\theta(x) = \bigcup x$. Por fim, isto justifica a definição dos V_α !

68.3 Propriedades básicas dos estágios

Para evidenciar a importância fundacional da definição dos V_α , apresentaremos alguns resultados básicos sobre eles. Começamos com uma definição:³

sth:spine:Valphabasic:
sec

Definição 68.6. O conjunto A é *potente* sse $\forall x((\exists y \in A)x \subseteq y \rightarrow x \in A)$.

Lema 68.7. Para cada ordinal α :

sth:spine:Valphabasic:
Valphabasicprops

1. Cada V_α é transitivo.

sth:spine:Valphabasic:
Valphatrans

2. Cada V_α é potente.

sth:spine:Valphabasic:
Valphapotent

3. Se $\gamma \in \alpha$, então $V_\gamma \in V_\alpha$ (e portanto também $V_\gamma \subseteq V_\alpha$, por (1))

sth:spine:Valphabasic:
Valphacum

Prova. Provamos isto por indução transfinita (simultânea). Para a indução, supomos que (1)–(3) valem para cada ordinal $\beta < \alpha$.

O caso $\alpha = \emptyset$ é trivial.

Suponhamos $\alpha = \beta^+$. Para mostrar (3), se $\gamma \in \alpha$ então $V_\gamma \subseteq V_\beta$ por hipótese, logo $V_\gamma \in \wp(V_\beta) = V_\alpha$. Para mostrar (2), suponhamos $A \subseteq B \in V_\alpha$, isto é, $A \subseteq B \subseteq V_\beta$; então $A \subseteq V_\beta$, logo $A \in V_\alpha$. Para mostrar (1), notemos que se $x \in A \in V_\alpha$ temos $A \subseteq V_\beta$, logo $x \in V_\beta$, logo $x \subseteq V_\beta$ pois V_β é transitivo por hipótese, e assim $x \in V_\alpha$.

Suponhamos que α é um ordinal limite. Para mostrar (3), se $\gamma \in \alpha$ então $\gamma \in \gamma^+ \in \alpha$, de modo que $V_\gamma \in V_{\gamma^+}$ por hipótese, logo $V_\gamma \in \bigcup_{\beta \in \alpha} V_\beta = V_\alpha$. Para mostrar (1) e (2), basta observar que uma união de conjuntos transitivos (respectivamente, potentes) é transitiva (respectivamente, potente). $\square$

Lema 68.8. Para cada ordinal α , $V_\alpha \notin V_\alpha$.

sth:spine:Valphabasic:
Valphanotref

Prova. Por indução transfinita. Evidentemente $V_\emptyset \notin V_\emptyset$.

Se $V_{\alpha^+} \in V_{\alpha^+} = \wp(V_\alpha)$, então $V_{\alpha^+} \subseteq V_\alpha$; e como $V_\alpha \in V_{\alpha^+}$ por Lema 68.7, temos $V_\alpha \in V_\alpha$. Reciprocamente: se $V_\alpha \notin V_\alpha$ então $V_{\alpha^+} \notin V_{\alpha^+}$.

Se α é limite e $V_\alpha \in V_\alpha = \bigcup_{\beta \in \alpha} V_\beta$, então $V_\alpha \in V_\beta$ para algum $\beta \in \alpha$; mas então também $V_\beta \in V_\alpha$, logo $V_\beta \in V_\beta$ por Lema 68.7 (duas vezes). Reciprocamente, se $V_\beta \notin V_\beta$ para todo o $\beta \in \alpha$, então $V_\alpha \notin V_\alpha$. $\square$

Corolário 68.9. Para quaisquer ordinais α, β : $\alpha \in \beta$ sse $V_\alpha \in V_\beta$

Prova. Lema 68.7 dá um sentido. Reciprocamente, suponhamos $V_\alpha \in V_\beta$. Então $\alpha \neq \beta$ por Lema 68.8; e $\beta \notin \alpha$, pois caso contrário teríamos $V_\beta \in V_\alpha$ e logo $V_\beta \in V_\beta$ por Lema 68.7 (duas vezes), contradizendo Lema 68.8. Logo $\alpha \in \beta$ por tricotomia. $\square$

³Não há terminologia padrão para «potente»; este é o nome usado por Button (2021).

68.4. FUNDAÇÃO

Tudo isto permite pensar em cada V_α como o α -ésimo estágio da hierarquia. Eis a razão.

Certamente os nossos V_α podem ser vistos como formados num processo *iterativo*, pois o uso de ordinais acompanha a noção de iteração. Além disso, se um estágio é formado antes do outro, isto é, $V_\beta \in V_\alpha$, isto é, $\beta \in \alpha$, então o nosso processo de formação é *cumulativo*, pois $V_\beta \subseteq V_\alpha$. Finalmente, estamos de fato a formar *todas* as coleções possíveis de conjuntos que estavam disponíveis em qualquer estágio anterior, pois qualquer estágio sucessor $V_{\alpha+}$ é o conjunto potência do seu predecessor V_α .

Em resumo: com $\mathbf{ZF}^-$, estamos *quase* a terminar de articular a nossa visão da hierarquia cumulativa e iterativa de conjuntos. (Embora, naturalmente, ainda precisemos de justificar o Esquema de Substituição.)

content/set-theory/spine/foundation.tex

68.4 Fundação

sth:spine:foundation:
sec

Estamos apenas *quase* concluídos—e não *totalmente* concluídos—porque nada em $\mathbf{ZF}^-$ garante que *todo* o conjunto está em algum V_α , isto é, que todo o conjunto é formado em algum estágio.

Há, contudo, um sentido (matemático) bastante direto em que não *importa* se existem conjuntos fora da hierarquia. (Se os houver, podemos simplesmente ignorá-los.) Mas motivamos o nosso *conceito* de conjunto com a ideia de que todo o conjunto é formado em algum estágio (ver *Estágios-são-chave* na [Seção 66.1](#)). Logo queremos excluir a possibilidade de conjuntos que caem fora da hierarquia. Por conseguinte, devemos acrescentar um novo axioma, que assegura que todo o conjunto surge em algum lugar da hierarquia.

Uma vez que os V_α são os nossos estágios, poderíamos simplesmente considerar acrescentar o seguinte como axioma:

Regularidade. $\forall A \exists \alpha A \subseteq V_\alpha$

Seria uma abordagem perfeitamente razoável. Contudo, por razões que se explicarão adiante, adotaremos antes um axioma alternativo:

Fundação (Fundação). $(\forall A \neq \emptyset)(\exists B \in A)A \cap B = \emptyset$.

Com algum trabalho, podemos mostrar (em $\mathbf{ZF}^-$) que a Fundação implica a Regularidade:

Definição 68.10. Para cada conjunto A , definimos:

$$\begin{aligned} \text{cl}_0(A) &= A, \\ \text{cl}_{n+1}(A) &= \bigcup \text{cl}_n(A), \\ \text{trcl}(A) &= \bigcup_{n < \omega} \text{cl}_n(A). \end{aligned}$$

Chamamos a $\text{trcl}(A)$ o *fecho transitivo* de A .

O nome «fecho transitivo» é adequado:

Proposição 68.11. $A \subseteq \text{trcl}(A)$ e $\text{trcl}(A)$ é um conjunto transitivo.

*sth:spine:foundation:
subsetoftrcl*

Prova. Evidentemente $A = \text{cl}_0(A) \subseteq \text{trcl}(A)$. E se $x \in b \in \text{trcl}(A)$, então $b \in \text{cl}_n(A)$ para algum n , logo $x \in \text{cl}_{n+1}(A) \subseteq \text{trcl}(A)$. $\square$

Lema 68.12. Se A é um conjunto transitivo, então existe algum α tal que $A \subseteq V_\alpha$.

*sth:spine:foundation:
lem:TransitiveWellFounded*

Prova. Recordando a definição de $\llbracket \text{sub}(X) \rrbracket$ em [Definição 67.32](#), definimos dois conjuntos:

$$\begin{aligned} D &= \{x \in A : \forall \delta \ x \not\subseteq V_\delta\} \\ \alpha &= \text{lsub}\{\delta : (\exists x \in A)(x \subseteq V_\delta \wedge (\forall \gamma \in \delta)x \not\subseteq V_\gamma)\} \end{aligned}$$

Suponhamos $D = \emptyset$. Se $x \in A$, então existe algum δ tal que $x \subseteq V_\delta$ e, pela boa ordenação dos ordinais, $(\forall \gamma \in \delta)x \not\subseteq V_\gamma$; logo $\delta \in \alpha$ e assim $x \in V_\alpha$ por [Lema 68.7](#). Logo $A \subseteq V_\alpha$, como pretendido.

Basta pois mostrar que $D = \emptyset$. Por redução ao absurdo, suponhamos o contrário. Pela Fundação, existe algum $B \in D \subseteq A$ tal que $D \cap B = \emptyset$. Se $x \in B$ então $x \in A$, pois A é transitivo, e como $x \notin D$, segue que $\exists \delta \ x \subseteq V_\delta$. Seja agora

$$\beta = \text{lsub}\{\delta : (\exists x \in B)(x \subseteq V_\delta \wedge (\forall \gamma < \delta)x \not\subseteq V_\gamma)\}.$$

Como antes, $B \subseteq V_\beta$, contradizendo a afirmação de que $B \in D$. $\square$

Teorema 68.13. A Regularidade vale.

*sth:spine:foundation:
zfentailsregularity*

Prova. Fixemos A ; agora $A \subseteq \text{trcl}(A)$ por [Proposição 68.11](#), que é transitivo. Logo existe algum α tal que $A \subseteq \text{trcl}(A) \subseteq V_\alpha$ por [Lema 68.12](#). $\square$

Estes resultados mostram que $\mathbf{ZF}^-$ prova a implicação *Fundação* $\Rightarrow$ *Regularidade*. Em [Proposição 68.22](#), mostraremos que $\mathbf{ZF}^-$ prova *Regularidade* $\Rightarrow$ *Fundação*. Como tal, a Fundação e a Regularidade são *equivalentes* (módulo $\mathbf{ZF}^-$). Mas isto significa que, dado $\mathbf{ZF}^-$, podemos justificar a Fundação notando que é equivalente à Regularidade. E podemos justificar a Regularidade de imediato com base em *Estágios-são-chave*.

68.5 $\mathbf{Z}$ e $\mathbf{ZF}$: um marco

sth:spine:zf:
sec Com a Fundação, atingimos outro marco importante. Consideramos as teorias $\mathbf{Z}^-$ e $\mathbf{ZF}^-$, que dissemos ser certas teorias «menos» alguma coisa. Essa coisa é a Fundação. Logo:

Definição 68.14. A teoria $\mathbf{Z}$ acrescenta a Fundação a $\mathbf{Z}^-$. Os seus axiomas são Extensionalidade, União, Pares, Conjuntos das Partes, Infinito, Fundação, e todas as instâncias do esquema de Separação.

A teoria $\mathbf{ZF}$ acrescenta a Fundação a $\mathbf{ZF}^-$. Por outras palavras, $\mathbf{ZF}$ acrescenta todas as instâncias de Substituição a $\mathbf{Z}$.

Ainda assim, uma pergunta pode ter surgido. Se a Regularidade é equivalente sobre $\mathbf{ZF}^-$ à Fundação, e a justificação da Regularidade é clara, por que dar voltas e tomar a Fundação como axioma básico, em vez da Regularidade?

Deixando de lado razões históricas (quem formulou o quê e quando), a razão essencial é que a Fundação pode ser apresentada sem recorrer à definição dos V_α . Essa definição dependeu de todo o trabalho na [Seção 68.2](#): foi preciso provar a recursão transfinita, para mostrar que era legítima. Mas a nossa prova da recursão transfinita usou o *Esquema de Substituição*. Logo, embora a Fundação e a Regularidade sejam equivalentes módulo $\mathbf{ZF}^-$, não são equivalentes módulo $\mathbf{Z}^-$.

De fato, a questão é mais grave do que esta observação simples sugere. Embora ultrapasse o âmbito deste livro, verifica-se que tanto $\mathbf{Z}^-$ como $\mathbf{Z}$ são demasiado fracas para definir os V_α . Logo, se se trabalha apenas em $\mathbf{Z}$, então a Regularidade (tal como a formulamos) nem sequer faz *sentido*. É por isso que o nosso axioma oficial é a Fundação, e não a Regularidade.

Daqui em diante, trabalharemos em $\mathbf{ZF}$ (salvo indicação em contrário), sem mais comentários.

content/set-theory/spine/rank.tex

68.6 Posto

sth:spine:rank:
sec Tendo definido os estágios como os V_α e sabendo que todo o conjunto é subconjunto de algum estágio, podemos definir o *posto* de um conjunto. Intuitivamente, o posto de A é o primeiro momento em que A é formado. Com mais precisão:

sth:spine:rank:
defnsetrank **Definição 68.15.** Para cada conjunto A , $\text{posto}(A)$ é o menor ordinal α tal que $A \subseteq V_\alpha$.

sth:spine:rank:
rankexist **Proposição 68.16.** $\text{posto}(A)$ *existe*, para qualquer A .

Prova. Fica como exercício. □

Problema 68.1. Prove [Proposição 68.16](#).

A boa ordenação dos postos permite provar alguns resultados importantes:

Proposição 68.17. *Para qualquer ordinal α , $V_\alpha = \{x : \text{posto}(x) \in \alpha\}$.*

*sth:spine:rank:
valphalowerrank*

Prova. Se $\text{posto}(x) \in \alpha$ então $x \subseteq V_{\text{posto}(x)} \in V_\alpha$, logo $x \in V_\alpha$ pois V_α é potente (invocando **Lema 68.7** várias vezes). Reciprocamente, se $x \in V_\alpha$ então $x \subseteq V_\alpha$, logo $\text{posto}(x) \leq \alpha$; agora uma indução transfinita simples mostra que $x \notin V_\alpha$. $\square$

Problema 68.2. Complete a indução transfinita simples mencionada na **Proposição 68.17**.

Proposição 68.18. *Se $B \in A$, então $\text{posto}(B) \in \text{posto}(A)$.*

*sth:spine:rank:
rankmemberslower*

Prova. $A \subseteq V_{\text{posto}(A)} = \{x : \text{posto}(x) \in \text{posto}(A)\}$ por **Proposição 68.17**. $\square$

Com este fato, podemos estabelecer um resultado que nos permite provar afirmações sobre *todos os conjuntos* por uma forma de indução:

Teorema 68.19 (Esquema de indução- $\in$). *Para qualquer fórmula φ :*

$$\forall A((\forall x \in A)\varphi(x) \rightarrow \varphi(A)) \rightarrow \forall A\varphi(A).$$

Prova. Provaremos a contrapositiva. Suponhamos, pois, $\neg \forall A\varphi(A)$. Pela indução transfinita (**Teorema 67.16**), existe algum não- φ de posto mínimo possível; isto é, algum A tal que $\neg \varphi(A)$ e $\forall x(\text{posto}(x) \in \text{posto}(A) \rightarrow \varphi(x))$. Ora, se $x \in A$ então $\text{posto}(x) \in \text{posto}(A)$, por **Proposição 68.18**, logo $\varphi(x)$; isto é, $(\forall x \in A)\varphi(x) \wedge \neg \varphi(A)$. $\square$

Eis uma forma informal de interpretar este resultado poderoso. Digamos que φ é *hereditária* sse sempre que todo o **membro** de um conjunto é φ , o próprio conjunto é φ . Então a indução- $\in$ diz o seguinte: se φ é hereditária, todo o conjunto é φ .

Para concluir a discussão dos postos (por agora), provaremos algumas afirmações que já antecipamos várias vezes.

Proposição 68.20. $\text{posto}(A) = \text{lsub}_{x \in A} \text{posto}(x)$.

*sth:spine:rank:
ranksupstrict*

Prova. Seja $\alpha = \text{lsub}_{x \in A} \text{posto}(x)$. Por **Proposição 68.18**, $\alpha \leq \text{posto}(A)$. Mas se $x \in A$ então $\text{posto}(x) \in \alpha$, logo $x \in V_\alpha$ por **Proposição 68.17**, e portanto $A \subseteq V_\alpha$, isto é, $\text{posto}(A) \leq \alpha$. Logo $\text{posto}(A) = \alpha$. $\square$

Corolário 68.21. *Para qualquer ordinal α , $\text{posto}(\alpha) = \alpha$.*

*sth:spine:rank:
ordsetrankalpha*

Prova. Suponhamos, por indução transfinita, que $\text{posto}(\beta) = \beta$ para todo o $\beta \in \alpha$. Ora, $\text{posto}(\alpha) = \text{lsub}_{\beta \in \alpha} \text{posto}(\beta) = \text{lsub}_{\beta \in \alpha} \beta = \alpha$ por **Proposição 68.20**. $\square$

Finalmente, eis uma prova rápida do resultado prometido no fim de **Seção 68.4**, a saber que **ZF**[−] prova a implicação *Regularidade* $\Rightarrow$ *Fundação*. (Notemos que a noção de «posto» e **Proposição 68.18** estão disponíveis para uso nesta prova pois—como indicamos no início deste texto—podem ser apresentadas usando **ZF**[−] + Regularidade.)

Proposição 68.22 (trabalhando em $\mathbf{ZF}^- + \text{Regularidade}$). *A Fundação vale.*

Prova. Fixemos $A \neq \emptyset$, e algum $B \in A$ de posto mínimo possível. Se $c \in B$ então $\text{posto}(c) \in \text{posto}(B)$ por **Proposição 68.18**, logo $c \notin A$ pela escolha de B . $\square$

Capítulo 69

Substituição

content/set-theory/replacement/introduction.tex

69.1 Introdução

sth:replacement:intro:
sec

A Substituição é o esquema axiomático que distingue $\mathbf{ZF}$ de $\mathbf{Z}$. Recorremos a ele ao longo de **Capítulos 67** a **68**. Neste capítulo, abordaremos finalmente a questão: a Substituição está justificada?

Para tornar a questão precisa, convém observar que a Substituição é realmente bastante *forte*. Teremos uma ideia de quão forte é ao longo deste capítulo (e de novo na **Seção 72.5**). Isto sugere que uma justificação é de fato necessária.

Discutiremos dois tipos de justificação. Em traços largos: uma justificação *extrínseca* é uma tentativa de justificar um axioma pelos seus frutos; uma justificação *intrínseca* é uma tentativa de justificar um axioma sugerindo que ele é vindicado pelos conceitos matemáticos em causa. Teremos uma ideia mais clara do que isto significa ao longo deste capítulo, mas não passa da ponta de um icebergue. Para mais pormenores, veja em particular **Maddy (1988a e 1988b)**.

content/set-theory/replacement/strength.tex

69.2 A Força da Substituição

sth:replacement:strength:
sec

Começamos com uma observação simples sobre a força da Substituição: a menos que ultrapassemos $\mathbf{Z}$, não podemos provar a existência de qualquer ordinal de von Neumann maior ou igual a $\omega + \omega$.

Eis um esboço do porquê. Trabalhando em $\mathbf{ZF}$, consideremos o conjunto $V_{\omega+\omega}$. Este conjunto serve como domínio de um *modelo* para $\mathbf{Z}$. Para ver isto, introduzimos alguma notação para a *relativização* de uma fórmula:

Definição 69.1. Para qualquer conjunto M , e qualquer fórmula φ , seja φ^M a fórmula que resulta de restringir todos os quantificadores de φ a M . Isto é, substituir “ $\exists x$ ” por “ $(\exists x \in M)$ ”, e substituir “ $\forall x$ ” por “ $(\forall x \in M)$ ”.

sth:replacement:strength:
formularelativization

Pode mostrar-se que, para todo o axioma φ de $\mathbf{Z}$, tem-se $\mathbf{ZF} \vdash \varphi^{V_{\omega+\omega}}$. Mas $\omega + \omega$ não está em $V_{\omega+\omega}$, por **Corolário 68.21**. Logo $\mathbf{Z}$ é consistente com a não existência de $\omega + \omega$.

Foi por isso que dissemos, na **Seção 67.7**, que **Teorema 67.26** não pode ser provado sem Substituição. Pois é fácil, dentro de $\mathbf{Z}$, definir uma boa ordenação explícita que intuitivamente *deveria* ter tipo de ordem $\omega + \omega$. De fato, demos um exemplo informal disto em **Seção 67.2**, quando apresentamos a ordenação nos números naturais dada por:

$$\begin{aligned} n < m \text{ sse ou } n < m \text{ e } m - n \text{ é par,} \\ \text{ou } n \text{ é par e } m \text{ é ímpar.} \end{aligned}$$

Mas se $\omega + \omega$ não existe, esta boa ordenação não é isomorfa a nenhum ordinal. Logo $\mathbf{Z}$ não prova **Teorema 67.26**.

Invertendo o prisma: a Substituição permite provar a existência de $\omega + \omega$, e portanto tem de permitir provar a existência de $V_{\omega+\omega}$. E não só. Para *qualquer* boa ordenação que possamos definir, **Teorema 67.26** nos diz que existe algum α isomorfo com essa boa ordenação, e portanto que V_α existe. De forma direta, então, a Substituição garante que a hierarquia de conjuntos tem de ser *muito alta*.

Nas próximas partes do capítulo, e de novo em **Seção 72.5**, teremos uma ideia mais clara de quão *alta* a Substituição força a hierarquia a ser. O ponto simples, por agora, é que a Substituição *precisa* mesmo de justificação!

content/set-theory/replacement/extrinsic.tex

69.3 Considerações Extrínsecas sobre a Substituição

Começamos por considerar uma tentativa *extrínseca* de justificar a Substituição. Boolos sugere uma, como segue.

sth:replacement:extrinsic:
sec

[...] a razão para adotar os axiomas da substituição é bastante simples: têm muitas consequências desejáveis e (aparentemente) nenhuma indesejável. Para além de teoremas sobre a concepção iterativa, as consequências incluem uma teoria satisfatória, embora não ideal, dos números infinitos, e um resultado altamente desejável

69.3. CONSIDERAÇÕES EXTRÍNSECAS

que justifica definições indutivas em relações bem fundadas. (Boolos, 1971, 229)

A ideia central de Boolos é que devemos justificar a Substituição pelos seus frutos. E os frutos específicos que menciona são os que temos discutido nos últimos capítulos. A Substituição permitiu-nos provar que os ordinais de von Neumann eram excelentes sucedâneos da ideia de um tipo de boa ordenação (isto é a nossa “teoria satisfatória, embora não ideal, dos números infinitos”). A Substituição permitiu-nos também definir os V_α , estabelecer a noção de posto e provar a indução- $\in$ (isto corresponde aos nossos “teoremas sobre a concepção iterativa”). Por fim, a Substituição permite provar o teorema de recursão transfinita (isto são as “definições indutivas em relações bem fundadas”).

São, de fato, consequências desejáveis. Mas estas consequências desejáveis bastam para *justificar* a Substituição? *Não*. Ou pelo menos, não de forma direta.

Eis um problema simples. Embora tenhamos enunciado algumas consequências desejáveis da Substituição, muitas delas poderiam ter sido obtidas por outras vias. Isto não é tão conhecido como devia, pelo que convém explicar a situação.

Existe uma teoria simples de conjuntos, a Teoria dos Níveis, ou **LT** abreviadamente.¹ Os axiomas de **LT** são apenas Extensionalidade, Separação e a afirmação de que todo o conjunto é subconjunto de algum *nível*, onde “nível” é definido de modo astuto para que os níveis se comportem como os nossos amigos, os V_α . Logo **ZF** prova **LT**; mas **LT** é *muito* mais fraca que **ZF**. De fato, **LT** não dá Pares, Conjuntos potencia, Infinito nem Substituição. Seja **Zr** o resultado de acrescentar Infinito e Conjuntos potencia a **LT**; isto fornece também Pares, logo **Zr** é pelo menos tão forte quanto **Z**. Mas, de fato, **Zr** é estritamente mais forte que **Z**, pois acrescenta a afirmação de que todo o conjunto tem posto (daí a sugestão de o chamar **Zr**). De fato, **Zr** fornece: uma teoria perfeitamente satisfatória dos ordinais; resultados que estratificam a hierarquia em estágios bem ordenados; uma prova da indução- $\in$; e uma *versão* da recursão transfinita.

Em resumo: embora Boolos não soubesse disto, todas as consequências desejáveis que menciona poderiam ter sido obtidas *sem* Substituição; bastaria usar **Zr** em vez de **Z**.

(Diante disto, por que seguimos a rota convencional, ensinando-lhe **ZF** em vez de **LT** e **Zr**? Há duas razões. Primeiro: por razões puramente históricas, começar por **LT** é pouco standard; queríamos dotá-lo de ferramentas para ler discussões mais standard de teoria de conjuntos. Segundo: quando estiver preparado para apreciar **LT** e **Zr**, pode simplesmente ler Potter 2004 e Button 2021.)

Naturalmente, como **Zr** é estritamente mais fraca que **ZF**, há resultados que **ZF** prova e que **Zr** deixa em aberto. Logo alguém poderia tentar justificar a Substituição em bases extrínsecas apontando para um desses resultados. Mas,

¹As primeiras versões de **LT** são apresentadas por Montague (1965) e Scott (1974); foi simplificada e recebeu tratamento de livro por Potter (2004); e Button (2021) simplificou recentemente ainda mais **LT**.

uma vez que saiba usar **Zr**, é bastante difícil encontrar muitos exemplos de coisas que (a) ficam decididas pela Substituição mas não de outro modo, e (b) são intuitivamente verdadeiras. (Para mais sobre isto, veja [Potter 2004](#), §13.2.)

A conclusão é esta. Para fornecer uma justificação extrínseca convincente da Substituição, seria preciso encontrar um resultado que *não* possa ser obtido sem Substituição. E isso não é empresa fácil.

Consideremos um problema adicional que surge para qualquer tentativa de oferecer uma justificação puramente extrínseca da Substituição. (Este problema é talvez mais fundamental que o primeiro.) Boolos não se limita a observar que a Substituição tem muitas consequências desejáveis. Afirmar também que a Substituição “(aparentemente) não tem” consequências indesejáveis. Mas esta ressalva entre parênteses, “aparentemente”, é seguramente absolutamente crucial.

Relembre como chegamos aqui: a Compreensão ingênua encontrou inconsistência, e respondemos a essa inconsistência abraçando a concepção cumulativa-iterativa de conjunto. Esta concepção vem equipada com uma narrativa que, esperamos, nos assegura a sua consistência. Mas se não pudermos justificar a Substituição dentro dessa narrativa, então não temos (ainda) razão para crer que **ZF** é consistente. Ou melhor: não temos razão para crer que **ZF** é consistente, para além do fato (talvez meramente contingente) de que ninguém descobriu ainda uma contradição. Exatamente nesse sentido, o comentário de Boolos parece reduzir-se a isto: “(aparentemente) **ZF** é consistente”. Deveríamos exigir maior garantia de consistência do que isto.

Esta questão afecta qualquer tentativa *puremente* extrínseca de justificar a Substituição, isto é, qualquer justificação formulada unicamente em termos das consequências (conhecidas) de **ZF**. Como tal, convém procurar uma justificação *intrínseca* da Substituição, isto é, uma justificação que sugira que a narrativa que contamos sobre conjuntos de algum modo “já” nos compromete com a Substituição.

[content/set-theory/replacement/limofsize.tex](#)

69.4 Limitação de Tamanho

Talvez a tentativa mais comum de oferecer uma justificação “intrínseca” da Substituição passe pela seguinte noção:

[sth:replacement:limofsize:sec](#)

Limitação-de-tamanho. Quaisquer coisas formam um conjunto, desde que não haja demasiadas.

Este princípio vindicará de imediato a Substituição. Afinal, qualquer conjunto formado por Substituição não pode ser maior do que o conjunto a partir do qual foi formado. Enunciado com precisão: suponha que forma um conjunto $\tau[A] = \{\tau(x) : x \in A\}$ usando Substituição; então $\tau[A] \preceq A$; logo, se os [membros](#) de A não eram demasiado numerosos para formar um conjunto, as suas imagens não são demasiado numerosas para formar $\tau[A]$.

69.5. SUBSTITUIÇÃO E “INFINITO ABSOLUTO”

A dificuldade óbvia em invocar *Limitação-de-tamanho* para justificar a Substituição é que *ainda* não estabelecemos qualquer princípio como *Limitação-de-tamanho*. Além disso, quando contamos a nossa história sobre a concepção cumulativa-iterativa de conjunto em **Capítulos 65 a 66**, nada *sugeriu* a direção de *Limitação-de-tamanho*. Isto é, de fato, precisamente a razão pela qual Boolos escreveu num dado momento: “Talvez se possa concluir que há pelo menos dois pensamentos ‘por detrás’ da teoria de conjuntos” (1989, p. 19). Por um lado, as ideias em torno da concepção cumulativa-iterativa de conjunto destinam-se a vindicar **Z**. Por outro, *Limitação-de-tamanho* visa vindicar a Substituição.

Mas a questão não é apenas termos estado até aqui *em silêncio* sobre *Limitação-de-tamanho*. Antes, a questão é que *Limitação-de-tamanho* (tal como acabamos de o formular) parece assentar mal com a noção cumulativa-iterativa de conjunto. Afinal, não menciona nada sobre a ideia de conjuntos formados em *estágios*.

Isto não surpreende muito, dada a história destes “dois pensamentos” (isto é, a concepção cumulativa-iterativa de conjunto, e *Limitação-de-tamanho*). Estes “dois pensamentos” acabam por ser dois projetos bastante diferentes para bloquear os paradoxos da teoria de conjuntos. A noção cumulativa-iterativa de conjunto bloqueia o paradoxo de Russell dizendo, em traços largos: *nunca deveríamos ter esperado que existisse um conjunto de Russell, porque não seria “formado” em nenhum estágio*. Por contraste, *Limitação-de-tamanho* visa excluir o conjunto de Russell, dizendo, em traços largos: *nunca deveríamos ter esperado que existisse um conjunto de Russell, porque seria demasiado grande*.

Posto assim, sejamos diretos: considerada como resposta aos paradoxos, *Limitação-de-tamanho* carece de *muito* mais justificação. Considere, por exemplo, esta versão do paradoxo de Russell: *nenhum pug fareja exatamente os pugs que não se cheiram a si próprios* (ver **Seção 65.2**). Se perguntar “por que não há tal pug?”, não é boa resposta dizer-lhe que tal pug teria de cheirar demasiados pugs. Logo por que seria boa explicação intuitiva, da não existência de um conjunto de Russell, que teria de ser “demasiado grande” para existir?

Em suma, é perdoável estar um pouco perplexo quanto à motivação “intuitiva” para *Limitação-de-tamanho*.

content/set-theory/replacement/absinf.tex

69.5 Substituição e “Infinito Absoluto”

sth:replacement:absinf:
sec

Deixamos agora *Limitação-de-tamanho* para trás e exploramos uma outra família de tentativas (intrínsecas) de justificar a Substituição, que levam a sério a ideia de conjuntos formados em estágios.

Quando delineamos pela primeira vez o processo iterativo, oferecemos alguns princípios que explicam o que acontece em cada estágio. Foram *Estágios-são-chave*, *Estágios-são-ordenados*, e *Estágios-acumulam*. Mais tarde, acrescentamos alguns princípios que nos dizem algo sobre o número de estágios:

Estágios-prosseguem disse-nos que o processo de formação de conjuntos nunca termina, e *Estágios-atingem-o-infinito* disse-nos que o processo passa por um estágio infinito-ésimo.

É razoável sugerir que estes dois últimos princípios decorrem de algum princípio mais amplo, como:

Estágios-são-inesgotáveis. Há absolutamente infinitamente muitos estágios; a hierarquia é tão alta quanto possível.

Obviamente isto é um princípio informal. Mas mesmo que não seja imediatamente *implicado* pela concepção cumulativa-iterativa de conjunto, certamente parece *consonante* com ela. Pelo menos, e ao contrário de *Limitação-de-tamanho*, retém a ideia de que os conjuntos são formados estágio a estágio.

A esperança, agora, é alavancar *Estágios-são-inesgotáveis* numa justificação da Substituição. Vejamos como isso poderia fazer-se.

Na [Seção 67.2](#), vimos que é fácil construir uma boa ordenação que (moralmente) deveria ser isomorfa a $\omega + \omega$. Dito de outro modo, podemos facilmente imaginar um processo iterativo estágio a estágio, cujo tipo de ordem (moralmente) é $\omega + \omega$. Como tal, se aceitamos *Estágios-são-inesgotáveis*, então deveríamos certamente aceitar que há pelo menos um estágio $\omega + \omega$ -ésimo da hierarquia, isto é, $V_{\omega+\omega}$, pois a hierarquia seguramente *poderia* continuar até aqui.

Este pensamento generaliza-se como segue: para qualquer boa ordenação, o processo de construir a hierarquia iterativa deveria correr pelo menos tão longe quanto essa boa ordenação. E poderíamos garantir isto tratando [Teorema 67.26](#) como um *axioma*. Isto diria-nos que toda a boa ordenação é isomorfa a um ordinal de von Neumann. Como cada ordinal de von Neumann será igual ao seu próprio posto, [Teorema 67.26](#) dir-nos-á então que, sempre que possamos descrever uma boa ordenação na nossa teoria de conjuntos, o processo iterativo de formação de conjuntos tem de ultrapassar essa boa ordenação.

Esta ideia parece certamente um corolário de *Estágios-são-inesgotáveis*. Infelizmente, se o nosso objetivo é extrair a Substituição desta ideia, enfrentamos um obstáculo simples e técnico: a Substituição é estritamente mais forte que [Teorema 67.26](#). (Esta observação é feita por [Potter \(2004, §13.2\)](#); prová-lo-emos na [Seção 69.8](#).)

A conclusão é que, se vamos entender *Estágios-são-inesgotáveis* de modo a obter a Substituição, então não pode *meramente* dizer que a hierarquia ultrapassa qualquer boa ordenação. Tem de fazer uma afirmação mais forte. Com este fim, [Shoenfield \(1977\)](#) propôs um reforço muito natural da ideia, como segue: a hierarquia não é *cofinal* com nenhum conjunto.² Com um pouco mais de pormenor: se τ é um mapeamento que envia conjuntos para estágios da hierarquia, a imagem de qualquer conjunto A sob τ não esgota a hierarquia. Dito de outro modo (esquemáticamente):

²Gödel parece ter proposto um pensamento semelhante; ver [Potter \(2004, p. 223\)](#). Para discussão de Gödel e [Shoenfield](#), ver [Incurvati \(2020, 90–5\)](#).

69.6. SUBSTITUIÇÃO E REFLEXÃO

Estágios-são-supercofinais. Se A é um conjunto e $\tau(x)$ é um estágio para todo o $x \in A$, então existe um estágio que vem depois de cada $\tau(x)$ para $x \in A$.

É óbvio que **ZF** prova uma versão adequadamente formalizada de *Estágios-são-supercofinais*. Reciprocamente, podemos argumentar informalmente que *Estágios-são-supercofinais* justifica a Substituição.³ De fato, suponha $(\forall x \in A)\exists!y\varphi(x, y)$. Então, para cada $x \in A$, seja $\sigma(x)$ o y tal que $\varphi(x, y)$, e seja $\tau(x)$ o estágio em que $\sigma(x)$ é formado pela primeira vez. Por *Estágios-são-supercofinais*, existe um estágio V tal que $(\forall x \in A)\tau(x) \in V$. Ora, como cada $\tau(x) \in V$ e $\sigma(x) \subseteq \tau(x)$, por Separação podemos obter $\{y \in V : (\exists x \in A)\sigma(x) = y\} = \{y : (\exists x \in A)\varphi(x, y)\}$.

Problema 69.1. Apresente uma formalização de *Estágios-são-supercofinais* em **ZF**.

Logo *Estágios-são-supercofinais* vindica a Substituição. E é pelo menos plausível que *Estágios-são-inesgotáveis* vindique *Estágios-são-supercofinais*. De fato, suponha que *Estágios-são-supercofinais* falha. Logo a hierarquia é cofinal com algum conjunto A , isto é, temos uma aplicação τ tal que para qualquer estágio S existe algum $x \in A$ tal que $S \in \tau(x)$. Nesse caso, temos de fato uma forma de apreender o suposto “infinito absoluto” da hierarquia: ele é *esgotado* pela imagem de τ aplicada a A . E isso compromete a ideia de que a hierarquia é “absolutamente infinita”. Contrapondo: *Estágios-são-inesgotáveis* implica *Estágios-são-supercofinais*, que por sua vez justifica a Substituição.

Isto representa uma tentativa genuinamente promissora de fornecer uma justificação intrínseca da Substituição. Mas se funciona ultimamente ou não, teremos de deixar para si decidir.

content/set-theory/replacement/ref.tex

69.6 Substituição e Reflexão

sth:replacement:ref:
sec

A nossa última tentativa de justificar a Substituição, via *Estágios-são-inesgotáveis*, começa com um resultado profundo e elegante:⁴

sth:replacement:ref:
reflectionschema

Teorema 69.2 (Esquema de reflexão). Para qualquer fórmula φ :

$$\forall\alpha\exists\beta > \alpha(\forall x_1 \dots, x_n \in V_\beta)(\varphi(x_1, \dots, x_n) \leftrightarrow \varphi^{V_\beta}(x_1, \dots, x_n))$$

³Seria mais difícil provar a Substituição usando alguma formalização de *Estágios-são-supercofinais*, pois **Z** por si só não é forte o suficiente para definir os estágios, logo não é claro como se formalizaria *Estágios-são-supercofinais*. Uma opção, contudo, é trabalhar nalguma extensão de **LT**, como discutido na Seção 69.3.

⁴Lembrete: todas as fórmulas podem ter parâmetros (salvo indicação explícita em contrário).

Como na [Definição 69.1](#), φ^{V_β} é o resultado de restringir todos os quantificadores de φ ao conjunto V_β . Assim, intuitivamente, a Reflexão diz o seguinte: se φ é verdadeira em toda a hierarquia, então φ é verdadeira em arbitrariamente muitos *segmentos iniciais* da hierarquia.

[Montague \(1961\)](#) e [Lévy \(1960\)](#) mostraram que (formulações adequadas da) Substituição e da Reflexão são equivalentes, módulo **Z**, de modo que acrescentar qualquer uma delas dá **ZF**. (Provamos estes resultados na [Seção 69.7](#).) Dada esta equivalência, poder-se-á esperar justificar a Reflexão e a Substituição via *Estágios-são-inesgotáveis* do seguinte modo: dado *Estágios-são-inesgotáveis*, a hierarquia deverá ser muito, muito alta; tão alta, de fato, que nada do que podemos dizer sobre ela é suficiente para limitar a sua altura. E podemos entender isto como a ideia de que, se qualquer sentença φ é verdadeira em toda a hierarquia, então é verdadeira em arbitrariamente muitos segmentos iniciais da hierarquia. E isso é precisamente a Reflexão.

De novo, isto parece uma tentativa genuinamente promissora de fornecer uma justificação intrínseca da Substituição. Mas há demasiado a dizer sobre o assunto aqui. Cabe-lhe agora decidir por si se tem êxito.⁵

[content/set-theory/replacement/refproofs.tex](#)

69.7 Apêndice: Resultados sobre Substituição

A seguir, provaremos a Reflexão dentro de **ZF**. Provaremos também em que sentido a Reflexão é equivalente à Substituição. E provaremos uma consequência interessante de tudo isto, relativa à força da Reflexão/Substituição. *Aviso: isto é, de longe, a parte mais avançada de matemática deste manual.*

[sth:replacement:refproofs:sec](#)

Começamos com um lema que, por brevidade, recorre ao expediente notacional do *sobrelinhado* para lidar com seqüências de variáveis ou objetos. Assim: “ $\bar{a}_k$ ” abrevia “ $a_{k_1}, \dots, a_{k_n}$ ”, onde n é determinado pelo contexto.

Lema 69.3. *Para cada $1 \leq i \leq k$, seja $\varphi_i(\bar{v}_i, x)$ uma fórmula. Então, para cada α existe algum $\beta > \alpha$ tal que, para quaisquer $\bar{a}_1, \dots, \bar{a}_k \in V_\beta$ e cada $1 \leq i \leq k$:*

[sth:replacement:refproofs:lemreflection](#)

$$\exists x \varphi_i(\bar{a}_i, x) \rightarrow (\exists x \in V_\beta) \varphi_i(\bar{a}_i, x)$$

Prova. Definimos um termo μ como segue: $\mu(\bar{a}_1, \dots, \bar{a}_k)$ é o estágio mínimo, V , que satisfaz todos os seguintes condicionais, para $1 \leq i \leq k$:

$$\exists x \varphi_i(\bar{a}_i, x) \rightarrow (\exists x \in V) \varphi_i(\bar{a}_i, x)$$

⁵Poderá, no entanto, querer continuar lendo [Incurvati \(2020, 95–100\)](#).

69.7. APÊNDICE: RESULTADOS SOBRE SUBSTITUIÇÃO

É fácil verificar que $\mu(\bar{a}_1, \dots, \bar{a}_k)$ existe para todos $\bar{a}_1, \dots, \bar{a}_k$. Ora, usando a Substituição e o nosso teorema de recursão, definimos:

$$\begin{aligned} S_0 &= V_{\alpha+1} \\ S_{n+1} &= S_n \cup \bigcup \{ \mu(\bar{a}_1, \dots, \bar{a}_k) : \bar{a}_1, \dots, \bar{a}_k \in S_n \} \\ S &= \bigcup_{m < \omega} S_m. \end{aligned}$$

Cada S_n , e portanto o próprio S , é um estágio depois de V_α . Fixemos $\bar{a}_1, \dots, \bar{a}_k \in S$; logo existe algum $n < \omega$ tal que $\bar{a}_1, \dots, \bar{a}_k \in S_n$. Fixemos algum $1 \leq i \leq k$, e suponhamos que $\exists x \varphi_i(\bar{a}_i, x)$. Então $(\exists x \in \mu(\bar{a}_1, \dots, \bar{a}_k)) \varphi_i(\bar{a}_i, x)$ por construção, logo $(\exists x \in S_{n+1}) \varphi_i(\bar{a}_i, x)$ e portanto $(\exists x \in S) \varphi_i(\bar{a}_i, x)$. Logo S é o nosso V_β . $\square$

Podemos agora provar **Teorema 69.2** de forma bastante direta:

Prova. Fixemos α . Sem perda de generalidade, podemos assumir que os únicos conectivos de φ são $\exists, \neg$ e $\wedge$ (pois estes são expressivamente adequados). Sejam $\psi_1, \dots, \psi_k$ uma enumeração de cada uma das subfórmulas de φ por ordem de complexidade, de modo que $\psi_k = \varphi$. Por **Lema 69.3**, existe um $\beta > \alpha$ tal que, para qualquer $\bar{a}_i \in V_\beta$ e cada $1 \leq i \leq k$:

$$\exists x \psi_i(\bar{a}_i, x) \rightarrow (\exists x \in V_\beta) \psi_i(\bar{a}_i, x) \quad (*)$$

Por indução na complexidade de ψ_i , mostraremos que $\psi_i(\bar{a}_i) \leftrightarrow \psi_i^{V_\beta}(\bar{a}_i)$, para qualquer $\bar{a}_i \in V_\beta$. Se ψ_i é atômica, isto é trivial. A bicondicional estabelece também que, quando ψ_i é uma negação ou conjunção de subfórmulas que satisfazem esta propriedade, a própria ψ_i satisfaz esta propriedade. Logo o único caso interessante é o da quantificação. Fixemos $\bar{a}_i \in V_\beta$; então:

$$\begin{aligned} (\exists x \psi_i(\bar{a}_i, x))^{V_\beta} &\text{ sse } (\exists x \in V_\beta) \psi_i^{V_\beta}(\bar{a}_i, x) && \text{por definição} \\ &\text{ sse } (\exists x \in V_\beta) \psi_i(\bar{a}_i, x) && \text{por hipótese} \\ &\text{ sse } \exists x \psi_i(\bar{a}_i, x) && \text{por } (*) \end{aligned}$$

Isto completa a indução; o resultado segue pois $\psi_k = \varphi$. $\square$

Provamos a Reflexão em **ZF**. A nossa prova seguiu essencialmente **Montague (1961)**. Queremos agora provar em **Z** que a Reflexão implica a Substituição. A prova segue **Lévy (1960)**, com uma simplificação.

Como trabalhamos em **Z**, não podemos apresentar a Reflexão exatamente na forma dada acima. Afinal, formulamos a Reflexão usando a notação “ V_α ”, e isso não pode ser definido em **Z** (ver **Seção 68.5**). Em vez disso, ofereceremos uma formulação aparentemente mais fraca da Substituição, como segue:

Reflexão fraca. Para qualquer fórmula φ , existe um conjunto transitivo S tal que $0, 1$ e quaisquer parâmetros de φ são **membros** de S , e $(\forall \bar{x} \in S)(\varphi \leftrightarrow \varphi^S)$.

Para usar isto na prova da Substituição, seguiremos primeiro Lévy (1960, primeira parte do Teorema 2) e mostraremos que podemos “refletir” duas fórmulas de uma vez:

Lema 69.4 (em $\mathbf{Z} + \text{Reflexão fraca}$). *Para quaisquer fórmulas ψ, χ , existe um conjunto transitivo S tal que 0 e 1 (e quaisquer parâmetros das fórmulas) são **membros** de S , e $(\forall \bar{x} \in S)((\psi \leftrightarrow \psi^S) \wedge (\chi \leftrightarrow \chi^S))$.* sth:replacement:refproofs:
lem:reflect

Prova. Seja φ a fórmula $(z = 0 \wedge \psi) \vee (z = 1 \wedge \chi)$.

Aqui usamos uma abreviatura; devemos explicitar “ $z = 0$ ” como “ $\forall t t \notin z$ ” e “ $z = 1$ ” como “ $\forall s(s \in z \leftrightarrow \forall t t \notin s)$ ”. Mas como $0, 1 \in S$ e S é transitivo, estas fórmulas são *absolutas* para S ; isto é, aplicam-se ao mesmo objeto quer restrinjamos os seus quantificadores a S .⁶

Pela Reflexão fraca, temos algum S adequado tal que:

$$\begin{aligned} & (\forall z, \bar{x} \in S)(\varphi \leftrightarrow \varphi^S) \\ \text{isto é } & (\forall z, \bar{x} \in S)((z = 0 \wedge \psi) \vee (z = 1 \wedge \chi)) \leftrightarrow \\ & ((z = 0 \wedge \psi) \vee (z = 1 \wedge \chi))^S \\ \text{isto é } & (\forall z, \bar{x} \in S)((z = 0 \wedge \psi) \vee (z = 1 \wedge \chi)) \leftrightarrow \\ & ((z = 0 \wedge \psi^S) \vee (z = 1 \wedge \chi^S)) \\ \text{isto é } & (\forall \bar{x} \in S)((\psi \leftrightarrow \psi^S) \wedge (\chi \leftrightarrow \chi^S)) \end{aligned}$$

A segunda afirmação implica a terceira porque “ $z = 0$ ” e “ $z = 1$ ” são absolutas para S ; a quarta segue pois $0 \neq 1$. □

Podemos agora obter a Substituição, seguindo e simplificando Lévy (1960, Teorema 6):

Teorema 69.5 (em $\mathbf{Z} + \text{Reflexão fraca}$). *Para qualquer fórmula $\varphi(v, w)$, e qualquer A , se $(\forall x \in A)\exists!y \varphi(x, y)$, então $\{y : (\exists x \in A)\varphi(x, y)\}$ existe.*

Prova. Fixemos A tal que $(\forall x \in A)\exists!y \varphi(x, y)$, e definimos fórmulas:

$$\begin{aligned} \psi & \text{ é } (\varphi(x, z) \wedge A = A) \\ \chi & \text{ é } \exists y \varphi(x, y) \end{aligned}$$

Usando Lema 69.4, como A é um parâmetro de ψ , existe um conjunto transitivo S tal que $0, 1, A \in S$ (juntamente com quaisquer outros parâmetros), e tal que:

$$(\forall x, z \in S)((\psi \leftrightarrow \psi^S) \wedge (\chi \leftrightarrow \chi^S))$$

⁶Mais formalmente, seja ξ qualquer uma destas fórmulas; então $\xi(z) \leftrightarrow \xi^S(z)$.

69.8. APÊNDICE: AXIOMATIZABILIDADE FINITA

Logo, em particular:

$$\begin{aligned} (\forall x, z \in S)(\varphi(x, z) \leftrightarrow \varphi^S(x, z)) \\ (\forall x \in S)(\exists y \varphi(x, y) \leftrightarrow (\exists y \in S)\varphi^S(x, y)) \end{aligned}$$

Combinando isto, e observando que $A \subseteq S$ pois $A \in S$ e S é transitivo:

$$(\forall x \in A)(\exists y \varphi(x, y) \leftrightarrow (\exists y \in S)\varphi(x, y))$$

Ora, $(\forall x \in A)(\exists! y \in S)\varphi(x, y)$, porque $(\forall x \in A)\exists! y \varphi(x, y)$. Ora, a Separação dá $\{y \in S : (\exists x \in A)\varphi(x, y)\} = \{y : (\exists x \in A)\varphi(x, y)\}$. $\square$

content/set-theory/replacement/finiteaxiomatizability.tex

69.8 Apêndice: Axiomatizabilidade finita

Terminamos este capítulo extraindo alguns resultados da Substituição. O primeiro resultado é devido a [Montague \(1961\)](#); note que não é uma prova *dentro* de **ZF**, mas uma prova *sobre* **ZF**:

Teorema 69.6. **ZF** não é finitamente axiomatizável. Mais geralmente: se **T** é finita e $\mathbf{T} \vdash \mathbf{ZF}$, então **T** é inconsistente.

(Aqui nos restringimos tacitamente a sentenças de primeira ordem cujo único primitivo não lógico é $\in$, e escrevemos $\mathbf{T} \vdash \mathbf{ZF}$ para indicar que $\mathbf{T} \vdash \varphi$ para todo o $\varphi \in \mathbf{ZF}$.)

Prova. Fixemos **T** finita tal que $\mathbf{T} \vdash \mathbf{ZF}$. Logo **T** prova a Reflexão, isto é, [Teorema 69.2](#). Como **T** é finita, podemos reescrevê-la como uma única conjunção, θ . Refletindo com esta fórmula, $\mathbf{T} \vdash \exists \beta (\theta \leftrightarrow \theta^{V_\beta})$. Como trivialmente $\mathbf{T} \vdash \theta$, obtemos $\mathbf{T} \vdash \exists \beta \theta^{V_\beta}$.

Ora, seja $\psi(X)$ uma abreviatura de:

$$\theta^X \wedge X \text{ é transitivo} \wedge (\forall Y \in X)(Y \text{ é transitivo} \rightarrow \neg \theta^Y)$$

isto diz, em traços largos: X é um modelo transitivo de θ , e $\in$ -minimal a esse respeito. Ora, recordando que $\mathbf{T} \vdash \exists \beta \theta^{V_\beta}$, por fatos básicos sobre postos dentro de **ZF** e portanto dentro de **T**, temos:

$$\mathbf{T} \vdash \exists M \psi(M). \quad (*)$$

Usando o primeiro conjunto de $\psi(X)$, sempre que $\mathbf{T} \vdash \sigma$, temos $\mathbf{T} \vdash \forall X (\psi(X) \rightarrow \sigma^X)$. Logo, por (*):

$$\mathbf{T} \vdash \forall X (\psi(X) \rightarrow (\exists N \psi(N))^X)$$

Usando isto, e (*) de novo:

$$\mathbf{T} \vdash \exists M (\psi(M) \wedge (\exists N \psi(N))^M)$$

Em particular, então:

$$\mathbf{T} \vdash \exists M(\psi(M) \wedge (\exists N \in M)((N \text{ é transitivo})^N \wedge (\theta^N)^M))$$

Logo, por raciocínio elementar sobre transitividade:

$$\mathbf{T} \vdash \exists M(\psi(M) \wedge (\exists N \in M)(N \text{ é transitivo} \wedge \theta^N))$$

Logo $\mathbf{T}$ é inconsistente.⁷

□

Eis um resultado semelhante, assinalado por [Potter \(2004, 223\)](#):

Proposição 69.7. *Seja $\mathbf{T}$ uma extensão de $\mathbf{Z}$ com um número finito de novos axiomas. Se $\mathbf{T} \vdash \mathbf{ZF}$, então $\mathbf{T}$ é inconsistente. (Aqui usamos as mesmas restrições tácitas que para [Teorema 69.6](#).)*

[sth:replacement:finiteaxiomatizab](#)
[finiteextensionofZ](#)

Prova. Usemos θ para a conjunção de todos os axiomas de $\mathbf{T}$ exceto as (infinitas) instâncias de Separação. Definindo ψ a partir de θ como no [Teorema 69.6](#), podemos mostrar que $\mathbf{T} \vdash \exists M\psi(M)$.

Como no [Teorema 69.6](#), podemos estabelecer o esquema de que, sempre que $\mathbf{T} \vdash \sigma$, temos $\mathbf{T} \vdash \forall X(\psi(X) \rightarrow \sigma^X)$. Terminamos a prova exatamente como no [Teorema 69.6](#).

No entanto, estabelecer o esquema envolve um pouco mais de trabalho do que no [Teorema 69.6](#). Afinal, as instâncias de Separação estão em $\mathbf{T}$, mas não são conjuntos de θ . Contudo, podemos ultrapassar este obstáculo provando que $\mathbf{T} \vdash \forall X(X \text{ é transitivo} \rightarrow \sigma^X)$, para toda a instância de Separação σ . Deixamos isto ao leitor. □

Problema 69.2. Mostre que, para toda a instância de Separação σ , tem-se: $\mathbf{Z} \vdash \forall X(X \text{ é transitivo} \rightarrow \sigma^X)$. (Usamos este esquema na [Proposição 69.7](#).)

Problema 69.3. Mostre que, para todo o $\varphi \in \mathbf{Z}$, tem-se $\mathbf{ZF} \vdash \varphi^{V_{\omega+\omega}}$.

Problema 69.4. Confirme os restantes resultados esquemáticos invocados nas provas de [Teorema 69.6](#) e [Proposição 69.7](#).

Como observamos na [Seção 69.5](#), isto mostra que a Substituição é estritamente mais forte que [Teorema 67.26](#). Ou, um pouco mais precisamente: se $\mathbf{Z}$

⁷Este “raciocínio elementar” envolve provar certos “fatos de absolutidade” para conjuntos transitivos.

+ “toda a boa ordenação é isomorfa a um ordinal único” for consistente, então não prova alguma instância da Substituição.

Capítulo 70

Aritmética Ordinal

`content/set-theory/ord-arithmetic/introduction.tex`

70.1 Introdução

`sth:ord-arithmetic:intro:`
`sec`

No **Capítulo 67**, desenvolvemos uma teoria dos números ordinais. Vimos no **Capítulo 68** que podemos pensar nos ordinais como uma espinha dorsal em torno da qual se constrói o restante da hierarquia. Mas esse não é o único papel dos ordinais. Há também a tarefa de efetuar a aritmética ordinal.

Já aludimos a isto, em **Seção 67.2**, quando falamos de ω , $\omega + 1$ e $\omega + \omega$. Na altura falamos informalmente; chegou a altura de o expor com rigor. Devemos contudo mencionar que não há muita filosofia neste capítulo; apenas desenvolvimentos técnicos, juntamente com uma observação (ligeiramente) interessante de que podemos fazer a mesma coisa de duas maneiras diferentes.

`content/set-theory/ord-arithmetic/addition.tex`

70.2 Adição Ordinal

`sth:ord-arithmetic:add:`
`sec`

Suponha-se que se pretende somar α e β . Pode simplesmente colocar-se uma cópia de β imediatamente a seguir a uma cópia de α . (É preciso tomar *cópias*, pois sabemos por **Proposição 67.22** que ou $\alpha \subseteq \beta$ ou $\beta \subseteq \alpha$.) O efeito intuitivo disto é percorrer uma α -sucessão de passos, e *depois* percorrer uma β -sucessão. A sucessão resultante fica bem ordenada; logo, por **Teorema 67.26** é isomorfa a um ordinal (único). Esse ordinal pode considerar-se a *soma* de α e β .

Esta é a ideia intuitiva por detrás da adição ordinal. Para a definir com rigor, começa-se pela ideia de tomar *cópias* de conjuntos. A ideia é usar etiquetas arbitrárias, 0 e 1, para saber de onde veio cada objeto:

`sth:ord-arithmetic:add:`
`defdissum`

Definição 70.1. A *soma disjunta* de A e B é $A \sqcup B = (A \times \{0\}) \cup (B \times \{1\})$.

Define-se em seguida uma ordem em pares de ordinais:

Definição 70.2. Para quaisquer ordinais $\alpha_1, \alpha_2, \beta_1, \beta_2$, diga-se que:

$$\langle \alpha_1, \alpha_2 \rangle \triangleleft \langle \beta_1, \beta_2 \rangle \text{ sse ou } \alpha_2 \in \beta_2 \\ \text{ou tanto } \alpha_2 = \beta_2 \text{ como } \alpha_1 \in \beta_1$$

Trata-se de uma ordem *lexicográfica inversa*, pois ordena-se pelo segundo elemento e depois pelo primeiro. Recorde-se que se pretendia definir $\alpha + \beta$ como o tipo de ordem de uma cópia de α seguida de uma cópia de β . Para o conseguir, define-se:

Definição 70.3. Para quaisquer ordinais α, β , a sua soma é $\alpha + \beta = \text{ord}(\alpha \sqcup \beta, \triangleleft)$. sth:ord-arithmetic:add: defordplus

Note-se que aqui se abusou ligeiramente da notação; estritamente deveria escrever-se “ $\{\langle x, y \rangle \in \alpha \sqcup \beta : x \triangleleft y\}$ ” em lugar de “ $\triangleleft$ ”. Por brevidade, contudo, continuaremos a abusar da notação deste modo no que se segue.

O resultado seguinte, juntamente com **Teorema 67.26**, confirma que a nossa definição está bem formada:

Lema 70.4. $\langle \alpha \sqcup \beta, \triangleleft \rangle$ é um bom ordenamento, para quaisquer ordinais α e β . sth:ord-arithmetic:add: ordsumlessiswo

Prova. Obviamente $\triangleleft$ é conexa em $\alpha \sqcup \beta$. Para mostrar que é bem fundada, fixe-se um $X \subseteq \alpha \sqcup \beta$ não vazio. Seja Y o subconjunto de X cuja segunda coordenada é a menor possível, isto é, $Y = \{\langle \gamma, i \rangle \in X : (\forall \langle \delta, j \rangle \in X) i \leq j\}$. Escolha-se agora o elemento de Y com menor primeira coordenada. $\square$

Temos assim uma definição explícita e clara da adição ordinal. Eis um fato não surpreendente (recorde-se que $1 = \{0\}$, por **Definição 66.7**):

Proposição 70.5. $\alpha + 1 = \alpha^+$, para qualquer ordinal α .

Prova. Considere-se o isomorfismo f de $\alpha^+ = \alpha \cup \{\alpha\}$ para $\alpha \sqcup 1 = (\alpha \times \{0\}) \sqcup (\{0\} \times \{1\})$ dado por $f(\gamma) = \langle \gamma, 0 \rangle$ para $\gamma \in \alpha$, e $f(\alpha) = \langle 0, 1 \rangle$. $\square$

Além disso, é fácil mostrar que a adição obedece a certas condições recursivas:

Lema 70.6. Para quaisquer ordinais α, β , tem-se: sth:ord-arithmetic:add: ordadditionrecursion

$$\begin{aligned} \alpha + 0 &= \alpha \\ \alpha + (\beta + 1) &= (\alpha + \beta) + 1 \\ \alpha + \beta &= \text{lsub}_{\delta < \beta}(\alpha + \delta) \end{aligned} \quad \text{se } \beta \text{ é um ordinal limite}$$

70.2. ADIÇÃO ORDINAL

Prova. Verifica-se caso a caso; primeiro:

$$\begin{aligned}
 \alpha + 0 &= \text{ord}((\alpha \times \{0\}) \cup (0 \times \{1\}), \triangleleft) \\
 &= \text{ord}((\alpha \times \{0\}) \cup \{0\}, \triangleleft) \\
 &= \alpha \\
 \alpha + (\beta + 1) &= \text{ord}((\alpha \times \{0\}) \cup (\beta^+ \times \{1\}), \triangleleft) \\
 &= \text{ord}((\alpha \times \{0\}) \cup (\beta \times \{1\}), \triangleleft) + 1 \\
 &= (\alpha + \beta) + 1
 \end{aligned}$$

Seja agora $\beta \neq \emptyset$ um limite. Se $\delta < \beta$ então também $\delta + 1 < \beta$, logo $\alpha + \delta$ é um segmento inicial próprio de $\alpha + \beta$. Logo $\alpha + \beta$ é um majorante estrito de $X = \{\alpha + \delta : \delta < \beta\}$. Além disso, se $\alpha \leq \gamma < \alpha + \beta$, então claramente $\gamma = \alpha + \delta$ para algum $\delta < \beta$. Logo $\alpha + \beta = \text{lsub}_{\delta < \beta}(\alpha + \delta)$. $\square$

Mas eis um fato marcante. Para definir a adição ordinal, poder-se-ia *em alternativa* ter usado simplesmente o teorema da recursão transfinita, e estipulado as equações recursivas, exatamente como em **Lema 70.6** (embora com “ β^+ ” em vez de “ $\beta + 1$ ”).

Há, pois, duas maneiras diferentes de definir operações nos ordinais. Podem definir-se *sinteticamente*, construindo explicitamente um conjunto bem ordenado e considerando o seu tipo de ordem. Ou podem definir-se *recursivamente*, apenas estipulando as equações recursivas. Feito corretamente, porém, o resultado é idêntico. De fato, **Teorema 67.26** garante que estas equações recursivas determinam ordinais *únicos*.

Em muitos aspectos, a aritmética ordinal comporta-se como a adição dos números naturais. Por exemplo, pode provar-se o seguinte:

Lema 70.7. *Se α, β, γ são ordinais, então:*

1. *se $\beta < \gamma$, então $\alpha + \beta < \alpha + \gamma$*
2. *se $\alpha + \beta = \alpha + \gamma$, então $\beta = \gamma$*
3. *$\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma$, isto é, a adição é associativa*
4. *Se $\alpha \leq \beta$, então $\alpha + \gamma \leq \beta + \gamma$*

Prova. Provamos (3), deixando o resto como exercício. A prova é por indução transfinita simples em γ , usando **Lema 70.6**. Quando $\gamma = 0$:

$$(\alpha + \beta) + 0 = \alpha + \beta = \alpha + (\beta + 0)$$

Quando $\gamma = \delta + 1$, suponha-se por indução que $(\alpha + \beta) + \delta = \alpha + (\beta + \delta)$; usando **Lema 70.6** três vezes:

$$\begin{aligned}
 (\alpha + \beta) + (\delta + 1) &= ((\alpha + \beta) + \delta) + 1 \\
 &= (\alpha + (\beta + \delta)) + 1 \\
 &= \alpha + ((\beta + \delta) + 1) \\
 &= \alpha + (\beta + (\delta + 1))
 \end{aligned}$$

Quando γ é um ordinal limite, suponha-se por indução que se $\delta \in \gamma$ então $(\alpha + \beta) + \delta = \alpha + (\beta + \delta)$; então:

$$\begin{aligned} (\alpha + \beta) + \gamma &= \text{lsub}_{\delta < \gamma}((\alpha + \beta) + \delta) \\ &= \text{lsub}_{\delta < \gamma}(\alpha + (\beta + \delta)) \\ &= \alpha + \text{lsub}_{\delta < \gamma}(\beta + \delta) \\ &= \alpha + (\beta + \gamma) \end{aligned} \quad \square$$

Problema 70.1. Prove o resto de [Lema 70.7](#).

Deste modo, a adição ordinal deve parecer muito familiar. Há contudo um ponto crucial em que a adição ordinal *não* se parece com a adição nos naturais.

Proposição 70.8. *A adição ordinal não é comutativa; $1 + \omega = \omega < \omega + 1$.*

[sth:ord-arithmetic:add:](#)
[ordsumnotcommute](#)

Prova. Note-se que $1 + \omega = \text{lsub}_{n < \omega}(1 + n) = \omega \in \omega \cup \{\omega\} = \omega^+ = \omega + 1$. $\square$

Embora isto possa surpreender à primeira vista, *não devia*. Por um lado, ao considerar $1 + \omega$, pensa-se no tipo de ordem que se obtém colocando um elemento extra *antes* de todos os números naturais. Raciocinando como com o Hotel de Hilbert em [Seção 6.1](#), intuitivamente esse primeiro elemento extra não altera o tipo de ordem global. Por outro lado, ao considerar $\omega + 1$, pensa-se no tipo de ordem que se obtém colocando um elemento extra *depois* de todos os números naturais. E isso é algo radicalmente diferente!

[content/set-theory/ord-arithmetic/using-addition.tex](#)

70.3 Usando Adição Ordinal

Com a adição nos ordinais, podemos calcular explicitamente os postos de vários conjuntos, no sentido de [Definição 68.15](#):

[sth:ord-arithmetic:using-addition:](#)
[sec](#)

Lema 70.9. *Se $\text{posto}(A) = \alpha$ e $\text{posto}(B) = \beta$, então:*

1. $\text{posto}(\wp(A)) = \alpha + 1$
2. $\text{posto}(\{A, B\}) = \max(\alpha, \beta) + 1$
3. $\text{posto}(A \cup B) = \max(\alpha, \beta)$
4. $\text{posto}(\langle A, B \rangle) = \max(\alpha, \beta) + 2$
5. $\text{posto}(A \times B) \leq \max(\alpha, \beta) + 2$
6. $\text{posto}(\bigcup A) = \alpha$ quando α é vazio ou limite; $\text{posto}(\bigcup A) = \gamma$ quando $\alpha = \gamma + 1$

[sth:ord-arithmetic:using-addition:](#)
[rankcomputation](#)
[sth:ord-arithmetic:using-addition:](#)
[exrankpow](#)
[sth:ord-arithmetic:using-addition:](#)
[exrankpair](#)
[sth:ord-arithmetic:using-addition:](#)
[exrankcup](#)
[sth:ord-arithmetic:using-addition:](#)
[exranktuple](#)
[sth:ord-arithmetic:using-addition:](#)
[exranktimes](#)
[sth:ord-arithmetic:using-addition:](#)
[exrankunion](#)

70.3. USANDO ADIÇÃO ORDINAL

Prova. Ao longo de todo o raciocínio, invocamos repetidamente **Proposição 68.20**.

(1). Se $x \subseteq A$ então $\text{posto}(x) \leq \text{posto}(A)$. Logo $\text{posto}(\wp(A)) \leq \alpha + 1$. Como em particular $A \in \wp(A)$, tem-se $\text{posto}(\wp(A)) = \alpha + 1$.

(2). Por **Proposição 68.20**

(3). Por **Proposição 68.20**.

(4). Por (2), duas vezes.

(5). Note-se que $A \times B \subseteq \wp(\wp(A \cup B))$, e invoque (4).

(6). Se $\alpha = \gamma + 1$, existe algum $c \in A$ com $\text{posto}(c) = \gamma$, e nenhum membro de A tem posto superior; logo $\text{posto}(\bigcup A) = \gamma$. Se α é um ordinal limite, então A tem membros com posto arbitrariamente próximo (mas estritamente inferior) de α , pelo que $\bigcup A$ também tem membros com posto arbitrariamente próximo (mas estritamente inferior) de α , e portanto $\text{posto}(\bigcup A) = \alpha$. $\square$

Deixamos como exercício mostrar por que (5) envolve uma desigualdade.

Problema 70.2. Exiba conjuntos A e B tais que $\text{posto}(A \times B) = \max(\text{posto}(A), \text{posto}(B))$. Exiba conjuntos A e B tais que $\text{posto}(A \times B) = \max(\text{posto}(A), \text{posto}(B)) + 2$. São possíveis outros postos?

Estamos também agora em condições de mostrar que várias noções razoáveis do que possa significar descrever um ordinal como “finito” ou “infinito” coincidem:

Lema 70.10. Para qualquer ordinal α , as seguintes afirmações são equivalentes:

1. $\alpha \notin \omega$, isto é, α não é um número natural
2. $\omega \leq \alpha$
3. $1 + \alpha = \alpha$
4. $\alpha \approx \alpha + 1$, isto é, α e $\alpha + 1$ são equipotentes
5. α é infinito de Dedekind

Temos assim cinco formas provavelmente equivalentes de precisar o que exige que um ordinal seja (in)finito.

Prova. (1) $\Rightarrow$ (2). Por tricotomia.

(2) $\Rightarrow$ (3). Fixe $\alpha \geq \omega$. Por indução transfinita, existe um ordinal γ mínimo (possivelmente 0) tal que existe um ordinal limite β com $\alpha = \beta + \gamma$. Ora:

$$1 + \alpha = 1 + (\beta + \gamma) = (1 + \beta) + \gamma = \text{lsub}_{\delta < \beta} (1 + \delta) + \gamma = \beta + \gamma = \alpha.$$

(3) $\Rightarrow$ (4). Existe claramente uma bijeção $f: (\alpha \sqcup 1) \rightarrow (1 \sqcup \alpha)$. Se $1 + \alpha = \alpha$, existe um isomorfismo $g: (1 \sqcup \alpha) \rightarrow \alpha$. Considere-se $g \circ f$.

(4) $\Rightarrow$ (5). Se $\alpha \approx \alpha + 1$, existe uma bijeção $f: (\alpha \sqcup 1) \rightarrow \alpha$. Defina $g(\gamma) = f(\gamma, 0)$ para cada $\gamma < \alpha$; esta injeção mostra que α é infinito de Dedekind, pois $f(0, 1) \in \alpha \setminus \text{Im}(g)$.

(5) $\Rightarrow$ (1). Isto é **Proposição 66.8**. $\square$

content/set-theory/ord-arithmetic/multiplication.tex

70.4 Multiplicação Ordinal

Passamos agora à multiplicação ordinal, abordando-a de forma muito semelhante à adição ordinal. Suponha-se que se pretende multiplicar α por β . Pode imaginar-se uma grelha retangular, com largura α e altura β ; o produto de α e β é então o resultado de percorrer cada linha e depois passar à linha seguinte. . . até percorrer toda a grelha. Dito de outro modo, o produto de α e β obtém-se substituindo *cada* elemento de β por uma cópia de α .

sth:ord-arithmetic:mult:
sec

Para formalizar, usa-se simplesmente a ordem lexicográfica inversa no produto cartesiano de α e β :

Definição 70.11. Para quaisquer ordinais α, β , o seu produto $\alpha \cdot \beta = \text{ord}(\alpha \times \beta, \triangleleft)$.

É preciso de novo confirmar que esta definição está bem formada:

Lema 70.12. $\langle \alpha \times \beta, \triangleleft \rangle$ é um bom ordenamento, para quaisquer ordinais α e β .

sth:ord-arithmetic:mult:
ordtimeslessiswo

Prova. Exatamente como para [Lema 70.4](#). □

E não é difícil provar que a multiplicação se comporta assim:

Lema 70.13. Para quaisquer ordinais α, β :

sth:ord-arithmetic:mult:
ordtimesrecursion

$$\begin{aligned} \alpha \cdot 0 &= 0 \\ \alpha \cdot (\beta + 1) &= (\alpha \cdot \beta) + \alpha \\ \alpha \cdot \beta &= \text{lsub}_{\delta < \beta}(\alpha \cdot \delta) \quad \text{quando } \beta \text{ é um ordinal limite.} \end{aligned}$$

Prova. Deixado como exercício. □

De fato, tal como no caso da adição, poderíamos ter definido a multiplicação ordinal por estas equações recursivas, em vez de dar uma definição direta. Tal como na adição, certo comportamento é familiar:

Lema 70.14. Se α, β, γ são ordinais, então:

sth:ord-arithmetic:mult:
ordinalmultiplicationisnice
sth:ord-arithmetic:mult:
ordtimes1
sth:ord-arithmetic:mult:
ordtimes2
sth:ord-arithmetic:mult:
ordtimes3
sth:ord-arithmetic:mult:
ordtimes4
sth:ord-arithmetic:mult:
ordtimes5

1. se $\alpha \neq 0$ e $\beta < \gamma$, então $\alpha \cdot \beta < \alpha \cdot \gamma$;
2. se $\alpha \neq 0$ e $\alpha \cdot \beta = \alpha \cdot \gamma$, então $\beta = \gamma$;
3. $\alpha \cdot (\beta \cdot \gamma) = (\alpha \cdot \beta) \cdot \gamma$;
4. Se $\alpha \leq \beta$, então $\alpha \cdot \gamma \leq \beta \cdot \gamma$;
5. $\alpha \cdot (\beta + \gamma) = (\alpha \cdot \beta) + (\alpha \cdot \gamma)$.

70.5. EXPONENCIAÇÃO ORDINAL

Prova. Deixado como exercício. $\square$

Pode provar-se (ou consultar) outros resultados, à vontade. Mas, dado **Proposição 70.8**, o que se segue não deve surpreender:

Proposição 70.15. *A multiplicação ordinal não é comutativa: $2 \cdot \omega = \omega < \omega \cdot 2$*

Prova. $2 \cdot \omega = \text{lsub}_{n < \omega}(2 \cdot n) = \omega \in \text{lsub}_{n < \omega}(\omega + n) = \omega + \omega = \omega \cdot 2$. $\square$

De novo, a justificação intuitiva é direta. Para calcular $2 \cdot \omega$, substitui-se cada número natural por duas entidades. Obtém-se o mesmo tipo de ordem se simplesmente se inserirem todos os números “meios” nos naturais, isto é, se se considerar a ordem natural em $\{n/2 : n \in \omega\}$. Assim posto, o tipo de ordem é claramente o mesmo que o de ω em si. Mas, para calcular $\omega \cdot 2$, dispõem-se duas cópias de ω , uma a seguir à outra.

Problema 70.3. Prove **Lema 70.12**, **Lema 70.13**, e **Lema 70.14**

content/set-theory/ord-arithmetic/exponentiation.tex

70.5 Exponenciação Ordinal

sth:ord-arithmetic:expo:
sec

Passamos agora à exponenciação ordinal. Infelizmente, não há uma definição *simpática* sintética para a exponenciação ordinal.

Certo que *existem* definições sintéticas explícitas. Eis uma. Seja $\text{finfun}(\alpha, \beta)$ o conjunto de todas as funções $f: \alpha \rightarrow \beta$ tais que $\{\gamma \in \alpha : f(\gamma) \neq 0\}$ é equipotente a algum número natural. Defina-se um bom ordenamento em $\text{finfun}(\alpha, \beta)$ por $f \sqsubset g$ sse $f \neq g$ e $f(\gamma_0) < g(\gamma_0)$, onde $\gamma_0 = \max\{\gamma \in \alpha : f(\gamma) \neq g(\gamma)\}$. Então pode definir-se $\alpha^{(\beta)}$ como $\text{ord}(\text{finfun}(\alpha, \beta), \sqsubset)$. Potter usa esta definição explícita e explica de seguida:

A escolha desta ordenação determina-se unicamente pelo desejo de obter uma definição da exponenciação ordinal que obedeça à condição recursiva adequada. . . , e é muito mais difícil de visualizar do que a soma ou o produto ordenados. (Potter, 2004, p. 199)

Exatamente. Explicamos a adição como “uma cópia de α seguida de uma cópia de β ”, e a multiplicação como “uma β -sucessão de cópias de α ”. Mas não temos nada de sucinto a dizer sobre $\text{finfun}(\alpha, \gamma)$. Por isso, apresentaremos a definição da exponenciação ordinal apenas *por* recursão transfinita, isto é:

sth:ord-arithmetic:expo:
ordexporecursion

Definição 70.16.

$$\begin{aligned} \alpha^{(0)} &= 1 \\ \alpha^{(\beta+1)} &= \alpha^{(\beta)} \cdot \alpha \\ \alpha^{(\beta)} &= \bigcup_{\delta < \beta} \alpha^{(\delta)} \end{aligned} \quad \text{quando } \beta \text{ é um ordinal limite}$$

Se trabalhássemos *como* teóricos de conjuntos, poderíamos querer explorar algumas propriedades da exponenciação ordinal. Mas pouco mais temos a acrescentar, salvo notar o fato pouco surpreendente de que a exponenciação ordinal não comuta. Assim $2^{(\omega)} = \bigcup_{\delta < \omega} 2^{(\delta)} = \omega$, enquanto $\omega^{(2)} = \omega \cdot \omega$. Mas então não se deve *esperar* que comute, pois já não comuta nos naturais: $2^{(3)} = 8 < 9 = 3^{(2)}$.

Problema 70.4. Usando indução transfinita, prove que, se se define $\alpha^{(\beta)} = \text{ord}(\text{finfun}(\alpha, \beta), \sqsubset)$, obtêm-se as equações recursivas de [Definição 70.16](#).

Capítulo 71

Cardinais

[content/set-theory/cardinals/cp.tex](#)

71.1 Princípio de Cantor

Volte o pensamento a [Seção 67.5](#). Estávamos a discutir conjuntos bem ordenados e sugerimos que seria desejável ter objetos que funcionem como representantes dos bons ordenamentos. Com isto em mente, introduzimos os ordinais e depois mostramos em [Corolário 67.28](#) que estes se comportam como desejamos, isto é:

[sth:cardinals:cp:sec](#)

$$\text{ord}(A, <) = \text{ord}(B, <) \text{ sse } \langle A, < \rangle \cong \langle B, < \rangle.$$

Volte ainda mais atrás, a [Seção 4.8](#). Ali, trabalhando ingenuamente, introduzimos a noção de “tamanho” de um conjunto. Especificamente, dissemos que dois conjuntos são equipotentes, $A \approx B$, precisamente quando existe uma [bijeção](#) $f: A \rightarrow B$. Esta noção é intrinsecamente mais simples do que a de um bom ordenamento: interessam-nos apenas as [bijeções](#), e não (como com os isomorfismos de ordem) se as [bijeções](#) “preservam alguma estrutura”.

Tudo isto sugere um pensamento óbvio. Tal como introduzimos certos objetos, *ordinais*, para calibrar bons ordenamentos, podemos introduzir certos objetos, *cardinais*, para calibrar tamanho. Esse é o objetivo deste capítulo.

Antes de dizer o que serão estes cardinais, convém enunciar um princípio que devem satisfazer. Escrevendo $|X|$ para a cardinalidade do conjunto X ,

71.2. CARDINAIS COMO ORDINAIS

desejávamos que obedecessem a:

$$|A| = |B| \text{ sse } A \approx B.$$

Chamaremos a isto o Princípio de Cantor, já que Cantor foi provavelmente o primeiro a tê-lo claramente presente. (Diremos mais sobre a sua relação com o Princípio de Hume na Seção 71.5.) O nosso objetivo é definir $|X|$, para cada X , de modo a que satisfaça o Princípio de Cantor.

content/set-theory/cardinals/cardsasords.tex

71.2 Cardinais como Ordinais

sth:cardinals:cardsasords:
sec

De fato, a nossa teoria dos cardinais fará uso (descarado) da nossa teoria dos ordinais. Ou seja: definiremos os cardinais como certos ordinais específicos. Em particular, proporemos o seguinte:

sth:cardinals:cardsasords:
defcardinalasordinal

Definição 71.1. Se A pode ser bem ordenado, então $|A|$ é o menor ordinal γ tal que $A \approx \gamma$. Para qualquer ordinal γ , diz-se que γ é um *cardinal* sse $\gamma = |\gamma|$.

Acabamos de usar a locução “ A pode ser bem ordenado”. Como quase sempre em matemática, a locução modal aqui é apenas um atalho para uma asserção existencial: dizer “ A pode ser bem ordenado” é dizer “existe uma relação que bem ordena A ”.

Há contudo um problema com Definição 71.1. Gostaríamos que *todo* o conjunto tivesse um tamanho, isto é, que $|A|$ existisse para todo A . A definição que acabamos de dar, porém, começa por uma condicional: “Se A pode ser bem ordenado...”. Se existir algum conjunto A que não possa ser bem ordenado, a nossa definição simplesmente deixa de definir um objeto $|A|$.

Logo, para usar Definição 71.1, precisamos de uma garantia de que todo o conjunto pode ser bem ordenado. Infelizmente, essa garantia não está disponível em **ZF**. Assim, se queremos usar Definição 71.1, não há alternativa senão acrescentar um novo axioma, tal como:

Bom ordenamento (Bom ordenamento). Todo o conjunto pode ser bem ordenado.

Discutiremos se o axioma do bom ordenamento é aceitável em Capítulo 73. Daqui em diante, porém, simplesmente o admitiremos. E, com ele, é bastante direto provar que os cardinais (tal como definidos na Definição 71.1) existem e comportam-se bem:

sth:cardinals:cardsasords:
lem:CardinalsExist
sth:cardinals:cardsasords:
cardaexists
sth:cardinals:cardsasords:
cardaapprox
sth:cardinals:cardsasords:
cardaidem

Lema 71.2. Para todo o conjunto A :

1. $|A|$ existe e é único;
2. $|A| \approx A$;
3. $|A|$ é um cardinal, isto é, $|A| = ||A||$;

Prova. Fixe A . Pelo Bom ordenamento, existe um bom ordenamento $\langle A, R \rangle$. Por [Teorema 67.26](#), $\langle A, R \rangle$ é isomorfo a um único ordinal, β . Logo $A \approx \beta$. Por indução transfinita, existe um único ordinal mínimo, γ , tal que $A \approx \gamma$. Logo $|A| = \gamma$, o que estabelece (1) e (2). Para (3), note-se que se $\delta \in \gamma$ então $\delta \prec A$, pela escolha de γ , logo também $\delta \prec \gamma$ pois a equipotência é uma relação de equivalência ([Proposição 4.20](#)). Logo $\gamma = |\gamma|$. $\square$

O resultado seguinte garante o Princípio de Cantor, e mais ainda. (Note-se que os cardinais herdam a ordem dos ordinais, isto é, $\mathfrak{a} < \mathfrak{b}$ sse $\mathfrak{a} \in \mathfrak{b}$. Ao formular isto, usaremos letras Fraktur para objetos que sabemos serem cardinais. Isto é bastante usual. Uma alternativa comum é usar letras gregas, já que os cardinais são ordinais, escolhendo-as do meio do alfabeto, por exemplo κ, λ .)

Lema 71.3. *Para quaisquer conjuntos A e B :*

*sth:cardinals:cardsasords:
lem:CardinalsBehaveRight*

$$A \approx B \text{ sse } |A| = |B|$$

$$A \preceq B \text{ sse } |A| \leq |B|$$

$$A \prec B \text{ sse } |A| < |B|$$

Prova. Provaremos o sentido da esquerda para a direita da segunda afirmação (os outros casos são semelhantes e ficam como exercício). Considere-se o diagrama seguinte:

$$\begin{array}{ccc} A & \xrightarrow{\quad} & B \\ \updownarrow & & \updownarrow \\ |A| & \dashrightarrow & |B| \end{array}$$

As setas duplas indicam [bijeções](#), cuja existência é garantida por [Lema 71.2](#). Ao supor $A \preceq B$, existe [uma injeção](#) $A \rightarrow B$. Seguindo as setas de $|A|$ para A , para B e para $|B|$, obtém-se [uma injeção](#) $|A| \rightarrow |B|$ (a seta tracejada). $\square$

Pode ainda usar-se [Lema 71.3](#) para voltar a provar Cantor–Bernstein. Trata-se da afirmação de que se $A \preceq B$ e $B \preceq A$ então $A \approx B$. A enunciamos como [Teorema 4.25](#), mas a provamos primeiro—com algum esforço—em [Seção 6.5](#). Considere-se:

Nova prova de Schröder–Bernstein. Se $A \preceq B$ e $B \preceq A$, então $|A| \leq |B|$ e $|B| \leq |A|$ por [Lema 71.3](#). Logo $|A| = |B|$ e $A \approx B$ por tricotomia e [Lema 71.3](#). $\square$

Embora seja uma prova muito simples, apoia-se implicitamente na Substituição (para assegurar [Teorema 67.26](#)) e no Bom ordenamento (para garantir [Lema 71.3](#)). Em contraste, a prova na [Seção 6.5](#) era muito mais autônoma (de fato, pode efetuar-se em $\mathbf{Z}^-$).

71.3. ZFC: UM MARCO

71.3 ZFC: um marco

sth:cardinals:zfc:
sec

Com a adição do axioma do bom ordenamento, atingimos o último marco teórico. Disponemos agora de todos os axiomas necessários para **ZFC**. Em detalhe:

Definição 71.4. A teoria **ZFC** tem estes axiomas: Extensionalidade, União, Pares, Conjuntos potência, Infinitude, Fundação, Bom ordenamento e todas as instâncias dos esquemas de Separação e Substituição. Dito de outro modo, **ZFC** acrescenta o Bom ordenamento a **ZF**.

ZFC significa teoria de conjuntos de *Zermelo–Fraenkel* com *Escolha*. Isto pode parecer ligeiramente estranho, pois o axioma que adicionamos se chama “Bom ordenamento”, não “Escolha”. Mas, quando mais tarde formularmos a *Escolha*, verificar-se-á que o Bom ordenamento lhe é equivalente (módulo **ZF**) (ver **Teorema 73.6**). Logo, qual tomar como axioma “básico” é indiferente. E o nome “**ZFC**” é inteiramente standard na literatura.

content/set-theory/cardinals/classing.tex

71.4 Finito, Enumerável, Não Enumerável

sth:cardinals:classing:
sec

Agora que fomos introduzidos aos cardinais, vale a pena dedicar um pouco de tempo a variedades de cardinais; nomeadamente, cardinais finitos, **enumerável** e **não enumerável**.

Os nossos dois primeiros resultados implicam que os cardinais finitos serão exatamente os ordinais finitos, que definimos como os nossos *números naturais* em **Definição 66.7**:

sth:cardinals:classing:
finitecardisoequal

Proposição 71.5. *Sejam $n, m \in \omega$. Então $n = m$ sse $n \approx m$.*

Prova. O sentido da esquerda para a direita é trivial. Para provar da direita para a esquerda, suponha-se $n \approx m$ embora $n \neq m$. Por tricotomia, ou $n \in m$ ou $m \in n$; suponha-se, sem perda de generalidade, $n \in m$. Então $n \subsetneq m$ e existe uma **bijeção** $f: m \rightarrow n$, logo m é infinito de Dedekind, o que contradiz **Proposição 66.8**. $\square$

sth:cardinals:classing:
naturalsarecardinals

Corolário 71.6. *Se $n \in \omega$, então n é um cardinal.*

Prova. Imediato. $\square$

Segue-se também que várias noções razoáveis do que possa significar descrever um cardinal como “finito” ou “infinito” coincidem:

sth:cardinals:classing:
generalinfinitycharacter

Teorema 71.7. *Para qualquer conjunto A , as seguintes afirmações são equivalentes:*

sth:cardinals:classing:
card:notinomega

1. $|A| \notin \omega$, isto é, A não é um número natural;

2. $\omega \leq |A|$;

3. A é infinito de Dedekind.

*sth:cardinals:classing:
card:omegaplus*

*sth:cardinals:classing:
card:infinite*

Prova. Por **Lema 70.10**, **Lema 71.3**, e **Corolário 71.6**. □

Isto autoriza a seguinte *definição* de algumas noções que usamos de forma informal na **Parte I**:

Definição 71.8. Diz-se que A é *finito* sse $|A|$ é um número natural, isto é, $|A| \in \omega$. Caso contrário, diz-se que A é *infinito*.

*sth:cardinals:classing:
defnfinite*

Note-se que esta definição é apresentada no contexto de **ZFC**. Afinal, precisamos do Bom ordenamento para garantir que todo o conjunto tem uma cardinalidade. E, de fato, sem o Bom ordenamento, pode existir um conjunto que nem é finito nem infinito de Dedekind. Voltaremos a este tipo de questão no **Capítulo 73**. Por ora, continuamos a assumir o Bom ordenamento.

Passemos agora dos cardinais finitos aos cardinais infinitos. Eis dois pontos elementares:

Corolário 71.9. ω é o menor cardinal infinito.

*sth:cardinals:classing:
omegaisacardinal*

Prova. ω é um cardinal, pois ω é infinito de Dedekind e se $\omega \approx n$ para algum $n \in \omega$ então n seria infinito de Dedekind, contrariando **Proposição 66.8**. Ora, ω é o menor cardinal infinito por definição. □

Corolário 71.10. *Todo o cardinal infinito é um ordinal limite.*

Prova. Seja α um ordinal sucessor infinito, logo $\alpha = \beta + 1$ para algum β . Por **Proposição 71.5**, β também é infinito, logo $\beta \approx \beta + 1$ por **Lema 70.10**. Ora, $|\beta| = |\beta + 1| = |\alpha|$ por **Lema 71.3**, logo $\alpha \neq |\alpha|$. □

Já na **Definição 4.27**, assinalamos que podemos distinguir conjuntos infinitos **enumerável** de conjuntos infinitos **não enumerável**. Essa definição conduz naturalmente ao seguinte:

Proposição 71.11. A é **enumerável** sse $|A| \leq \omega$, e A é **não enumerável** sse $\omega < |A|$.

Prova. Por tricotomia, as duas afirmações são equivalentes, bastando provar que A é **enumerável** sse $|A| \leq \omega$. Para *da direita para a esquerda*: se $|A| \leq \omega$, então $A \preceq \omega$ por **Lema 71.3** e **Corolário 71.9**. Para *da esquerda para a direita*: suponha-se A **enumerável**; então, por **Definição 4.27**, há três casos possíveis:

1. se $A = \emptyset$, então $|A| = 0 \in \omega$, por **Corolário 71.6** e **Lema 71.3**.
2. se $n \approx A$, então $|A| = n \in \omega$, por **Corolário 71.6** e **Lema 71.3**.
3. se $\omega \approx A$, então $|A| = \omega$, por **Corolário 71.9**.

71.5. APÊNDICE: PRINCÍPIO DE HUME

Em todos os casos, $|A| \leq \omega$. $\square$

De fato, ω tem um lugar especial. Embora existam muitos ordinais contáveis:

Corolário 71.12. ω é o único cardinal infinito *enumerável*.

Prova. Seja $\mathfrak{a}$ um cardinal infinito enumerável. Como $\mathfrak{a}$ é infinito, $\omega \leq \mathfrak{a}$. Como $\mathfrak{a}$ é um cardinal infinito enumerável, $\mathfrak{a} = |\mathfrak{a}| \leq \omega$. Logo $\mathfrak{a} = \omega$ por tricotomia. $\square$

Claro que existem infinitos cardinais. Poderíamos perguntar: *Quantos cardinais há?* Os resultados seguintes mostram que convém reconsiderar essa pergunta.

*sth:cardinals:classing:
unioncardinalscardinal*

Proposição 71.13. Se todo o membro de X é um cardinal, então $\bigcup X$ é um cardinal.

Prova. É fácil verificar que $\bigcup X$ é um ordinal. Seja $\alpha \in \bigcup X$ um ordinal; então $\alpha \in \mathfrak{b} \in X$ para algum cardinal $\mathfrak{b}$. Como $\mathfrak{b}$ é um cardinal, $\alpha \prec \mathfrak{b}$. Como $\mathfrak{b} \subseteq \bigcup X$, tem-se $\mathfrak{b} \preceq \bigcup X$, e portanto $\alpha \preceq \bigcup X$. Generalizando, $\bigcup X$ é um cardinal. $\square$

*sth:cardinals:classing:
lem:NoLargestCardinal*

Teorema 71.14. Não existe um cardinal máximo.

Prova. Para qualquer cardinal $\mathfrak{a}$, o teorema de Cantor (**Teorema 4.24**) e **Lema 71.2** implicam que $\mathfrak{a} < |\wp(\mathfrak{a})|$. $\square$

Teorema 71.15. O conjunto de todos os cardinais não existe.

Prova. Por redução ao absurdo, suponha-se $C = \{\mathfrak{a} : \mathfrak{a} \text{ é um cardinal}\}$. Ora, $\bigcup C$ é um cardinal por **Proposição 71.13**, logo por **Teorema 71.14** existe um cardinal $\mathfrak{b} > \bigcup C$. Por definição $\mathfrak{b} \in C$, logo $\mathfrak{b} \subseteq \bigcup C$, donde $\mathfrak{b} \leq \bigcup C$, uma contradição. $\square$

Compare isto com o paradoxo de Russell e com Burali-Forti.

content/set-theory/cardinals/hp.tex

71.5 Apêndice: Princípio de Hume

*sth:cardinals:hp:
sec*

Na **Seção 71.1**, descrevemos o Princípio de Cantor. Era:

$$|A| = |B| \text{ sse } A \approx B.$$

Isto é muito semelhante ao que hoje se chama *Princípio de Hume*, que diz:

$$\#x F(x) = \#x G(x) \text{ sse } F \sim G$$

onde ‘ $F \sim G$ ’ abrevia que há tantos F quantos G , isto é, os F podem pôr-se em bijeção com os G , isto é:

$$\begin{aligned} \exists R(\forall v\forall y(Rvy \rightarrow (Fv \wedge Gy)) \wedge \\ \forall v(Fv \rightarrow \exists!y Rvy) \wedge \\ \forall y(Gy \rightarrow \exists!v Rvy)) \end{aligned}$$

Mas há uma diferença de tipo entre o Princípio de Hume e o Princípio de Cantor. No enunciado do Princípio de Cantor, as variáveis “ A ” e “ B ” são termos de primeira ordem que denotam *conjuntos*. No enunciado do Princípio de Hume, “ F ”, “ G ” e “ R ” *não* são termos de primeira ordem; estão em *posição de predicado*. (Talvez denotem *propriedades*.) Assim, o Princípio de Hume pode parafrasear-se em português: o número de F s é o número de G s sse os F s estão em bijeção com os G s. Chama-se *Princípio de Hume* porque Hume escreveu o seguinte:

Quando dois números se combinam de tal modo que a cada unidade de um corresponde sempre uma unidade do outro, dizemos que são iguais. (Hume, 1740, Pt.III Bk.1 §1)

E o Princípio de Hume foi trazido à proeminência matemático-lógica contemporânea por Frege (1884, §63), que citou esta passagem de Hume antes de (em efeito) esboçar (o que chamamos) Princípio de Hume.

Note a semelhança estrutural entre o Princípio de Hume e a Lei Básica V. A formulamos na Seção 65.6 do seguinte modo:

$$\epsilon x F(x) = \epsilon x G(x) \text{ sse } \forall x (F(x) \leftrightarrow G(x)).$$

E, neste ponto, alguns comentários e comparações podem ajudar.

Há duas maneiras de entender um princípio como o Princípio de Hume ou a Lei Básica V: *predicativamente* ou *impredicativamente* (recorde Seção 65.3). Na leitura impredicativa da Lei Básica V, para cada F , o objeto $\epsilon x F(x)$ pertence ao domínio de quantificação usado na própria formulação da Lei Básica V. Do mesmo modo, na leitura impredicativa do Princípio de Hume, para cada F , o objeto $\#x F(x)$ pertence ao domínio de quantificação usado na formulação do Princípio de Hume. Em contraste, na compreensão *predicativa*, os objetos $\epsilon x F(x)$ e $\#x F(x)$ seriam entidades de um domínio *diferente*.

Agora, se a Lei Básica V se lê impredicativamente, conduz à inconsistência, via Compreensão ingênua (para pormenores, ver Seção 65.6). Tal como a Compreensão ingênua, pode tornar-se consistente se se lê *predicativamente*. Mas provavelmente não fará tudo o que desejávamos.

O Princípio de Hume, porém, *pode* ser lido de modo consistente impredicativamente. E, assim lido, é bastante poderoso.

Para ilustrar: considere-se o predicado “ $x \neq x$ ”, que obviamente nada satisfaz. O Princípio de Hume fornece agora um objeto $\#x(x \neq x)$. Podemos tratá-lo como o número 0. Agora, na compreensão *impredicativa*—mas *só* na

impredicativa—esta entidade 0 pertence ao nosso domínio original de quantificação. Logo podemos aplicar sensatamente o Princípio de Hume com o predicado “ $x = 0$ ” para obter um objeto $\#x(x = 0)$. Podemos tratá-lo como o número 1. Além disso, o Princípio de Hume implica que $0 \neq 1$, pois não pode haver bijeção dos objetos não auto-idênticos para os objetos idênticos a 0 (não há dos primeiros, mas há um dos segundos). Agora, de novo impredicativamente, 1 pertence ao domínio original de quantificação. Logo podemos aplicar o Princípio de Hume com o predicado “ $(x = 0 \vee x = 1)$ ” para obter um objeto $\#x(x = 0 \vee x = 1)$. Podemos tratá-lo como o número 2, e mostrar que $0 \neq 2$ e $1 \neq 2$, e assim por diante.

Em suma, tomado impredicativamente, o Princípio de Hume implica que há *infinitamente muitos objetos*. E isto encorajou os *logicistas neo-fregeanos* a tomar o Princípio de Hume como fundação da aritmética.

O próprio Frege, contudo, não tomou o Princípio de Hume como fundação da aritmética. Em vez disso, Frege provou o Princípio de Hume a partir de uma definição explícita: $\#x F(x)$ é definido como a extensão do conceito $F \sim \Phi$. Em termos modernos, poder-se-ia tentar render isto como $\#x F(x) = \{G : F \sim G\}$; mas isto volta a arrastar-nos para os problemas da Compreensão ingênua.

Capítulo 72

Aritmética Cardinal

`content/set-theory/card-arithmetic/opps.tex`

72.1 Definindo as Operações Básicas

`sth:card-arithmetic:opps:`
`sec`

Como não precisamos de controlar a ordem, a aritmética cardinal é bastante mais fácil de definir do que a aritmética ordinal. Definiremos adição, multiplicação e exponenciação simultaneamente.

Definição 72.1. Quando $\mathfrak{a}$ e $\mathfrak{b}$ são cardinais:

$$\begin{aligned}\mathfrak{a} \oplus \mathfrak{b} &= |\mathfrak{a} \sqcup \mathfrak{b}| \\ \mathfrak{a} \otimes \mathfrak{b} &= |\mathfrak{a} \times \mathfrak{b}| \\ \mathfrak{a}^{\mathfrak{b}} &= |{}^{\mathfrak{b}}\mathfrak{a}|\end{aligned}$$

onde ${}^XY = \{f : f \text{ é uma função } X \rightarrow Y\}$. (É fácil mostrar que ${}^XY$ existe para quaisquer conjuntos X e Y ; deixamos isto como exercício.)

Problema 72.1. Prove em $\mathbf{ZF}$ que ${}^XY$ existe para quaisquer conjuntos X e Y . Trabalhando em $\mathbf{ZF}$, calcule $\text{posto}({}^XY)$ a partir de $\text{posto}(X)$ e $\text{posto}(Y)$, à maneira de [Lema 70.9](#).

Pode ajudar explicar esta definição. Quanto à adição: usa-se a noção de soma disjunta, $\sqcup$, tal como definida em [Definição 70.1](#); e é fácil ver que esta definição dá o resultado certo nos casos finitos. Quanto à multiplicação: [Proposição 1.27](#) nos diz que, se A tem n membros e B tem m membros, então $A \times B$ tem $n \cdot m$ membros, pelo que a nossa definição generaliza simplesmente a ideia à multiplicação transfinita. A exponenciação é semelhante: estamos simplesmente a generalizar o pensamento do finito para o transfinito. De fato, sob certos aspectos, a aritmética cardinal transfinita parece muito mais a aritmética “ordinária” do que a aritmética ordinal transfinita:

Proposição 72.2. $\oplus$ e $\otimes$ são comutativas e associativas.

*sth:card-arithmetic:opps:
cardplustimescommute*

Prova. Para a comutatividade, por [Lema 71.3](#) basta observar que $(\mathfrak{a} \sqcup \mathfrak{b}) \approx (\mathfrak{b} \sqcup \mathfrak{a})$ e $(\mathfrak{a} \times \mathfrak{b}) \approx (\mathfrak{b} \times \mathfrak{a})$. Deixamos a associatividade como exercício. $\square$

Problema 72.2. Prove que $\oplus$ e $\otimes$ são associativas.

Proposição 72.3. A é infinito se e só se $|A| \oplus 1 = 1 \oplus |A| = |A|$.

Prova. Como em [Teorema 71.7](#), a partir de [Lema 70.10](#) e [Lema 71.3](#). $\square$

Isto explica porque precisamos de símbolos distintos para adição/multiplicação ordinal versus cardinal: são operações genuinamente *diferentes*. Este par de resultados mostra que a exponenciação ordinal versus cardinal também são operações distintas. (Recordar que [Definição 66.7](#) implica que $2 = \{0, 1\}$):

Lema 72.4. $|\wp(A)| = 2^{|A|}$, para qualquer A .

*sth:card-arithmetic:opps:
lem:SizePowerset2Exp*

Prova. Para cada subconjunto $B \subseteq A$, seja $\chi_B \in {}^A2$ dado por:

$$\chi_B(x) = \begin{cases} 1 & \text{se } x \in B \\ 0 & \text{caso contrário.} \end{cases}$$

Seja agora $f(B) = \chi_B$; isto define uma [bijeção](#) $f: \wp(A) \rightarrow {}^A2$. Logo $|\wp(A)| \approx {}^A2$. Daí $|\wp(A)| \approx |{}^A2|$, pelo que $|\wp(A)| = |{}^A2| = 2^{|A|}$. $\square$

Esta prova sucinta subsume essencialmente a discussão em [Seção 4.13](#). Ali, mostramos como “reduzir” a não enumerabilidade de $\wp(\omega)$ à não enumerabilidade do conjunto das sequências binárias infinitas, $\mathbb{B}^\omega$. Na prática, $\mathbb{B}^\omega$ é apenas ${}^\omega 2$; e a prova precedente mostrou que o raciocínio que seguimos na [Seção 4.13](#) continua a valer usando qualquer conjunto A no lugar de ω . O resultado também fornece um fato rápido sobre exponenciação cardinal:

72.2. SIMPLIFICANDO ADIÇÃO E MULTIPLICAÇÃO

sth:card-arithmetic:opps:
cantorcor

Corolário 72.5. $\mathfrak{a} < 2^{\mathfrak{a}}$ para qualquer cardinal $\mathfrak{a}$.

Prova. Pelo Teorema de Cantor (Teorema 4.24) e Lema 72.4. $\square$

Logo $\omega < 2^\omega$. Mas nota: isto é um resultado sobre exponenciação *cardinal*. Deve contrastar-se com a exponenciação *ordinal*, pois neste último caso $\omega = 2^{(\omega)}$ (ver Seção 70.5).

Enquanto estamos no tema da exponenciação cardinal, podemos também ser um pouco mais precisos sobre a “maneira” pela qual $\mathbb{R}$ é não enumerável.

sth:card-arithmetic:opps:
continuumis2aleph0

Teorema 72.6. $|\mathbb{R}| = 2^\omega$

Esboço de prova. Há muitas maneiras de provar isto. A mais direta é argumentar que $\wp(\omega) \preceq \mathbb{R}$ e $\mathbb{R} \preceq \wp(\omega)$, e depois usar Schröder-Bernstein para inferir que $\mathbb{R} \approx \wp(\omega)$, e Lema 72.4 para inferir que $|\mathbb{R}| = 2^\omega$. Deixamos como exercício (esclarecedor) definir injecções $f: \wp(\omega) \rightarrow \mathbb{R}$ e $g: \mathbb{R} \rightarrow \wp(\omega)$. $\square$

Problema 72.3. Complete a prova de Teorema 72.6, mostrando que $\wp(\omega) \preceq \mathbb{R}$ e $\mathbb{R} \preceq \wp(\omega)$.

content/set-theory/card-arithmetic/simp.tex

72.2 Simplificando Adição e Multiplicação

sth:card-arithmetic:simp:
sec

Acontece que a adição e a multiplicação cardinais transfinitas são *extremamente* fáceis. Isto segue do fato de os cardinais serem (certos) ordinais, logo bem ordenados, e poderem ser manipulados de certa forma. Mostrar isto, porém, não é tão fácil. Para começar, precisamos de uma definição hábil:

Definição 72.7. Definimos uma *ordenação canónica*, $\triangleleft$, em pares de ordinais, estipulando que $\langle \alpha_1, \alpha_2 \rangle \triangleleft \langle \beta_1, \beta_2 \rangle$ se e só se se verifica uma das seguintes condições:

1. $\max(\alpha_1, \alpha_2) < \max(\beta_1, \beta_2)$; ou
2. $\max(\alpha_1, \alpha_2) = \max(\beta_1, \beta_2)$ e $\alpha_1 < \beta_1$; ou
3. $\max(\alpha_1, \alpha_2) = \max(\beta_1, \beta_2)$ e $\alpha_1 = \beta_1$ e $\alpha_2 < \beta_2$

Lema 72.8. $\langle \alpha \times \alpha, \triangleleft \rangle$ é uma boa ordem, para qualquer ordinal α .

Prova. Evidentemente $\triangleleft$ é conexa em $\alpha \times \alpha$. De fato, suponhamos que nem $\langle \alpha_1, \alpha_2 \rangle$ nem $\langle \beta_1, \beta_2 \rangle$ é menor do que a outra no sentido de $\triangleleft$. Então $\max(\alpha_1, \alpha_2) = \max(\beta_1, \beta_2)$ e $\alpha_1 = \beta_1$ e $\alpha_2 = \beta_2$, pelo que $\langle \alpha_1, \alpha_2 \rangle = \langle \beta_1, \beta_2 \rangle$.

Para mostrar que é boa ordem, seja $X \subseteq \alpha \times \alpha$ não vazio. Como α é um ordinal, algum δ é o menor membro de $\{\max(\gamma_1, \gamma_2) : \langle \gamma_1, \gamma_2 \rangle \in X\}$. Descartemos agora todos os pares de $\{\langle \gamma_1, \gamma_2 \rangle \in X : \max(\gamma_1, \gamma_2) = \delta\}$ exceto os de primeira coordenada mínima; entre estes, o par de segunda coordenada mínima é o elemento $\triangleleft$ -mínimo de X . $\square$

Segue-se uma observação mínima, mas simples:

Proposição 72.9. *Se $\alpha \approx \beta$, então $\alpha \times \alpha \approx \beta \times \beta$.*

*sth:card-arithmetic:simp:
simplecardproduct*

Prova. Basta deixar que $f: \alpha \rightarrow \beta$ induza $\langle \gamma_1, \gamma_2 \rangle \mapsto \langle f(\gamma_1), f(\gamma_2) \rangle$. $\square$

E agora aplicaremos tudo isto para provar um lema crucial:

Lema 72.10. *$\alpha \approx \alpha \times \alpha$, para qualquer ordinal infinito α*

*sth:card-arithmetic:simp:
alphatimesalpha*

Prova. Por redução ao absurdo, seja α o menor ordinal infinito para o qual isto falha. **Proposição 4.12** mostra que $\omega \approx \omega \times \omega$, logo $\omega \in \alpha$. Além disso, α é um cardinal: suponhamos o contrário, por redução ao absurdo; então $|\alpha| \in \alpha$, pelo que $|\alpha| \approx |\alpha| \times |\alpha|$, por hipótese; e $|\alpha| \approx \alpha$ por definição; logo $\alpha \approx \alpha \times \alpha$ por **Proposição 72.9**.

Ora, para cada $\langle \gamma_1, \gamma_2 \rangle \in \alpha \times \alpha$, consideremos o segmento:

$$\text{Seg}(\gamma_1, \gamma_2) = \{ \langle \delta_1, \delta_2 \rangle \in \alpha \times \alpha : \langle \delta_1, \delta_2 \rangle \triangleleft \langle \gamma_1, \gamma_2 \rangle \}$$

Seja $\gamma = \max(\gamma_1, \gamma_2)$; note-se que $\langle \gamma_1, \gamma_2 \rangle \triangleleft \langle \gamma + 1, \gamma + 1 \rangle$. Assim, quando γ é infinito, observe-se:

$$\begin{aligned} \text{Seg}(\gamma_1, \gamma_2) &\lesssim ((\gamma + 1) \cdot (\gamma + 1)) \\ &\approx (\gamma \cdot \gamma), \text{ por } \text{Lema 70.10} \text{ e } \text{Proposição 72.9} \\ &\approx \gamma, \text{ pela hipótese de indução} \\ &< \alpha, \text{ pois } \alpha \text{ é um cardinal} \end{aligned}$$

Logo $\text{ord}(\alpha \times \alpha, \triangleleft) \leq \alpha$, e portanto $\alpha \times \alpha \preceq \alpha$. Como obviamente $\alpha \preceq \alpha \times \alpha$, o resultado segue por Schröder-Bernstein. $\square$

Finalmente, chegamos ao nosso resultado simplificador:

Teorema 72.11. *Se $\mathfrak{a}, \mathfrak{b}$ são cardinais infinitos, então:*

*sth:card-arithmetic:simp:
cardplustimesmax*

$$\mathfrak{a} \otimes \mathfrak{b} = \mathfrak{a} \oplus \mathfrak{b} = \max(\mathfrak{a}, \mathfrak{b}).$$

Prova. Sem perda de generalidade, suponhamos $\mathfrak{a} = \max(\mathfrak{a}, \mathfrak{b})$. Então, invocando **Lema 72.10**, $\mathfrak{a} \otimes \mathfrak{a} = \mathfrak{a} \leq \mathfrak{a} \oplus \mathfrak{b} \leq \mathfrak{a} \oplus \mathfrak{a} \leq \mathfrak{a} \otimes \mathfrak{a}$. $\square$

Do mesmo modo, se $\mathfrak{a}$ é infinito, uma união de cardinalidade $\mathfrak{a}$ de conjuntos de cardinalidade $\leq \mathfrak{a}$ tem cardinalidade $\leq \mathfrak{a}$:

Proposição 72.12. *Seja $\mathfrak{a}$ um cardinal infinito. Para cada ordinal $\beta \in \mathfrak{a}$, seja X_β um conjunto com $|X_\beta| \leq \mathfrak{a}$. Então $\left| \bigcup_{\beta \in \mathfrak{a}} X_\beta \right| \leq \mathfrak{a}$.*

*sth:card-arithmetic:simp:
kappaunionkappasize*

Prova. Para cada $\beta \in \mathfrak{a}$, fixe uma injeção $f_\beta: X_\beta \rightarrow \mathfrak{a}$.¹ Defina uma injeção $g: \bigcup_{\beta \in \mathfrak{a}} X_\beta \rightarrow \mathfrak{a} \times \mathfrak{a}$ por $g(v) = \langle \beta, f_\beta(v) \rangle$, onde $v \in X_\beta$ e $v \notin X_\gamma$ para qualquer $\gamma \in \beta$. Ora, $\bigcup_{\beta \in \mathfrak{a}} X_\beta \preceq \mathfrak{a} \times \mathfrak{a} \approx \mathfrak{a}$ por **Teorema 72.11**. $\square$

content/set-theory/card-arithmetic/expotough.tex

¹Como são estes “fixos”? Ver **Seção 73.5**.

72.3. ALGUMAS SIMPLIFICAÇÕES

72.3 Algumas Simplificações com Exponenciação Cardinal

Embora definir $\triangleleft$ fosse um pouco trabalhoso, o resultado é um teorema útil sobre adição e multiplicação cardinais, **Teorema 72.11**. A exponenciação transfinita, porém, não se simplifica tão diretamente. Para explicar por que, começamos por um resultado que prolonga um padrão familiar do caso finitário (embora a sua prova esteja a um nível elevado de abstração):

Proposição 72.13. $\mathfrak{a}^{\mathfrak{b} \oplus \mathfrak{c}} = \mathfrak{a}^{\mathfrak{b}} \otimes \mathfrak{a}^{\mathfrak{c}}$ e $(\mathfrak{a}^{\mathfrak{b}})^{\mathfrak{c}} = \mathfrak{a}^{\mathfrak{b} \otimes \mathfrak{c}}$, para quaisquer cardinais $\mathfrak{a}, \mathfrak{b}, \mathfrak{c}$.

Prova. Para a primeira afirmação, considere-se uma função $f: (\mathfrak{b} \sqcup \mathfrak{c}) \rightarrow \mathfrak{a}$. Ora, “parta-se” isto, definindo $f_{\mathfrak{b}}(\beta) = f(\beta, 0)$ para cada $\beta \in \mathfrak{b}$, e $f_{\mathfrak{c}}(\gamma) = f(\gamma, 1)$ para cada $\gamma \in \mathfrak{c}$. A aplicação $f \mapsto (f_{\mathfrak{b}} \times f_{\mathfrak{c}})$ é uma bijeção ${}^{\mathfrak{b} \sqcup \mathfrak{c}}\mathfrak{a} \rightarrow ({}^{\mathfrak{b}}\mathfrak{a} \times {}^{\mathfrak{c}}\mathfrak{a})$.

Para a segunda afirmação, considere-se uma função $f: \mathfrak{c} \rightarrow ({}^{\mathfrak{b}}\mathfrak{a})$; assim, para cada $\gamma \in \mathfrak{c}$ temos alguma função $f(\gamma): \mathfrak{b} \rightarrow \mathfrak{a}$. Ora, defina-se $f^*(\beta, \gamma) = (f(\gamma))(\beta)$ para cada $\langle \beta, \gamma \rangle \in \mathfrak{b} \times \mathfrak{c}$. A aplicação $f \mapsto f^*$ é uma bijeção ${}^{\mathfrak{b} \otimes \mathfrak{c}}\mathfrak{a} \rightarrow {}^{\mathfrak{c}}({}^{\mathfrak{b}}\mathfrak{a})$. $\square$

O que gostaríamos *mesmo* era de uma maneira fácil de calcular $\mathfrak{a}^{\mathfrak{b}}$ quando tratamos de cardinais infinitos. Eis um passo útil nessa direção:

Proposição 72.14. Se $2 \leq \mathfrak{a} \leq \mathfrak{b}$ e $\mathfrak{b}$ é infinito, então $\mathfrak{a}^{\mathfrak{b}} = 2^{\mathfrak{b}}$

Prova.

$$\begin{aligned} 2^{\mathfrak{b}} &\leq \mathfrak{a}^{\mathfrak{b}}, \text{ pois } 2 \leq \mathfrak{a} \\ &\leq (2^{\mathfrak{a}})^{\mathfrak{b}}, \text{ por Lema 72.4} \\ &= 2^{\mathfrak{a} \otimes \mathfrak{b}}, \text{ por Proposição 72.13} \\ &= 2^{\mathfrak{b}}, \text{ por Teorema 72.11} \end{aligned} \quad \square$$

Não devemos esperar poder simplificar mais além disto, pois $\mathfrak{b} < 2^{\mathfrak{b}}$ por **Lema 72.4**. Contudo, isto não nos diz o que afirmar sobre $\mathfrak{a}^{\mathfrak{b}}$ quando $\mathfrak{b} < \mathfrak{a}$. Naturalmente, se $\mathfrak{b}$ é finito, sabemos o que fazer.

Proposição 72.15. Se $\mathfrak{a}$ é infinito e $n \in \omega$, então $\mathfrak{a}^n = \mathfrak{a}$

Prova. $\mathfrak{a}^n = \mathfrak{a} \otimes \mathfrak{a} \otimes \dots \otimes \mathfrak{a} = \mathfrak{a}$, por **Teorema 72.11**. $\square$

Adicionalmente, nalguns outros casos, podemos controlar o tamanho de $\mathfrak{a}^{\mathfrak{b}}$:

Proposição 72.16. Se $2 \leq \mathfrak{b} < \mathfrak{a} \leq 2^{\mathfrak{b}}$ e $\mathfrak{b}$ é infinito, então $\mathfrak{a}^{\mathfrak{b}} = 2^{\mathfrak{b}}$

Prova. $2^{\mathfrak{b}} \leq \mathfrak{a}^{\mathfrak{b}} \leq (2^{\mathfrak{b}})^{\mathfrak{b}} = 2^{\mathfrak{b} \otimes \mathfrak{b}} = 2^{\mathfrak{b}}$, raciocinando como na **Proposição 72.14**. $\square$

Para além deste ponto, as coisas tornam-se bastante mais subtis.

72.4 A Hipótese do Contínuo

O resultado anterior sugere (corretamente) que a exponenciação cardinal seria bastante *fácil*, se os cardinais infinitos estivessem garantidos a “comportar-se de forma simples” com potências de 2, isto é, (por [Lema 72.4](#)) com a operação de tomar conjuntos potência. Mas não podemos supor que os cardinais infinitos *se comportem* de forma simples com os conjuntos potência.

Começando a desdobrar isto, introduzimos uma notação conveniente.

Definição 72.17. Onde $\aleph^\oplus$ é o menor cardinal estritamente maior do que $\aleph$, definimos duas sequências infinitas:

$$\begin{aligned} \aleph_0 &= \omega & \beth_0 &= \omega \\ \aleph_{\alpha+1} &= (\aleph_\alpha)^\oplus & \beth_{\alpha+1} &= 2^{\beth_\alpha} \\ \aleph_\alpha &= \bigcup_{\beta < \alpha} \aleph_\beta & \beth_\alpha &= \bigcup_{\beta < \alpha} \beth_\beta \quad \text{quando } \alpha \text{ é um ordinal limite.} \end{aligned}$$

A definição de $\aleph^\oplus$ faz sentido, pois [Teorema 71.14](#) nos diz que, para cada cardinal $\aleph$, existe algum cardinal maior do que $\aleph$, e a Indução Transfinita garante que existe um cardinal *mínimo* maior do que $\aleph$. O resto da definição é fornecido pela recursão transfinita.

Cantor introduziu esta notação “ $\aleph$ ”; trata-se do *alef*, a primeira letra do alfabeto hebraico e a primeira letra da palavra hebraica para “infinito”. Peirce introduziu a notação “ $\beth$ ”; trata-se do *beth*, a segunda letra do alfabeto hebraico.² Estas notações fornecem-nos cardinais infinitos.

Proposição 72.18. $\aleph_\alpha$ e $\beth_\alpha$ são cardinais, para todo o ordinal α .

Prova. Ambos os resultados valem por uma indução transfinita simples. $\aleph_0 = \beth_0 = \omega$ é um cardinal por [Corolário 71.9](#). Supondo que $\aleph_\alpha$ e $\beth_\alpha$ são ambos cardinais, $\aleph_{\alpha+1}$ e $\beth_{\alpha+1}$ estão explicitamente definidos como cardinais. E a união de um conjunto de cardinais é um cardinal, por [Proposição 71.13](#). $\square$

Além disso, todo o cardinal infinito é um $\aleph$:

Proposição 72.19. Se $\aleph$ é um cardinal infinito, então $\aleph = \aleph_\gamma$ para um certo γ único.

Prova. Por indução transfinita nos cardinais. Para a indução, suponhamos que se $\mathfrak{b} < \aleph$ então $\mathfrak{b} = \aleph_{\gamma_{\mathfrak{b}}}$. Se $\aleph = \mathfrak{b}^\oplus$ para algum $\mathfrak{b}$, então $\aleph = (\aleph_{\gamma_{\mathfrak{b}}})^\oplus = \aleph_{\gamma_{\mathfrak{b}}+1}$. Se $\aleph$ não é o sucessor de nenhum cardinal, então, por serem os cardinais ordinais, $\aleph = \bigcup_{\mathfrak{b} < \aleph} \mathfrak{b} = \bigcup_{\mathfrak{b} < \aleph} \aleph_{\gamma_{\mathfrak{b}}}$, logo $\aleph = \aleph_\gamma$ onde $\gamma = \bigcup_{\mathfrak{b} < \aleph} \gamma_{\mathfrak{b}}$. $\square$

²Peirce usou esta notação numa carta a Cantor de dezembro de 1900. Infelizmente, Peirce também apresentou ali um mau argumento de que $\beth_\alpha$ não existe para $\alpha \geq \omega$.

72.4. A HIPÓTESE DO CONTÍNUO

Como todo o cardinal infinito é um $\aleph$, surge naturalmente a pergunta: será que todo o cardinal infinito é um $\beth$? Certamente que, se isso *se verificasse*, os cardinais infinitos “comportar-se-iam de forma simples” com a operação de tomar conjuntos potência. De fato, teríamos o seguinte:

Hipótese Generalizada do Contínuo (HGC). $\aleph_\alpha = \beth_\alpha$, para todo o α .

Além disso, se a HGC valesse, poderíamos simplificar bastante a exponenciação cardinal. Em particular, poderíamos mostrar que, quando $\mathfrak{b} < \mathfrak{a}$, o valor de $\mathfrak{a}^{\mathfrak{b}}$ fica encerrado entre $\mathfrak{a} \leq \mathfrak{a}^{\mathfrak{b}} \leq \mathfrak{a}^{\oplus}$. Poderíamos então dar condições precisas que determinam qual das duas possibilidades se verifica (isto é, se $\mathfrak{a} = \mathfrak{a}^{\mathfrak{b}}$ ou $\mathfrak{a}^{\mathfrak{b}} = \mathfrak{a}^{\oplus}$).³

Mas a HGC é uma *hipótese*, não um *teorema*. De fato, Gödel (1938) provou que, se **ZFC** for consistente, então também o é **ZFC** + HGC. Mas mais tarde verificou-se que podemos acrescentar igualmente $\neg$ HGC a **ZFC**. De fato, considere-se a *instância* não trivial mais simples da HGC, nomeadamente:

Hipótese do Contínuo (HC). $\aleph_1 = \beth_1$.

Cohen (1963) provou que, se **ZFC** for consistente, então também o é **ZFC** + $\neg$ HC. Logo a Hipótese do Contínuo é independente de **ZFC**.

A Hipótese do Contínuo chama-se assim porque “o contínuo” é outro nome para a reta real, $\mathbb{R}$. Teorema 72.6 nos diz que $|\mathbb{R}| = \beth_1$. Assim, a Hipótese do Contínuo afirma que não há cardinal entre a cardinalidade dos números naturais, $\aleph_0 = \beth_0$, e a cardinalidade do contínuo, $\beth_1$.

Dada a *independência* da (H)HC relativamente a **ZFC**, o que devemos dizer sobre a sua *verdade*? Há *muito* a dizer. De fato, muito trabalho recente e fecundo em teoria dos conjuntos tem sido dedicado a estas questões. Mas dois pontos rápidos valem certamente a pena enfatizar.

Primeiro: não se segue *imediatamente* destes resultados formais de independência que a HGC ou a HC sejam *indeterminadas* no valor de verdade. Afinal, talvez bastem mais axiomas, que nos pareçam naturais, para decidir a questão num sentido ou noutro. O próprio Gödel sugeriu que essa era a resposta adequada.

Segundo: a independência da HC relativamente a **ZFC** é certamente *notável*, mas não é *inacreditável* (no sentido literal). O ponto é simplesmente que, para tudo o que **ZFC** nos diz, passar de um cardinal ao seu sucessor pode envolver uma ferramenta menos direta do que simplesmente tomar conjuntos potência.

Feitas estas duas observações, quem quiser saber mais terá de recorrer aos vários filósofos e matemáticos com cavalos na corrida.⁴

content/set-theory/card-arithmetic/fix.tex

³A condição é ditada pela *cofinalidade*.

⁴Embora possa querer começar por ler Potter (2004, §15.6).

72.5 Pontos Fixos de $\aleph$

No [Capítulo 68](#), sugerimos que o Esquema de Substituição carece de justificação, pois força a hierarquia a ser bastante alta. Tendo feito alguma aritmética cardinal, podemos dar uma pequena ilustração da altura da hierarquia.

sth:card-arithmetic:fix:
sec

Evidentemente $0 < \aleph_0$, e $1 < \aleph_1$, e $2 < \aleph_2 \dots$ e, de fato, a diferença de tamanho só fica *maior* a cada passo. Assim, é tentador conjecturar que $\kappa < \aleph_\kappa$ para todo o ordinal κ .

Mas esta conjectura é *falsa*, em **ZFC**. De fato, podemos provar que existem *pontos fixos* de $\aleph$, isto é, cardinais κ tais que $\kappa = \aleph_\kappa$.

Proposição 72.20. *Existe um ponto fixo de $\aleph$.*

sth:card-arithmetic:fix:
alephfixed

Prova. Por recursão, defina-se:

$$\begin{aligned}\kappa_0 &= 0 \\ \kappa_{n+1} &= \aleph_{\kappa_n} \\ \kappa &= \bigcup_{n < \omega} \kappa_n\end{aligned}$$

Ora, κ é um cardinal por [Proposição 71.13](#). Mas então:

$$\kappa = \bigcup_{n < \omega} \kappa_{n+1} = \bigcup_{n < \omega} \aleph_{\kappa_n} = \bigcup_{\alpha < \kappa} \aleph_\alpha = \aleph_\kappa$$

□

Boolos escreveu um artigo sobre precisamente o ponto fixo de $\aleph$ que acabamos de construir. Depois de notar a existência de κ , no início do artigo, disse:

[κ é] um número *bastante grande*, aos olhos de quem não teve contato prévio com a teoria dos conjuntos, tão grande, me parece, que põe em causa a verdade de qualquer teoria que, entre as suas afirmações, inclua a de que existem pelo menos κ objetos. ([Boolos, 2000](#), p. 257)

E concluiu o artigo perguntando:

[será que] suspeitamos que, seja qual for o começo da história, quando chegamos tão longe as rodas giram no vazio e já não estamos a ouvir uma descrição de coisa alguma que seja o caso? ([Boolos, 2000](#), p. 268)

Se, de fato, ultrapassamos “qualquer coisa que seja o caso”, então devemos apontar o dedo da culpa diretamente ao Esquema de Substituição. Pois é este axioma que permite a nossa prova funcionar. Nesse caso, supõe-se, Boolos teria de rever a afirmação que fizera, décadas antes, de que a Substituição não tem consequências “indesejáveis” (ver [Seção 69.3](#)).

72.5. PONTOS FIXOS DE $\aleph$

Mas será a existência de κ assim tão má? Pode ajudar, aqui, considerar o *paradoxo de Tristram Shandy* de Russell. Tristram Shandy documenta a sua vida no diário, mas demora um ano a registrar um único dia. A cada ano que passa, Tristram fica cada vez mais atrasado: passado um ano, só registrou um dia, e viveu 364 dias por registrar; passados dois anos, só registrou dois dias, e viveu 728 dias por registrar; passados três anos, só registrou três dias, e viveu 1092 dias por registrar ...⁵ Ainda assim, se Tristram é *imortal*, conseguirá registrar todos os dias, pois registrará o n -ésimo dia no n -ésimo ano da sua vida. E assim, “no fim dos tempos”, Tristram terá um diário completo.

Porque é que isto difere tanto da ideia de que α é menor do que $\aleph_\alpha$ —e, de fato, cada vez mais, desesperadamente menor—até κ , onde nos “apanhamos” e $\kappa = \aleph_\kappa$?

Deixando isto de lado, e supondo que aceitamos **ZFC**, fechemos com um pouco mais de diversão sobre construções de pontos fixos. Os três resultados seguintes estabelecem, intuitivamente, que existe um ponto (não trivial) em que a hierarquia é tão larga como alta:

*sth:card-arithmetic:fix:
bethfixed*

Proposição 72.21. *Existe um ponto fixo de $\beth$, isto é, um κ tal que $\kappa = \beth_\kappa$.*

Prova. Como na **Proposição 72.20**, usando “ $\beth$ ” no lugar de “ $\aleph$ ”. $\square$

*sth:card-arithmetic:fix:
stagesize*

Proposição 72.22. $|V_{\omega+\alpha}| = \beth_\alpha$. Se $\omega \cdot \omega \leq \alpha$, então $|V_\alpha| = \beth_\alpha$.

Prova. A primeira afirmação vale por uma indução transfinita simples. A segunda segue-se, pois se $\omega \cdot \omega \leq \alpha$ então $\omega + \alpha = \alpha$. Para o estabelecer, usamos fatos sobre aritmética ordinal de **Capítulo 70**. Note-se primeiro que $\omega \cdot \omega = \omega \cdot (1 + \omega) = (\omega \cdot 1) + (\omega \cdot \omega) = \omega + (\omega \cdot \omega)$. Ora, se $\omega \cdot \omega \leq \alpha$, isto é, $\alpha = (\omega \cdot \omega) + \beta$ para algum β , então $\omega + \alpha = \omega + ((\omega \cdot \omega) + \beta) = (\omega + (\omega \cdot \omega)) + \beta = (\omega \cdot \omega) + \beta = \alpha$. $\square$

Corolário 72.23. *Existe um κ tal que $|V_\kappa| = \kappa$.*

Prova. Seja κ um ponto fixo de $\beth$, dado por **Proposição 72.21**. Claramente $\omega \cdot \omega < \kappa$. Logo $|V_\kappa| = \beth_\kappa = \kappa$ por **Proposição 72.22**. $\square$

Há tantos estágios abaixo de V_κ quantos **membros** de V_κ . Intuitivamente, então, V_κ é tão largo como alto. Isto é muito à maneira de Tristram Shandy: passamos de um estágio ao seguinte tomando *conjuntos potência*, tornando a hierarquia *muito* maior a cada passo. Mas “no fim”, isto é, no estágio κ , a largura da hierarquia alcança a altura.

Alguém pode perguntar: *Com que frequência a largura da hierarquia coincide com a altura?* A resposta é: *Tantas vezes quantos os ordinais*. Mas isto precisa de uma pequena explicação.

⁵Ignorando anos bissextos.

Definimos um termo τ do seguinte modo. Para qualquer A , seja:

$$\begin{aligned}\tau_0(A) &= |A| \\ \tau_{n+1}(A) &= \beth_{\tau_n(A)} \\ \tau(A) &= \bigcup_{n < \omega} \tau_n(A)\end{aligned}$$

Como na [Proposição 72.21](#), $\tau(A)$ é um ponto fixo de $\beth$ para qualquer A , e trivialmente $|A| < \tau(A)$. Considere-se então esta definição recursiva:

$$\begin{aligned}W_0 &= 0 \\ W_{\alpha+1} &= \tau(W_\alpha) \\ W_\alpha &= \bigcup_{\beta < \alpha} W_\beta, \text{ quando } \alpha \text{ é limite}\end{aligned}$$

A construção está definida para todos os ordinais. Intuitivamente, então, W é “uma injeção” dos ordinais nos pontos fixos de $\beth$. E, exatamente como antes, V_{W_α} é tão largo como alto, para qualquer α .

Capítulo 73

Escolha

[content/set-theory/choice/introduction.tex](#)

73.1 Introdução

Em [Capítulos 71 a 72](#), desenvolvemos uma teoria de cardinais tratando os cardinais como ordinais. Essa abordagem depende do Axioma da Boa Ordenação. Verifica-se que a Boa Ordenação é equivalente a outro princípio—o Axioma da Escolha—e houve discussão filosófica séria sobre a sua aceitabilidade. As questões deste capítulo são: Como é que o Axioma é usado, e pode ser justificado?

[sth:choice:intro:sec](#)

[content/set-theory/choice/tarskiscott.tex](#)

73.2 O Truque de Tarski–Scott

sth:choice:tarskiscott:
sec

Na **Definição 71.1**, definimos cardinais como ordinais. Para isso, assumimos o Axioma da Boa Ordenação. Fizemo-lo apenas porque é o “padrão da indústria”.

Antes de discutir questões filosóficas em torno da Boa Ordenação, importa deixar claro que *podemos* afastar-nos desse padrão e desenvolver uma teoria de cardinais *sem* assumir a Boa Ordenação. Continuamos a poder usar as definições de $A \approx B$, $A \preceq B$ e $A \prec B$, tal como apareceram no **Capítulo 4**. Só precisamos de uma nova noção de *cardinal*.

Uma ideia ingênua seria tentar definir a cardinalidade de A assim:

$$\{x : A \approx x\}.$$

Talvez queira comparar isto com a definição fregeana de $\#xFx$, esboçada no final de **Seção 71.5**. E, pelas razões que ali adumbramos, esta definição falha. Qualquer conjunto singular é equipotente com $\{\emptyset\}$. Mas novos singulares são formados em cada estágio sucessor da hierarquia (basta considerar o singular do estágio anterior). Logo $\{x : A \approx x\}$ não existe, pois não pode ter posto.

Para contornar o problema, usamos um truque devido a Tarski e Scott:¹

Definição 73.1 (Tarski–Scott). Para qualquer fórmula $\varphi(x)$, seja $[x : \varphi(x)]$ o conjunto de todos os x , de posto mínimo possível, tais que $\varphi(x)$ (ou $\emptyset$, se não houver φ s).

Devemos verificar que esta definição é legítima. Trabalhando em **ZF**, **Teorema 68.13** garante que $\text{posto}(x)$ existe para todo o x . Agora, se houver entidades que satisfazem φ , podemos deixar α ser o posto mínimo tal que $(\exists x \subseteq V_\alpha) \varphi(x)$, isto é, $(\forall \beta \in \alpha)(\forall x \subseteq V_\beta) \neg \varphi(x)$. Podemos então definir $[x : \varphi(x)]$ por Separação como $\{x \in V_{\alpha+1} : \varphi(x)\}$.

Tendo justificado o truque de Tarski–Scott, podemos usá-lo para definir uma noção de cardinalidade:

Definição 73.2. A TS-cardinalidade de A é $\text{tsc}(A) = [x : A \approx x]$.

A definição de TS-cardinal não usa a Boa Ordenação. Mas, mesmo sem esse Axioma, podemos mostrar que os TS-*cardinais* se comportam de modo semelhante aos *cardinais* tal como definidos na **Definição 71.1**. Por exemplo, se reformularmos **Lema 71.3** e **Lema 72.4** em termos de TS-cardinais, as provas funcionam em **ZF**, sem assumir a Boa Ordenação.

Enquanto estamos no assunto, vale a pena notar que também podemos desenvolver uma teoria de ordinais usando o truque de Tarski–Scott. Onde $\langle A, < \rangle$ é uma boa ordenação, seja $\text{tso}(A, <) = [\langle X, R \rangle : \langle A, < \rangle \cong \langle X, R \rangle]$. Para mais sobre este tratamento de cardinais e ordinais, ver **Potter (2004, chs. 9–12)**.

content/set-theory/choice/hartogs.tex

¹Lembrete: todas as fórmulas podem ter parâmetros (salvo indicação em contrário).

73.3 Comparabilidade e Lema de Hartogs

Isto é o lado positivo. Eis o lado negativo. Sem Escolha, as coisas ficam complicadas. Para ver por que, eis um belo resultado devido a Hartogs (1915):

Lema 73.3 (em ZF). Para qualquer conjunto A , existe um ordinal α tal que $\alpha \not\preceq A$

Prova. Se $B \subseteq A$ e $R \subseteq B^2$, então $\langle B, R \rangle \subseteq V_{\text{posto}(A)+4}$ por Lema 70.9. Logo, usando Separação, considere-se:

$$C = \{\langle B, R \rangle \in V_{\text{posto}(A)+5} : B \subseteq A \text{ e } \langle B, R \rangle \text{ é uma boa ordenação}\}$$

Usando Substituição e Teorema 67.26, forme-se o conjunto:

$$\alpha = \{\text{ord}(B, R) : \langle B, R \rangle \in C\}.$$

Por Corolário 67.19, α é um ordinal, pois é um conjunto transitivo de ordinais. De fato, se $\gamma \in \beta \in \alpha$, então $\beta = \text{ord}(B, R)$ para algum $B \subseteq R$, donde $\gamma = \text{ord}(B_b, R_b)$ para algum $b \in B$ por Lema 67.10, logo $\gamma \in \alpha$.

Por redução ao absurdo, suponhamos que existe uma injeção $f: \alpha \rightarrow A$. Então, onde:

$$\begin{aligned} B &= \text{Im}(f) \\ R &= \{\langle f(\alpha), f(\beta) \rangle \in A \times A : \alpha \in \beta\}. \end{aligned}$$

Claramente $\alpha = \text{ord}(B, R)$ e $\langle B, R \rangle \in C$. Logo $\alpha \in \alpha$, o que é contradição. $\square$

Isto implica um resultado profundo:

Teorema 73.4 (em ZF). As afirmações seguintes são equivalentes:

1. O Axioma da Boa Ordenação
2. Ou $A \preceq B$ ou $B \preceq A$, para quaisquer conjuntos A e B

Prova. (1) $\Rightarrow$ (2). Fixem-se A e B . Invocando (1), existem boas ordenações $\langle A, R \rangle$ e $\langle B, S \rangle$. Invocando Teorema 67.26, sejam $f: \alpha \rightarrow \langle A, R \rangle$ e $g: \beta \rightarrow \langle B, S \rangle$ isomorfismos. Por Proposição 67.22, ou $\alpha \subseteq \beta$ ou $\beta \subseteq \alpha$. Se $\alpha \subseteq \beta$, então $g \circ f^{-1}: A \rightarrow B$ é uma injeção, logo $A \preceq B$; analogamente, se $\beta \subseteq \alpha$ então $B \preceq A$.

(2) $\Rightarrow$ (1). Fixe-se A ; por Lema 73.3 existe algum ordinal β tal que $\beta \not\preceq A$. Invocando (2), temos $A \preceq \beta$. Logo existe alguma injeção $f: A \rightarrow \beta$, e podemos usar esta injeção para bem ordenar os elementos de A , definindo uma ordem $\{\langle a, b \rangle \in A \times A : f(a) \in f(b)\}$. $\square$

Como consequência imediata: se a Boa Ordenação falha, então alguns conjuntos são *literalmente incomparáveis* quanto ao tamanho. Assim, se a Boa

73.4. O PROBLEMA DA BOA ORDENAÇÃO

Ordenação falha, a aritmética cardinal transfinita ficará complicada. Por exemplo, teremos de abandonar a ideia de que, se A e B são infinitos, então $A \sqcup B \approx A \times B \approx M$, onde M é o maior de A e B (ver [Teorema 72.11](#)). O problema é simples: se não podemos *comparar* o tamanho de A e B , então não faz sentido perguntar qual é o maior.

[content/set-theory/choice/wellorderingproblem.tex](#)

73.4 O Problema da Boa Ordenação

[sth:choice:woproblem:](#)
[sec](#)

Evidentemente que muito depende de aceitarmos ou não a Boa Ordenação. Mas a discussão deste princípio tendeu a concentrar-se num equivalente, o Axioma da Escolha. Voltemos então a atenção para esse axioma (e provemos a equivalência).

Em 1883, Cantor exprimiu o seu apoio ao Axioma da Boa Ordenação, chamando-lhe “uma lei do pensamento que me parece fundamental, rica nas suas consequências e particularmente notável pela sua generalidade” (citado em [Potter 2004](#), p. 243). Mas Cantor acabou por convencer-se de que o “Axioma” precisava de prova. A comunidade matemática também.

O problema foi “resolvido” por Zermelo em 1904. Para explicar a solução, precisamos de algumas definições.

Definição 73.5. Uma função f é uma *função de escolha* sse $f(x) \in x$ para todo o $x \in \text{dom}(f)$. Dizemos que f é uma *função de escolha para* A sse f é uma função de escolha com $\text{dom}(f) = A \setminus \{\emptyset\}$.

Intuitivamente, para cada conjunto (não vazio) $x \in A$, uma função de escolha para A *escolhe* um elemento particular, $f(x)$, de x . O Axioma da Escolha é então:

Escolha (Escolha). Todo o conjunto tem uma função de escolha.

Zermelo mostrou que a Escolha implica a boa ordenação, e vice-versa:

[sth:choice:woproblem:](#)
[thmwochoice](#)

Teorema 73.6 (em ZF). *A Boa Ordenação e a Escolha são equivalentes.*

Prova. Da esquerda para a direita. Seja A um conjunto de conjuntos. Então $\bigcup A$ existe pelo Axioma da União, e logo, pela Boa Ordenação, existe alguma relação $<$ que bem ordena $\bigcup A$. Seja agora $f(x) =$ o membro $<$ -mínimo de x . Isto é uma função de escolha para A .

Da direita para a esquerda. Fixe-se A . Pela Escolha, existe uma função de escolha, f , para $\wp(A) \setminus \{\emptyset\}$. Usando Recursão Transfinita, defina-se uma função:

$$g(0) = f(A)$$
$$g(\alpha) = \begin{cases} \text{parar!} & \text{se } A = g[\alpha] \\ f(A \setminus g[\alpha]) & \text{caso contrário} \end{cases}$$

A indicação “parar!” é apenas uma abreviatura para o que seria de outro modo uma definição mais longa. Ou seja, quando $A = g[\alpha]$ pela primeira vez, seja $g(\delta) = A$ para todo o $\delta \leq \alpha$. Agora, em primeiro lugar, só podemos ter a certeza de que isto define um *termo* (ver as observações após [Teorema 68.4](#)); mas mostraremos que de fato temos uma função.

Como f é função de escolha, para cada α (quando definido) temos $g(\alpha) = f(A \setminus g[\alpha]) \in A \setminus g[\alpha]$; isto é, $g(\alpha) \notin g[\alpha]$. Logo, se $g(\alpha) = g(\beta)$ então $g(\beta) \notin g[\alpha]$, isto é, $\beta \notin \alpha$, e analogamente $\alpha \notin \beta$. Logo $\alpha = \beta$, por Tricotomia. Assim, g é *injetiva*.

Observe-se a seguir que de fato paramos!, isto é, que existe algum ordinal mínimo α tal que $A = g[\alpha]$. De fato, suponhamos o contrário; então, como g é *injetiva*, teríamos $\alpha \prec \wp(A) \setminus \{\emptyset\}$ para todo o ordinal α , contradizendo [Lema 73.3](#). Daí também $\text{Im}(g) = A$.

Reunindo estes fatos, g é uma *bijecção* de algum ordinal para A . Agora, g pode ser usada para bem ordenar A . □

Assim, a Boa Ordenação e a Escolha estão de pé ou caem juntas. Mas a questão permanece: estão de pé ou caem?

[content/set-theory/choice/countablechoice.tex](#)

73.5 Escolha Enumerável

É fácil provar, sem usar Escolha/Boa Ordenação, que:

sth:choice:countablechoice:
sec

Lema 73.7 (em $\mathbf{Z}^-$). *Todo o conjunto finito tem uma função de escolha.*

Prova. Seja $a = \{b_1, \dots, b_n\}$. Suponhamos, por simplicidade, que cada $b_i \neq \emptyset$. Logo existem objetos $c_1, \dots, c_n$ tais que $c_1 \in b_1, \dots, c_n \in b_n$. Agora, por [Proposição 66.5](#), o conjunto $\{\langle b_1, c_1 \rangle, \dots, \langle b_n, c_n \rangle\}$ existe; e isto é uma função de escolha para a . □

Mas a situação complica-se logo que consideramos conjuntos infinitos. Por exemplo, considere-se esta extensão “mínima” do resultado acima:

Escolha Enumerável. Todo o conjunto *enumerável* tem uma função de escolha.

Isto é um caso particular da Escolha. E verifica-se que este princípio foi invocado com alguma frequência, sem uma consciência óbvia do seu uso. Eis dois bons exemplos.²

²Devidos a [Potter \(2004, §9.4\)](#) e Luca Incurvati.

73.5. ESCOLHA ENUMERÁVEL

Exemplo 73.8. Eis um pensamento natural: para qualquer conjunto A , ou $\omega \preceq A$, ou $A \approx n$ para algum $n \in \omega$. Esta é uma maneira de exprimir a ideia intuitiva de que todo o conjunto é finito ou infinito. Cantor, e muitos outros matemáticos, fizeram esta afirmação sem a provar. Com a cautela que nos caracteriza, a provamos em **Teorema 71.7**. Mas nessa prova estávamos a trabalhar em **ZFC**, pois assumíamos que qualquer conjunto A pode ser bem ordenado, e logo que $|A|$ está garantidamente definido. Ou seja: assumimos explicitamente a Escolha.

De fato, **Dedekind (1888)** ofereceu a sua própria prova desta afirmação, do seguinte modo:

Teorema 73.9 (em $\mathbf{Z}^- + \text{Escolha Enumerável}$). *Para qualquer A , ou $\omega \preceq A$ ou $A \approx n$ para algum $n \in \omega$.*

Prova. Suponhamos $A \not\approx n$ para todo o $n \in \omega$. Então, em particular, para cada $n < \omega$ existe um subconjunto $A_n \subseteq A$ com exatamente 2^n elementos. Usando esta sequência $A_0, A_1, A_2, \dots$, definimos para cada n :

$$B_n = A_n \setminus \bigcup_{i < n} A_i.$$

Note-se o seguinte

$$\begin{aligned} \left| \bigcup_{i < n} A_i \right| &\leq |A_0| + |A_1| + \dots + |A_{n-1}| \\ &= 1 + 2 + \dots + 2^{n-1} \\ &= 2^n - 1 \\ &< 2^n = |A_n| \end{aligned}$$

Logo cada B_n tem pelo menos um membro, c_n . Além disso, os B_n são dois a dois disjuntos; assim, se $c_n = c_m$ então $n = m$. Mas cada $c_n \in A$. Logo a função $f(n) = c_n$ é uma injeção $\omega \rightarrow A$. $\square$

Dedekind não assinalou que tinha usado a Escolha Enumerável. Mas reparou o leitor no seu uso? Olhe de novo. (De fato: *olhe de novo*.)

A prova usou a Escolha Enumerável duas vezes. A usamos uma vez, para obter a nossa sequência de conjuntos $A_0, A_1, A_2, \dots$. Voltamos a usá-la para seleccionar os elementos c_n de cada B_n . Além disso, este uso da Escolha é irreduzível. **Cohen (1966, p. 138)** provou que o resultado falha se não tivermos qualquer versão da Escolha. Ou seja: é consistente com **ZF** que existam conjuntos *incomparáveis* com ω .

Exemplo 73.10. Em **1878**, Cantor afirmou que uma união enumerável de conjuntos enumeráveis é enumerável. Não apresentou prova, talvez por considerar a prova óbvia. Agora, com a nossa cautela, provamos uma versão mais geral deste resultado em **Proposição 72.12**. Mas a nossa prova assumiu explicitamente a Escolha. E mesmo a prova do resultado menos geral requer a Escolha Enumerável.

Teorema 73.11 (em $\mathbf{Z}^- + \text{Escolha Enumerável}$). *Se A_n é enumerável para cada $n \in \omega$, então $\bigcup_{n < \omega} A_n$ é enumerável.*

Prova. Sem perda de generalidade, suponhamos que cada $A_n \neq \emptyset$. Logo, para cada $n \in \omega$ existe uma sobrejeção $f_n: \omega \rightarrow A_n$. Defina-se $f: \omega \times \omega \rightarrow \bigcup_{n < \omega} A_n$ por $f(m, n) = f_n(m)$. O resultado segue porque $\omega \times \omega$ é enumerável (Proposição 4.12) e f é uma sobrejeção. $\square$

Notou o uso da Escolha Enumerável? Serve para escolher a nossa sequência de funções $f_0, f_1, f_2, \dots$.³ E, de novo, o resultado falha na ausência de qualquer princípio de Escolha. Especificamente, Feferman and Levy (1963) provou que é consistente com $\mathbf{ZF}$ que uma união enumerável de conjuntos enumeráveis tenha cardinalidade $\aleph_1$. Mas eis uma formulação bem mais divertida do ponto, de Russell:

Isto ilustra-se com o milionário que comprava um par de meias sempre que comprava um par de botas, e nunca noutra ocasião, e que tinha tal paixão por comprar ambos que acabou por ter $\aleph_0$ pares de botas e $\aleph_0$ pares de meias. . . Nas botas podemos distinguir direita e esquerda, e portanto podemos fazer uma seleção de uma em cada par, nomeadamente, podemos escolher todas as botas direitas ou todas as esquerdas; mas com as meias não se impõe qualquer princípio de seleção, e não podemos ter a certeza, a menos que assumamos o axioma multiplicativo [isto é, em efeito, a Escolha], de que existe alguma classe constituída por uma meia de cada par. (Russell, 1919, p. 126)

Em suma, alguma forma de Escolha é necessária para provar o seguinte: se tem uma quantidade enumerável de pares de meias, então tem (apenas) uma quantidade enumerável de meias. E, de fato, sem a Escolha Enumerável (ou algo equivalente), uma união enumerável de conjuntos enumeráveis pode deixar de ser enumerável.

A moral é que a Escolha Enumerável foi usada repetidamente, sem grande consciência por parte de quem a usava. A questão filosófica é: como poderíamos justificar a Escolha Enumerável?

Uma tentativa de justificação intuitiva poderia apelar a uma super-tarefa. Suponhamos que fazemos a primeira escolha em $1/2$ minuto, a segunda em $1/4$ minuto, . . . , a n -ésima em $1/2^n$ minuto, . . . Então, dentro de 1 minuto, teremos feito uma ω -sequência de escolhas, e definido uma função de escolha.

Mas que poderia realmente dizer-nos um tal experimento mental? Para começar, depende de tomar a ideia de “escolher” de modo bastante literal. Para além disso, parece atar a matemática a possibilidades metafísicas.

Mais importante: não nos dará justificação para a Escolha *sem reservas*, em vez da *mera* Escolha Enumerável. Porque, se precisarmos que *todo* o conjunto

³Um uso semelhante da Escolha ocorreu em Proposição 72.12, quando demos a instrução “Para cada $\beta \in \alpha$, fixe uma injeção f_β ”.

73.6. CONSIDERAÇÕES INTRÍNSECAS SOBRE A ESCOLHA

tenha função de escolha, então precisaremos de poder realizar uma “super-tarefa de comprimento ordinal arbitrário.” Sem rodeios, essa ideia é risível.

`content/set-theory/choice/justifications.tex`

73.6 Considerações Intrínsecas sobre a Escolha

`sth:choice:justifications:`
`sec`

A questão mais ampla, então, é se a Boa Ordenação, ou a Escolha, ou de fato a comparabilidade de todos os conjuntos quanto ao tamanho—não importa qual—podem ser justificadas.

Eis uma tentativa de justificação *intrínseca*. De volta a [Seção 66.1](#), introduzimos vários princípios sobre a hierarquia. Um deles merece ser repetido:

Estágios-acumulam. Para qualquer estágio S , e para quaisquer conjuntos que foram formados *antes* do estágio S : forma-se no estágio S um conjunto cujos membros são exatamente esses conjuntos. Nada mais se forma no estágio S .

De fato, muitos autores sugeriram que o Axioma da Escolha pode ser justificado via (algo como) este princípio. Daremos brevemente um resumo dessa abordagem.

Começamos por um resultado simples, que oferece *mais uma* formulação equivalente da Escolha:

`sth:choice:justifications:`
`choiceset`

Teorema 73.12 (em ZF). *A Escolha é equivalente ao princípio seguinte. Se os **membros** de A são disjuntos dois a dois e não vazios, então existe algum C tal que $C \cap x$ é um singular para todo o $x \in A$. (Chamamos a tal C um conjunto de escolha para A .)*

A prova deste resultado é direta, e a deixamos como exercício para o leitor.

Problema 73.1. Prove [Teorema 73.12](#). Se tiver dificuldades, encontrará uma prova em ([Potter, 2004](#), pp. 242–3).

O ponto essencial é que um conjunto de escolha para A é precisamente o contradomínio de uma função de escolha para A . Assim, para justificar a Escolha, podemos simplesmente tentar justificar a sua formulação equivalente, em termos da existência de conjuntos de escolha. E é exatamente isso que faremos agora.

Sejam os **membros** de A disjuntos dois a dois e não vazios. Por *Estágios-são-chave* (ver [Seção 66.1](#)), A é formado em algum estágio S . Note-se que todos os **membros** de $\bigcup A$ estão disponíveis antes do estágio S . Agora, por *Estágios-acumulam*, para *quaisquer* conjuntos formados antes de S , forma-se um conjunto cujos membros são exatamente esses conjuntos. Dito de outro modo: toda a *possível* coleção de conjuntos disponíveis mais cedo existirá em S . Mas é certamente *possível* seleccionar objetos que possam ser reunidos num

conjunto de escolha para A ; isso é apenas um certo subconjunto muito específico de $\bigcup A$. Logo: existe tal conjunto de escolha, como requerido.

Bem, esta é uma tentativa *muito* rápida de oferecer uma justificação da Escolha por razões intrínsecas. Mas, para aprofundar esta ideia, deve ler o desenvolvimento elegante de Potter (2004, §14.8).

content/set-theory/choice/banach.tex

73.7 Paradoxo de Banach–Tarski

Também podemos tentar justificar a Escolha, como Boolos tentou justificar a Substituição, apelando a considerações *extrínsecas* (ver [Seção 69.3](#)). Afinal, adotar a Escolha tem muitas consequências desejáveis: poder comparar qualquer par de cardinais; poder bem ordenar qualquer conjunto; poder tratar os cardinais como um tipo particular de ordinal; etc.

Por vezes, contudo, afirma-se que a Escolha tem consequências *indesejáveis*. Na maior parte dos casos, isso deve-se a um resultado de [Banach and Tarski \(1924\)](#).

Teorema 73.13 (Paradoxo de Banach–Tarski (em ZFC)). *Qualquer bola pode ser decomposta num número finito de peças, que podem ser remontadas (por rotação e transporte) de modo a formar duas cópias dessa bola.*

À primeira vista, isto surpreende. Claramente as duas bolas têm *o dobro* do volume da bola original. Mas os movimentos rígidos—rotação e transporte—não alteram o volume. Parece assim que Banach–Tarski nos permite fazer aparecer matéria nova por magia.

Pior ainda.⁴ Raciocínio semelhante mostra que uma ervilha pode ser cortada num número finito de peças, que depois podem ser remontadas (por rotação e transporte) de modo a formar um objeto com a forma e o tamanho do Big Ben.

Nada disto, porém, vale em **ZF** isoladamente.⁵ Logo temos de decidir: rejeitar a Escolha, ou aprender a viver com o “paradoxo”.

Sugerimos que devemos aprender a viver com o “paradoxo”. De fato, não nos parece propriamente um paradoxo. Em particular, não vemos por que haveria de ser mais ou menos paradoxal do que qualquer um dos resultados seguintes:⁶

1. Há tantos pontos no intervalo $(0, 1)$ como em $\mathbb{R}$.
Prova: considere-se $\tan(\pi(r - 1/2))$.

⁴Ver [Tomkowicz and Wagon \(2016, Theorem 3.12\)](#).

⁵Embora Banach–Tarski possa ser provado com princípios estritamente mais fracos do que a Escolha; ver [Tomkowicz and Wagon \(2016, 303\)](#).

⁶[Potter \(2004, 276–7\)](#), [Weston \(2003, 16\)](#), [Tomkowicz and Wagon \(2016, 31, 308–9\)](#) fazem pontos semelhantes, com outros exemplos.

73.8. APÊNDICE: PARADOXO DE VITALI

2. Há tantos pontos numa reta como num quadrado.

Ver [Seção 77.3](#) e [Seção 77.5](#).

3. Existem curvas que preenchem o espaço.

Ver [Seção 77.3](#) e [Seção 77.6](#).

Nenhum destes três resultados requer a Escolha. De fato, já os tratamos simplesmente como matemática surpreendente e bela. Talvez devêssemos adotar a mesma atitude perante o Paradoxo de Banach–Tarski.

Com certeza, é necessária aqui uma observação técnica; mas basta manter a cabeça fria. Os movimentos rígidos preservam volume. Consequentemente, as cinco⁷ peças em que a bola é decomposta não podem todas ser *mensuráveis*. Grosso modo, não faz sentido atribuir volume a estas peças individualmente. Deve pensar nelas como dispersões infinitas de pontos impossíveis de visualizar. Agora, pode ser “estranho” conceber tais conjuntos “infinitamente dispersos”. Mas a sua existência parece seguir da injunção, incorporada em *Estágios-acumulam*, de que se formem *todas* as coleções possíveis de conjuntos disponíveis mais cedo.

Se nada disto convence, eis um último argumento (extrínseco) a favor de abraçar o Paradoxo de Banach–Tarski. Ele implica de imediato a melhor piada matemática de sempre:

Pergunta. O que é um anagrama de “Banach–Tarski”?

Resposta. “Banach–Tarski Banach–Tarski”.

[content/set-theory/choice/vitali.tex](#)

73.8 Apêndice: Paradoxo de Vitali

[sth:choice:vitali:](#)
[sec](#)

Para ter uma ideia real de se a construção de Banach–Tarski é aceitável ou não, devíamos examinar a sua *prova*. Infelizmente, isso exigiria muito mais álgebra do que podemos apresentar aqui. Contudo, podemos oferecer algumas observações rápidas que talvez esclareçam a prova de Banach–Tarski,⁸ concentrando-nos no resultado seguinte:

[sth:choice:vitali:](#)
[vitaliparadox](#)

Teorema 73.14 (Paradoxo de Vitali (em ZFC)). *Qualquer círculo pode ser decomposto numa quantidade enumerável de peças, que podem ser remontadas (por rotação e transporte) de modo a formar duas cópias desse círculo.*

⁷Enunciamos o Paradoxo em termos de “número finito de peças”. De fato, [Robinson \(1947\)](#) provou que a decomposição pode ser feita com *cinco* peças (mas não menos). Para uma prova, ver [Tomkowicz and Wagon \(2016\)](#), pp. 66–7).

⁸Para um tratamento bem mais completo, ver [Weston \(2003\)](#) ou [Tomkowicz and Wagon \(2016\)](#).

O Paradoxo de Vitali é muito mais fácil de provar do que o Paradoxo de Banach–Tarski. Chamamos-lhe “Paradoxo de Vitali”, pois segue-se da construção, em 1905, de um conjunto não mensurável. Mas os aspectos conjuntistas da prova do Paradoxo de Vitali e do Paradoxo de Banach–Tarski são muito semelhantes. A diferença essencial entre os resultados é que Banach–Tarski considera uma decomposição *finita*, enquanto o Paradoxo de Vitali considera uma decomposição *enumerável infinita*. Como Weston (2003) diz, o Paradoxo de Vitali “certamente não é tão notável como o paradoxo de Banach–Tarski, mas ilustra que paradoxos geométricos podem ocorrer mesmo em situações ‘simples’.”

O Paradoxo de Vitali trata de uma figura bidimensional, um círculo. Logo trabalhamos no plano, $\mathbb{R}^2$. Seja R o conjunto das rotações (no sentido dos ponteiros do relógio) de pontos em torno da origem por valores em radianos *racionais* no intervalo $[0, 2\pi)$. Eis alguns fatos algébricos sobre R (se não compreender o enunciado do resultado, a prova esclarecerá o seu significado):

Lema 73.15. *R forma um grupo abeliano sob a composição de funções.*

*sth:choice:vitali:
rotationsgroupabelian*

Prova. Escrevendo 0_R para a rotação por 0 radianos, este é um elemento neutro para R , pois $\rho \circ 0_R = 0_R \circ \rho = \rho$ para qualquer $\rho \in R$.

Todo o elemento tem inverso. Onde $\rho \in R$ roda por r radianos, $\rho^{-1} \in R$ roda por $2\pi - r$ radianos, de modo que $\rho \circ \rho^{-1} = 0_R$.

A composição é associativa: $(\tau \circ \sigma) \circ \rho = \tau \circ (\sigma \circ \rho)$ para quaisquer $\rho, \sigma, \tau \in R$

A composição é comutativa: $\sigma \circ \rho = \rho \circ \sigma$ para quaisquer $\rho, \sigma \in R$. $\square$

De fato, podemos partir o nosso grupo R ao meio e depois usar qualquer metade para recuperar o grupo inteiro:

Lema 73.16. *Existe uma partição de R em dois conjuntos disjuntos, R_1 e R_2 , ambos dos quais são uma base para R .*

*sth:choice:vitali:
disjointgroup*

Prova. Seja R_1 o conjunto das rotações por valores racionais em radianos em $[0, \pi)$; seja $R_2 = R \setminus R_1$. Por álgebra elementar, $\{\rho \circ \rho : \rho \in R_1\} = R$. Um resultado semelhante obtém-se para R_2 . $\square$

Usaremos este fato sobre grupos para estabelecer **Teorema 73.14**. Seja $\mathbf{S}$ o círculo unitário, isto é, o conjunto dos pontos exatamente a 1 unidade da origem do plano, isto é, $\{(r, s) \in \mathbb{R}^2 : \sqrt{r^2 + s^2} = 1\}$. Partiremos $\mathbf{S}$ em partes considerando a relação seguinte em $\mathbf{S}$:

$$r \sim s \text{ sse } (\exists \rho \in R)\rho(r) = s.$$

Ou seja, os pontos de $\mathbf{S}$ estão ligados por esta relação sse se pode passar de um para o outro por uma rotação de valor racional em torno da origem. Sem surpresa:

Lema 73.17. *$\sim$ é uma relação de equivalência.*

73.8. APÊNDICE: PARADOXO DE VITALI

Prova. Trivial, usando **Lema 73.15**. □

Invocamos agora a Escolha para obter um conjunto, C , contendo exatamente um membro de cada classe de equivalência de $\mathbf{S}$ sob $\sim$. Ou seja, consideramos uma função de escolha f no conjunto das classes de equivalência,⁹

$$E = \{[r]_{\sim} : r \in \mathbf{S}\},$$

e seja $C = \text{Im}(f)$. Para cada rotação $\rho \in R$, o conjunto $\rho[C]$ consiste nos pontos obtidos ao aplicar a rotação ρ a cada ponto de C . Os dois resultados seguintes mostram que estes conjuntos cobrem o círculo por completo e sem sobreposição:

sth:choice:vitali: **Lema 73.18.** $\mathbf{S} = \bigcup_{\rho \in R} \rho[C]$.
vitalicover

Prova. Fixe-se $s \in \mathbf{S}$; existe algum $r \in C$ tal que $r \in [s]_{\sim}$, isto é, $r \sim s$, isto é, $\rho(r) = s$ para algum $\rho \in R$. □

sth:choice:vitali: **Lema 73.19.** Se $\rho_1 \neq \rho_2$ então $\rho_1[C] \cap \rho_2[C] = \emptyset$.
vitalinooverlap

Prova. Suponhamos $s \in \rho_1[C] \cap \rho_2[C]$. Logo $s = \rho_1(r_1) = \rho_2(r_2)$ para alguns $r_1, r_2 \in C$. Daí $\rho_2^{-1}(\rho_1(r_1)) = r_2$, e $\rho_2^{-1} \circ \rho_1 \in R$, logo $r_1 \sim r_2$. Logo $r_1 = r_2$, pois C selecciona exatamente um membro de cada classe de equivalência sob $\sim$. Logo $s = \rho_1(r_1) = \rho_2(r_1)$, e portanto $\rho_1 = \rho_2$. □

Aplicamos agora os fatos algébricos anteriores ao nosso círculo:

sth:choice:vitali: **Lema 73.20.** Existe uma partição de $\mathbf{S}$ em dois conjuntos disjuntos, D_1 e D_2 ,
pseudobanacharski tal que D_1 pode ser particionado numa quantidade enumerável de conjuntos que podem ser rodados de modo a formar uma cópia de $\mathbf{S}$ (e analogamente para D_2).

Prova. Usando R_1 e R_2 de **Lema 73.16**, seja:

$$D_1 = \bigcup_{\rho \in R_1} \rho[C] \qquad D_2 = \bigcup_{\rho \in R_2} \rho[C]$$

Isto é uma partição de $\mathbf{S}$, por **Lema 73.18**, e D_1 e D_2 são disjuntos por **Lema 73.19**. Por construção, D_1 pode ser particionado numa quantidade enumerável de conjuntos, $\rho[C]$ para cada $\rho \in R_1$. E estes podem ser rodados de modo a formar uma cópia de $\mathbf{S}$, pois $\mathbf{S} = \bigcup_{\rho \in R} \rho[C] = \bigcup_{\rho \in R_1} (\rho \circ \rho)[C]$ por **Lema 73.16** e **Lema 73.18**. O mesmo raciocínio vale para D_2 . □

⁹Como R é enumerável, cada membro de E é enumerável. Como $\mathbf{S}$ é não enumerável, segue-se de **Lema 73.18** e **Proposição 72.12** que E é não enumerável. Logo isto é um uso de Escolha não enumerável.

Isto implica de imediato o Paradoxo de Vitali. Pois podemos gerar *duas* cópias de S a partir de S , apenas partindo-o numa quantidade enumerável de peças (os vários $\rho[C]$) e depois movendo-as rigidamente (basta rodar cada peça de D_1 , e primeiro transportar e depois rodar cada peça de D_2).

Recapitulemos a estratégia de prova. Começamos com fatos algébricos sobre o grupo de rotações no plano. Usamos este grupo para partir S em classes de equivalência. Chegamos então a um “paradoxo”, usando a Escolha para seleccionar elementos de cada classe.

Usamos exatamente a mesma estratégia para provar Banach–Tarski. A principal diferença é que os fatos algébricos usados na prova de Banach–Tarski são bastante mais complicados do que os usados no Paradoxo de Vitali. Mas esses fatos algébricos nada têm a ver com a Escolha. Resumimos-os rapidamente.

Para provar Banach–Tarski, começamos por estabelecer um análogo de **Lema 73.16**: qualquer grupo livre pode ser partido em quatro peças que, intuitivamente, podemos “mover” de modo a recuperar duas cópias do grupo inteiro.¹⁰ Mostramos depois que podemos usar duas rotações particulares em torno da origem de $\mathbb{R}^3$ para gerar um grupo livre de rotações, F .¹¹ (Ainda sem Escolha.) Consideramos agora pontos na superfície da esfera como “semelhantes” sse um pode obter-se do outro por uma rotação em F . Usamos então a Escolha para seleccionar exatamente um ponto de cada classe de equivalência de pontos “semelhantes”. Aplicando a nossa divisão de F à superfície da esfera, como em **Lema 73.20**, partimos essa superfície em quatro peças, que podemos “mover” de modo a obter duas cópias da superfície da esfera. E isto estabelece (Hausdorff, 1914):

Teorema 73.21 (Paradoxo de Hausdorff (em ZFC)). *A superfície de qualquer esfera pode ser decomposta num número finito de peças, que podem ser remontadas (por rotação e transporte) de modo a formar duas cópias disjuntas dessa esfera.*

Ainda são precisos alguns truques algébricos para obter o Teorema completo de Banach–Tarski (que não diz só respeito à superfície da esfera, mas também ao seu interior). Francamente, porém, isto é apenas a cereja no topo do bolo algébrico. Por isso Weston escreve:

[...] o resultado sobre grupos livres é o passo *chave* na prova do paradoxo de Banach–Tarski. Deste ponto de vista, o paradoxo de Banach–Tarski não é tanto uma afirmação sobre $\mathbb{R}^3$ quanto uma afirmação sobre a complexidade do grupo [das translações e rotações em $\mathbb{R}^3$]. (Weston, 2003, p. 16)

¹⁰O fato de podermos usar *quatro* peças deve-se a Robinson (1947). Para uma prova recente, ver Tomkowicz and Wagon (2016, Theorem 5.2). Seguimos Weston (2003, p. 3) ao descrever isto como “mover” as peças do grupo.

¹¹Ver Tomkowicz and Wagon (2016, Theorem 2.1).

73.8. APÊNDICE: PARADOXO DE VITALI

Ou seja: podermos ou não oferecer uma decomposição *finita* (como em Banach–Tarski) ou uma decomposição *enumerável infinita* (como no Paradoxo de Vitali) resume-se a certos fatos de teoria dos grupos sobre trabalhar em duas ou três dimensões.

Admitidamente, esta última observação estraga ligeiramente a piada no final de [Seção 73.7](#). Por ser bidimensional, “Banach–Tarski” tem de ser dividido numa *infinitude* enumerável de peças, se se quiserem rearranjar essas peças de modo a formar “Banach–Tarski Banach–Tarski”. Para consertar a piada, é preciso escrever em três dimensões. Deixamos isto como exercício para o leitor.

Um último comentário. Na [Seção 73.7](#), mencionamos que as “peças” da esfera que se obtêm não podem ser *mensuráveis*, mas têm de ser dispersões infinitas impossíveis de visualizar. O mesmo vale para o nosso uso da Escolha ao obter [Lema 73.20](#). E tudo isto merece explicação.

De novo, devemos esboçar alguma base (mas isto é *apenas* um esboço; talvez queira consultar uma entrada de manual sobre *medida*). Definir uma medida para um conjunto X é atribuir um valor $\mu(E) \in \mathbb{R}$ a cada E nalguma “ σ -álgebra” sobre X . Os detalhes aqui não são essenciais, exceto que a função μ deve obedecer ao princípio da aditividade enumerável: a medida de uma união enumerável de conjuntos disjuntos é a soma das medidas individuais, isto é, $\mu(\bigcup_{n < \omega} X_n) = \sum_{n < \omega} \mu(X_n)$ sempre que os X_n são disjuntos. Dizer que um conjunto é “não mensurável” é dizer que nenhuma medida pode ser atribuída de modo adequado. Agora, usando o nosso R de antes:

Corolário 73.22 (Vitali). *Seja μ uma medida tal que $\mu(\mathbf{S}) = 1$, e tal que $\mu(X) = \mu(Y)$ se X e Y são congruentes. Então $\rho[C]$ é não mensurável para todo o $\rho \in R$.*

Prova. Por redução ao absurdo, suponhamos o contrário. Seja então $\mu(\sigma[C]) = r$ para algum $\sigma \in R$ e algum $r \in \mathbb{R}$. Para qualquer $\rho \in R$, $\rho[C]$ e $\sigma[C]$ são congruentes, logo $\mu(\rho[C]) = r$ para qualquer $\rho \in R$. Por [Lema 73.18](#) e [Lema 73.19](#), $\mathbf{S} = \bigcup_{\rho \in R} \rho[C]$ é uma união enumerável de conjuntos disjuntos dois a dois. Logo a aditividade enumerável dita que $\mu(\mathbf{S}) = 1$ é a soma das medidas de cada $\rho[C]$, isto é,

$$1 = \mu(\mathbf{S}) = \sum_{\rho \in R} \mu(\rho[C]) = \sum_{\rho \in R} r$$

Mas se $r = 0$ então $\sum_{\rho \in R} r = 0$, e se $r > 0$ então $\sum_{\rho \in R} r = \infty$. □

Parte XVI

Methods

This part covers general and methodological material, especially explanations of various proof methods a non-mathematics student may be unfamiliar with. It currently contains a chapter on how to write proofs, and a chapter on induction, but additional sections for those, exercises, and a chapter on mathematical terminology is also planned.

Capítulo 74

Provas

`content/methods/proofs/introduction.tex`

74.1 Introdução

Com base em suas experiências na lógica introdutória, você talvez se sinta à vontade com um sistema de **derivação**—provavelmente um sistema de **derivação** de dedução natural ou no estilo Fitch, ou talvez um sistema de árvores de prova. Você provavelmente se lembra de fazer provas nesses sistemas, seja provando **a fórmula**, seja mostrando que um dado argumento é válido. Para fazer isso, você aplicava as regras do sistema até obter o resultado final desejado. Ao raciocinar *sobre* lógica, também provamos coisas, mas na maioria dos casos não estamos usando um sistema de **derivação**. De fato, a maior parte das provas que consideramos é feita em português (talvez com alguma linguagem simbólica inserida) e não inteiramente na linguagem da lógica de primeira ordem. Ao construir tais provas, você pode a princípio ficar perdido—como provo algo sem um sistema de **derivação**? Como começo? Como sei se minha prova está correta?

Antes de tentar uma prova, é importante saber o que é uma prova e como construir uma. Como o nome sugere, uma *prova* destina-se a mostrar que algo

74.2. COMEÇANDO UMA PROVA

é verdadeiro. Você pode pensar nisso em termos de um diálogo—alguém lhe pergunta se algo é verdadeiro, digamos, se todo primo diferente de dois é um número ímpar. Responder “sim” não é suficiente; a pessoa pode querer saber *por quê*. Nesse caso, você lhe daria uma prova.

No discurso cotidiano, pode ser suficiente esboçar uma resposta, ou dar uma resposta incompleta. Na lógica e na matemática, porém, queremos uma prova rigorosa—queremos mostrar que algo é verdadeiro além de *qualquer* dúvida. Isso significa que cada passo de nossa prova deve ser justificado, e a justificativa deve ser convincente (isto é, a suposição que você está usando é de fato assumida no enunciado do teorema que você está provando, as definições que você aplica devem ser aplicadas corretamente, as justificativas invocadas devem ser inferências corretas, etc.).

Em geral, estamos provando algum enunciado. Chamamos os enunciados que provamos por vários nomes: proposições, teoremas, lemas ou corolários. Uma proposição é um enunciado básico digno de prova: importante o suficiente para ser registrado, mas talvez não particularmente profundo nem aplicado com frequência. Um teorema é uma proposição significativa e importante. Sua prova é frequentemente dividida em vários passos, e às vezes recebe o nome da pessoa que primeiro o provou (por exemplo, o Teorema de Cantor, o teorema de Löwenheim–Skolem) ou do fato de que trata (por exemplo, o teorema da completude). Um lema é uma proposição ou teorema usado na prova de um resultado mais importante. De modo confuso, às vezes os lemas são resultados importantes em si mesmos, e também recebem o nome da pessoa que os introduziu (por exemplo, o Lema de Zorn). Um corolário é um resultado que decorre facilmente de outro.

Um enunciado a ser provado frequentemente contém suposições que esclarecem sobre quais tipos de coisas estamos provando algo. Ele pode começar com “Seja φ a fórmula da forma $\psi \rightarrow \chi$ ” ou “Suponha $\Gamma \vdash \varphi$ ” ou algo do gênero. Essas são as *hipóteses* da proposição, teorema ou lema, e você pode assumi-las como verdadeiras em sua prova. Elas restringem o que estamos provando e também introduzem alguns nomes para os objetos sobre os quais estamos falando. Por exemplo, se sua proposição começa com “Seja φ a fórmula da forma $\psi \rightarrow \chi$ ”, você está provando algo apenas sobre todas as fórmulas de certo tipo (a saber, condicionais), e fica entendido que $\psi \rightarrow \chi$ é um condicional arbitrário sobre o qual sua prova falará.

`content/methods/proofs/starting-proofs.tex`

74.2 Começando uma Prova

`math:prf:str:` Mas por onde começar?
`sec`

Foi-lhe dado algo a provar, de modo que isso deveria ser a última coisa mencionada na prova (você pode, obviamente, *anunciar* que vai prová-lo no começo, mas não convém usá-lo como suposição). Escreva o que você está

tentando provar no rodapé de uma folha de papel em branco—assim você não perde de vista seu objetivo.

Em seguida, você pode ter algumas suposições que pode usar (isso ficará mais claro quando falarmos sobre o *tipo* de prova que você está fazendo na próxima seção). Escreva-as no topo da página e certifique-se de sinalizar que são suposições (isto é, se você está supondo p , escreva “suponha que p ” ou “assuma que p ”). Por fim, pode haver na questão algumas definições que você precisa conhecer. Pode ser que lhe digam para usar uma definição específica, ou pode haver várias definições nas suposições ou na conclusão para a qual você está trabalhando. *Anote-as e certifique-se de entender o que significam.*

A forma como você organiza sua prova também dependerá da forma da questão. A próxima seção fornece detalhes sobre como organizar sua prova com base no tipo de sentença.

`content/methods/proofs/using-definitions.tex`

74.3 Usando Definições

Mencionamos que você deve estar familiarizado com todas as definições que possam ser usadas na prova, e que saiba aplicá-las corretamente. Esse é um ponto realmente importante, e vale a pena examiná-lo em um pouco mais de detalhe. As definições são usadas para abreviar propriedades e relações, de modo que possamos falar delas de forma mais sucinta. A abreviação introduzida é chamada de *definiendum*, e aquilo que ela abrevia é o *definiens*. Em provas, com frequência temos que voltar a como o definiendum foi introduzido, porque temos que explorar a estrutura lógica do definiens (a versão longa da qual o termo definido é a abreviação) para concluir nossa prova. Ao desdobrar as definições, você garante que está chegando ao cerne de onde a ação lógica acontece.

`mod:prf:def:`
`sec`

Começaremos com um exemplo. Suponha que você queira provar o seguinte:

Proposição 74.1. *Para quaisquer conjuntos A e B , $A \cup B = B \cup A$.*

Para sequer iniciar a prova, precisamos saber o que significa que dois conjuntos sejam idênticos; isto é, precisamos saber o que o “=” naquela equação significa para conjuntos. Os conjuntos são definidos como idênticos sempre que têm os mesmos **membros**. Assim, a definição que temos que desdobrar é:

Definição 74.2. Os conjuntos A e B são *idênticos*, $A = B$, sse todo **membro** de A é **um membro** de B , e vice-versa.

Esta definição usa A e B como marcadores de posição para conjuntos arbitrários. O que ela define—o *definiendum*—é a expressão “ $A = B$ ”, dando a condição sob a qual $A = B$ é verdadeira. Essa condição—“todo **membro** de A

74.3. USANDO DEFINIÇÕES

é um membro de B , e vice-versa”—é o *definiens*.¹ A definição especifica que $A = B$ é verdadeira se, e somente se (abreviamos isso por “sse”) a condição vale.

Quando você aplica a definição, tem que fazer corresponder o A e o B da definição ao caso de que está tratando. No nosso caso, isso significa que, para que $A \cup B = B \cup A$ seja verdadeira, cada $z \in A \cup B$ também deve estar em $B \cup A$, e vice-versa. A expressão $A \cup B$ na proposição faz o papel de A na definição, e $B \cup A$, o de B . Como A e B são usados tanto na definição quanto no enunciado da proposição que estamos provando, mas em usos diferentes, você precisa ter cuidado para não confundir os dois. Por exemplo, seria um erro pensar que você poderia provar a proposição mostrando que todo membro de A é um membro de B , e vice-versa—isso mostraria que $A = B$, não que $A \cup B = B \cup A$. (Além disso, como A e B podem ser quaisquer dois conjuntos, você não chegaria muito longe, pois, se nada for suposto sobre A e B , eles bem podem ser conjuntos diferentes.)

Dentro da prova, estamos lidando com noções da teoria dos conjuntos, como união, e portanto também precisamos conhecer os significados do símbolo $\cup$ a fim de entender como a prova deve prosseguir. E, às vezes, desdobrar a definição dá origem a outras definições a desdobrar. Por exemplo, $A \cup B$ é definido como $\{z : z \in A \text{ ou } z \in B\}$. Então, se você quer provar que $x \in A \cup B$, desdobrar a definição de $\cup$ lhe diz que você tem que provar $x \in \{z : z \in A \text{ ou } z \in B\}$. Agora você também tem que lembrar que $x \in \{z : \dots z \dots\}$ sse $\dots x \dots$. Assim, desdobrando ainda mais a definição da notação $\{z : \dots z \dots\}$, o que você tem que mostrar é: $x \in A$ ou $x \in B$. Assim, “todo membro de $A \cup B$ é também um membro de $B \cup A$ ” significa, na verdade: “para todo x , se $x \in A$ ou $x \in B$, então $x \in B$ ou $x \in A$.” Se desdobramos completamente as definições na proposição, vemos que o que temos que mostrar é isto:

Proposição 74.3. *Para quaisquer conjuntos A e B : (a) para todo x , se $x \in A$ ou $x \in B$, então $x \in B$ ou $x \in A$, e (b) para todo x , se $x \in B$ ou $x \in A$, então $x \in A$ ou $x \in B$.*

O que importa é que desdobrar definições é uma parte necessária da construção de uma prova. Fazê-lo corretamente é às vezes difícil: você deve ter cuidado para distinguir e fazer corresponder as variáveis da definição e os termos na afirmação que está provando. Para ter sucesso, você deve saber o que a questão está pedindo e o que todos os termos usados na questão significam—muitas vezes você precisará desdobrar mais de uma definição. Em provas simples como as que seguem, a solução decorre quase imediatamente das próprias definições. É claro que nem sempre será assim tão simples.

¹Neste caso particular—e de modo bastante confuso!—quando $A = B$, os conjuntos A e B são apenas um único e mesmo conjunto, ainda que usemos letras diferentes para ele do lado esquerdo e do lado direito. Mas os modos pelos quais esse conjunto é identificado podem ser diferentes, e isso torna a definição não trivial.

Problema 74.1. Suponha que lhe peçam para provar que $A \cap B \neq \emptyset$. Desdobre todas as definições que ocorrem aqui, isto é, reenuncie isto de um modo que não mencione “ $\cap$ ”, “ $=$ ”, ou “ $\emptyset$ ”.

`content/methods/proofs/inference-patterns.tex`

74.4 Padrões de Inferência

As provas são compostas de inferências individuais. Quando fazemos uma inferência, normalmente a indicamos usando uma palavra como “então,” “assim,” ou “portanto.” A inferência muitas vezes se apoia em um ou dois fatos que já temos disponíveis em nossa prova—pode ser algo que supusemos, ou algo que concluímos por uma inferência anterior. Para deixar claro, podemos rotular essas coisas, e na inferência indicamos quais outras afirmações estamos usando na inferência. Uma inferência muitas vezes também conterá uma explicação de *por que* nossa nova conclusão decorre das coisas que a precedem. Há alguns padrões comuns de inferência que são usados com muita frequência em provas; percorreremos alguns deles abaixo. Alguns padrões de inferência, como as provas por indução, são mais elaborados (e serão discutidos mais adiante).

math:prf:pat:
sec

Já discutimos um padrão de inferência: desdobrar, ou aplicar, uma definição. Quando desdobramos uma definição, apenas reenunciamos algo que envolve o definiendum usando o definiens. Por exemplo, suponha que já tenhamos estabelecido no curso de uma prova que $D = E$ (a). Então podemos aplicar a definição de $=$ para conjuntos e inferir: “Assim, por definição, de (a), todo **membro** de D é **um membro** de E e vice-versa.”

De modo um tanto confuso, muitas vezes não escrevemos a justificativa de uma inferência quando de fato a fazemos, mas antes. Suponha que ainda não tenhamos provado que $D = E$, mas queremos. Se $D = E$ é a conclusão que buscamos, então podemos reenunciar esse objetivo também aplicando a definição: para provar $D = E$ temos que provar que todo **membro** de D é **um membro** de E e vice-versa. Assim, nossa prova terá a forma: (a) provar que todo **membro** de D é **um membro** de E ; (b) todo **membro** de E é **um membro** de D ; (c) portanto, de (a) e (b), pela definição de $=$, $D = E$. Mas normalmente não escreveríamos desse jeito. Em vez disso, poderíamos escrever algo como,

Queremos mostrar $D = E$. Pela definição de $=$, isso equivale a mostrar que todo **membro** de D é **um membro** de E e vice-versa.

- (a) ... (uma prova de que todo **membro** de D é **um membro** de E)
- ...
- (b) ... (uma prova de que todo **membro** de E é **um membro** de D)
- ...

Usando uma Conjunção

Talvez o padrão de inferência mais simples seja o de tirar como conclusão uma das conjuntivas de uma conjunção. Em outras palavras: se supusemos ou já provamos que p e q , então temos o direito de inferir que p (e também que q). Essa é uma inferência tão básica que muitas vezes não é mencionada. Por exemplo, uma vez que tenhamos desdobrado a definição de $D = E$, estabelecemos que todo membro de D é um membro de E e vice-versa. Disso podemos concluir que todo membro de E é um membro de D (essa é a parte do “vice-versa”).

Provando uma Conjunção

Às vezes, o que lhe pedirão para provar terá a forma de uma conjunção; pedirão a você que “prove p e q .” Nesse caso, você simplesmente tem que fazer duas coisas: provar p , e então provar q . Você poderia dividir sua prova em duas seções e, para maior clareza, rotulá-las. Quando estiver fazendo suas primeiras anotações, você pode escrever “(1) Provar p ” no topo da página, e “(2) Provar q ” no meio da página. (É claro que você pode não ser explicitamente solicitado a provar uma conjunção, mas descobrir que sua prova exige que você prove uma conjunção. Por exemplo, se lhe pedem para provar que $D = E$, você descobrirá que, após desdobrar a definição de $=$, você tem que provar: todo membro de D é um membro de E e todo membro de E é um membro de D).

Provando uma Disjunção

Quando o que você está provando assume a forma de uma disjunção (isto é, é uma afirmação da forma “ p ou q ”), basta mostrar que um dos disjuntos é verdadeiro. No entanto, basicamente nunca acontece de qualquer um dos disjuntos simplesmente decorrer das suposições do seu teorema. Mais frequentemente, as suposições do seu teorema são elas próprias disjuntivas, ou você está mostrando que todas as coisas de certo tipo têm uma de duas propriedades, mas algumas das coisas têm uma e outras têm a outra propriedade. É aqui que a prova por casos é útil (veja abaixo).

Prova Condicional

Muitos teoremas que você encontrará estão na forma condicional (isto é, mostre que se p vale, então q também é verdadeiro). Esses casos são bons e fáceis de montar—basta supor o antecedente do condicional (neste caso, p) e provar a conclusão q a partir dele. Então, se o seu teorema diz “Se p então q ,” você começa sua prova com “suponha p ” e, ao final, você deve ter provado q .

Os condicionais podem ser enunciados de maneiras diferentes. Assim, em vez de “Se p então q ,” um teorema pode afirmar que “ p somente se q ,” “ q se p ,” ou “ q , desde que p .” Todos eles significam o mesmo e exigem supor p e provar q a partir dessa suposição. Lembre-se de que um bicondicional (“ p se e somente se (sse) q ”) é, na verdade, dois condicionais postos juntos: se p então

q , e se q então p . Tudo o que você tem que fazer, então, são duas instâncias de prova condicional: uma para o primeiro condicional e outra para o segundo. Às vezes, no entanto, é possível provar uma afirmação “sse” encadeando um monte de outras afirmações “sse”, de modo que você comece com “ p ” e termine com “ q ”—mas, nesse caso, você tem que garantir que cada passo realmente seja um “sse.”

Afirmações Universais

Usar uma afirmação universal é simples: se algo é verdadeiro para qualquer coisa, é verdadeiro para cada coisa particular. Então, se, digamos, a hipótese da sua prova é $A \subseteq B$, isso significa (desdobrando a definição de $\subseteq$) que, para todo $x \in A$, $x \in B$. Assim, se você já sabe que $z \in A$, você pode concluir $z \in B$.

Provar uma afirmação universal pode parecer um pouco complicado. Normalmente essas afirmações assumem a seguinte forma: “Se x tem P , então ele tem Q ” ou “Todos os P são Q .” É claro que pode não se ajustar a essa forma perfeitamente, e é preciso um pouco de prática para descobrir o que exatamente lhe pedem para provar. Mas: muitas vezes temos que provar que todos os objetos com alguma propriedade têm uma certa outra propriedade.

A maneira de provar uma afirmação universal é introduzir nomes ou variáveis para as coisas que têm uma propriedade e então mostrar que elas também têm a outra propriedade. Poderíamos expressar isso dizendo que, para provar algo para *todos* os P , você tem que prová-lo para um P *arbitrário*. E o nome introduzido é um nome para um P arbitrário. Normalmente usamos letras isoladas como esses nomes para coisas arbitrárias, e as letras costumam seguir convenções: por exemplo, usamos n para números naturais, φ para fórmulas, A para conjuntos, f para funções, etc.

O truque é manter a generalidade ao longo da prova. Você começa supondo que um objeto arbitrário (“ x ”) tem a propriedade P , e mostra (com base apenas em definições ou no que lhe é permitido supor) que x tem a propriedade Q . Como você não estipulou o que x é especificamente, a não ser que ele tem a propriedade P , então você pode afirmar que tudo que tem P tem a propriedade Q . Em suma, x é um substituto para *todas* as coisas com a propriedade P .

Proposição 74.4. *Para todos os conjuntos A e B , $A \subseteq A \cup B$.*

Prova. Sejam A e B conjuntos arbitrários. Queremos mostrar que $A \subseteq A \cup B$. Pela definição de $\subseteq$, isso equivale a: para todo x , se $x \in A$ então $x \in A \cup B$. Então seja $x \in A$ um membro arbitrário de A . Temos que mostrar que $x \in A \cup B$. Como $x \in A$, $x \in A$ ou $x \in B$. Assim, $x \in \{x : x \in A \vee x \in B\}$. Mas isso, pela definição de $\cup$, significa $x \in A \cup B$. $\square$

Prova por Casos

Suponha que você tenha uma disjunção como suposição ou como conclusão já estabelecida—você supôs ou provou que p ou q é verdadeira. Você quer provar

74.4. PADRÕES DE INFERÊNCIA

r . Você faz isso em dois passos: primeiro você supõe que p é verdadeira, e prova r , e depois supõe que q é verdadeira e prova r novamente. Isso funciona porque supomos ou sabemos que uma das duas alternativas vale. Os dois passos estabelecem que qualquer uma delas é suficiente para a verdade de r . (Se ambas são verdadeiras, temos não uma, mas duas razões para que r seja verdadeira. Não é necessário provar separadamente que r é verdadeira supondo tanto p quanto q .) Para indicar o que estamos fazendo, anunciamos que “distinguimos casos.” Por exemplo, suponha que saibamos que $x \in B \cup C$. $B \cup C$ é definido como $\{x : x \in B \text{ ou } x \in C\}$. Em outras palavras, por definição, $x \in B$ ou $x \in C$. Provaremos que $x \in A$ a partir disso supondo primeiro que $x \in B$, e provando $x \in A$ a partir dessa suposição, e então supondo $x \in C$, e novamente provando $x \in A$ a partir disso. Você escreveria “Distinguimos casos” sob a suposição, depois “Caso (1): $x \in B$ ” abaixo, e “Caso (2): $x \in C$ ” na metade da página. Então você prosseguiria preenchendo a metade superior e a metade inferior da página.

A prova por casos é especialmente útil se o que você está provando é, ele mesmo, disjuntivo. Eis um exemplo simples:

Proposição 74.5. *Suponha $B \subseteq D$ e $C \subseteq E$. Então $B \cup C \subseteq D \cup E$.*

Prova. Suponha (a) que $B \subseteq D$ e (b) $C \subseteq E$. Por definição, qualquer $x \in B$ também está $\in D$ (c) e qualquer $x \in C$ também está $\in E$ (d). Para mostrar que $B \cup C \subseteq D \cup E$, temos que mostrar que se $x \in B \cup C$ então $x \in D \cup E$ (pela definição de $\subseteq$). $x \in B \cup C$ sse $x \in B$ ou $x \in C$ (pela definição de $\cup$). De modo semelhante, $x \in D \cup E$ sse $x \in D$ ou $x \in E$. Então, temos que mostrar: para qualquer x , se $x \in B$ ou $x \in C$, então $x \in D$ ou $x \in E$.

Até aqui só desdobramos definições! Reformulamos nossa proposição sem $\subseteq$ e $\cup$ e ficamos tentando provar uma afirmação condicional universal. Pelo que discutimos acima, isso é feito supondo que x é algo a respeito do qual supomos que a parte do “se” é verdadeira, e passaremos a mostrar que a parte do “então” também é verdadeira. Em outras palavras, suporemos que $x \in B$ ou $x \in C$ e mostraremos que $x \in D$ ou $x \in E$.²

Suponha que $x \in B$ ou $x \in C$. Temos que mostrar que $x \in D$ ou $x \in E$. Distinguimos casos.

Caso 1: $x \in B$. Por (c), $x \in D$. Assim, $x \in D$ ou $x \in E$. (Aqui fizemos a inferência discutida na subseção anterior!)

Caso 2: $x \in C$. Por (d), $x \in E$. Assim, $x \in D$ ou $x \in E$. □

Provando uma Afirmação de Existência

Quando lhe pedem para provar uma afirmação de existência, a questão normalmente será da forma “prove que há um x tal que $\dots x \dots$ ”, isto é, que

²Este parágrafo apenas explica o que estamos fazendo—não é parte da prova, e você não precisa entrar em todo esse detalhe quando escreve suas próprias provas.

algum objeto tem a propriedade descrita por “... x ...”. Nesse caso, você terá que identificar um objeto adequado e mostrar que ele tem a propriedade exigida. Isso parece direto, mas uma prova desse tipo pode ser complicada. Tipicamente, ela envolve *construir* ou *definir* um objeto e provar que o objeto assim definido tem a propriedade exigida. Encontrar o objeto certo pode ser difícil, provar que ele tem a propriedade exigida pode ser difícil, e às vezes é até complicado mostrar que você teve sucesso em definir um objeto!

Em geral, você escreveria isso especificando o objeto, por exemplo, “seja x ...” (em que ... especifica qual objeto você tem em mente), possivelmente provando que ... de fato descreve um objeto que existe, e então passaria a mostrar que x tem a propriedade Q . Eis um exemplo simples.

Proposição 74.6. *Suponha que $x \in B$. Então há um A tal que $A \subseteq B$ e $A \neq \emptyset$.*

Prova. Suponha $x \in B$. Seja $A = \{x\}$.

Aqui definimos o conjunto A enumerando seus **membros**. Como supomos que x é um objeto, e sempre podemos formar um conjunto enumerando seus **membros**, não temos que mostrar que tivemos sucesso em definir um conjunto A aqui. No entanto, ainda temos que mostrar que A tem as propriedades exigidas pela proposição. A prova não está completa sem isso!

Como $x \in A$, $A \neq \emptyset$.

Isso se apoia na definição de A como $\{x\}$ e nos fatos óbvios de que $x \in \{x\}$ e $x \notin \emptyset$.

Como x é o único **membro** de $\{x\}$, e $x \in B$, todo **membro** de A é também **um membro** de B . Pela definição de $\subseteq$, $A \subseteq B$. $\square$

Usando Afirmações de Existência

Suponha que você saiba que alguma afirmação de existência é verdadeira (você a provou, ou ela é uma hipótese que você pode usar), digamos, “para algum x , $x \in A$ ” ou “há um $x \in A$.” Se você quiser usá-la em sua prova, você pode simplesmente fingir que tem um nome para uma das coisas que sua hipótese diz existir. Como A contém pelo menos uma coisa, há coisas a que esse nome poderia se referir. É claro que você pode não ser capaz de escolher uma ou de descrevê-la mais (a não ser que ela é $\in A$). Mas, para os fins da prova, você pode fingir que a escolheu e dar-lhe um nome. É importante escolher um nome que você ainda não tenha usado (ou que apareça nas suas hipóteses), caso contrário as coisas podem dar errado. Em sua prova, você indica isso passando de “para algum x , $x \in A$ ” para “Seja $a \in A$.” Agora você pode raciocinar sobre a , usar algumas outras hipóteses, etc., até chegar a uma conclusão, p . Se p não menciona mais a , p é independente da suposição de que $a \in A$, e você mostrou que ele decorre apenas da suposição “para algum x , $x \in A$.”

74.4. PADRÕES DE INFERÊNCIA

Proposição 74.7. *Se $A \neq \emptyset$, então $A \cup B \neq \emptyset$.*

Prova. Suponha $A \neq \emptyset$. Então, para algum x , $x \in A$.

Aqui apenas reenunciamos a hipótese da proposição. Essa hipótese, isto é, $A \neq \emptyset$, esconde uma afirmação existencial, à qual você só chega desdobrando algumas definições. A definição de $=$ nos diz que $A = \emptyset$ sse todo $x \in A$ também está $\in \emptyset$ e todo $x \in \emptyset$ também está $\in A$. Negando ambos os lados, obtemos: $A \neq \emptyset$ sse ou algum $x \in A$ está $\notin \emptyset$ ou algum $x \in \emptyset$ está $\notin A$. Como nada está $\in \emptyset$, o segundo disjuncto nunca pode ser verdadeiro, e “ $x \in A$ e $x \notin \emptyset$ ” se reduz a apenas $x \in A$. Então $A \neq \emptyset$ sse, para algum x , $x \in A$. Essa é uma afirmação de existência. Agora usamos essa afirmação de existência introduzindo um nome para um dos **membros** de A :

Seja $a \in A$.

Agora introduzimos um nome para uma das coisas $\in A$. Continuaremos a argumentar sobre a , mas teremos o cuidado de supor apenas que $a \in A$ e nada mais:

Como $a \in A$, $a \in A \cup B$, pela definição de $\cup$. Então, para algum x , $x \in A \cup B$, isto é, $A \cup B \neq \emptyset$.

Naquele último passo, fomos de “ $a \in A \cup B$ ” para “para algum x , $x \in A \cup B$.” Isso não menciona mais a , então sabemos que “para algum x , $x \in A \cup B$ ” decorre de “para algum x , $x \in A$ ” apenas. Mas isso significa que $A \cup B \neq \emptyset$. □

Talvez seja boa prática manter variáveis ligadas como “ x ” separadas de nomes hipotéticos como a , como fizemos. Na prática, porém, muitas vezes não o fazemos e apenas usamos x , assim:

Suponha $A \neq \emptyset$, isto é, há um $x \in A$. Pela definição de $\cup$, $x \in A \cup B$. Então $A \cup B \neq \emptyset$.

No entanto, quando você faz isso, tem que ter um cuidado extra para usar x e y diferentes para afirmações existenciais diferentes. Por exemplo, a seguinte *não* é uma prova correta de “Se $A \neq \emptyset$ e $B \neq \emptyset$ então $A \cap B \neq \emptyset$ ” (o que não é verdade).

Suponha $A \neq \emptyset$ e $B \neq \emptyset$. Então, para algum x , $x \in A$ e também, para algum x , $x \in B$. Como $x \in A$ e $x \in B$, $x \in A \cap B$, pela definição de $\cap$. Então $A \cap B \neq \emptyset$.

Você consegue identificar onde ocorre o passo incorreto e explicar por que o resultado não vale?

74.5 Um Exemplo

Nosso primeiro exemplo é o seguinte fato simples sobre uniões e interseções de conjuntos. Ele ilustrará o desdobramento de definições, provas de conjunções, de afirmações universais, e a prova por casos. mth:prf:ex1:
sec

Proposição 74.8. *Para quaisquer conjuntos A , B e C , $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$*

Vamos prová-la!

Prova. Queremos mostrar que, para quaisquer conjuntos A , B e C , $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

Primeiro desdobramos a definição de “=” no enunciado da proposição. Lembre-se de que provar que conjuntos são idênticos significa mostrar que os conjuntos têm os mesmos **membros**. Isto é, todos os **membros** de $A \cup (B \cap C)$ são também **membros** de $(A \cup B) \cap (A \cup C)$, e vice-versa. O “vice-versa” significa que também todo **membro** de $(A \cup B) \cap (A \cup C)$ deve ser **um membro** de $A \cup (B \cap C)$. Assim, ao desdobrar a definição, vemos que temos que provar uma conjunção. Vamos registrar isso:

Por definição, $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ sse todo **membro** de $A \cup (B \cap C)$ é também **um membro** de $(A \cup B) \cap (A \cup C)$, e todo **membro** de $(A \cup B) \cap (A \cup C)$ é **um membro** de $A \cup (B \cap C)$.

Como isso é uma conjunção, devemos provar cada conjuntiva separadamente. Vamos começar pela primeira: vamos provar que todo **membro** de $A \cup (B \cap C)$ é também **um membro** de $(A \cup B) \cap (A \cup C)$.

Esta é uma afirmação universal, e por isso consideramos um **membro** arbitrário de $A \cup (B \cap C)$ e mostramos que ele também deve ser **um membro** de $(A \cup B) \cap (A \cup C)$. Vamos escolher uma variável para chamar esse **membro** arbitrário, digamos, z . Nossa prova continua:

Primeiro, provamos que todo **membro** de $A \cup (B \cap C)$ é também **um membro** de $(A \cup B) \cap (A \cup C)$. Seja $z \in A \cup (B \cap C)$. Temos que mostrar que $z \in (A \cup B) \cap (A \cup C)$.

Agora é hora de desdobrar a definição de $\cup$ e $\cap$. Por exemplo, a definição de $\cup$ é: $A \cup B = \{z : z \in A \text{ ou } z \in B\}$. Quando aplicamos a definição a “ $A \cup (B \cap C)$,” o papel do “ B ” na definição passa a ser desempenhado por “ $B \cap C$,” de modo que $A \cup (B \cap C) = \{z : z \in A \text{ ou } z \in B \cap C\}$. Então nossa suposição de que $z \in A \cup (B \cap C)$ equivale a: $z \in \{z : z \in A \text{ ou } z \in B \cap C\}$. E $z \in \{z : \dots z \dots\}$ sse $\dots z \dots$, isto é, neste caso, $z \in A$ ou $z \in B \cap C$.

Pela definição de $\cup$, ou $z \in A$ ou $z \in B \cap C$.

74.5. UM EXEMPLO

Como isso é uma disjunção, será útil aplicar a prova por casos. Tomamos os dois casos, e mostramos que, em cada um, a conclusão que buscamos (a saber, “ $z \in (A \cup B) \cap (A \cup C)$ ”) se obtém.

Caso 1: Suponha que $z \in A$.

Não há muito mais com que trabalhar a partir das nossas suposições. Então vamos olhar para o que temos com que trabalhar na conclusão. Queremos mostrar que $z \in (A \cup B) \cap (A \cup C)$. Com base na definição de $\cap$, se queremos mostrar que $z \in (A \cup B) \cap (A \cup C)$, temos que mostrar que ele está tanto em $(A \cup B)$ quanto em $(A \cup C)$. Mas $z \in A \cup B$ sse $z \in A$ ou $z \in B$, e já temos (como a suposição do caso 1) que $z \in A$. Pelo mesmo raciocínio—trocando C por B — $z \in A \cup C$. Esse argumento foi na direção inversa, então vamos registrar nosso raciocínio na direção necessária à nossa prova.

Como $z \in A$, $z \in A$ ou $z \in B$, e logo, pela definição de $\cup$, $z \in A \cup B$. De modo semelhante, $z \in A \cup C$. Mas isso significa que $z \in (A \cup B) \cap (A \cup C)$, pela definição de $\cap$.

Isso completa o primeiro caso da prova por casos. Agora queremos derivar a conclusão no segundo caso, em que $z \in B \cap C$.

Caso 2: Suponha que $z \in B \cap C$.

Novamente, estamos trabalhando com a interseção de dois conjuntos. Vamos aplicar a definição de $\cap$:

Como $z \in B \cap C$, z deve ser **um membro** tanto de B quanto de C , pela definição de $\cap$.

É hora de olhar de novo para a nossa conclusão. Temos que mostrar que z está tanto em $(A \cup B)$ quanto em $(A \cup C)$. E, novamente, a solução é imediata.

Como $z \in B$, $z \in (A \cup B)$. Como $z \in C$, também $z \in (A \cup C)$. Então, $z \in (A \cup B) \cap (A \cup C)$.

Aqui aplicamos as definições de $\cup$ e $\cap$ novamente, mas como já relembramos essas definições, e já mostramos que, se z está em um de dois conjuntos, ele está na união deles, não precisamos ser tão explícitos no que fizemos.

Completamos o segundo caso da prova por casos, então agora podemos afirmar nossa primeira conclusão.

Então, se $z \in A \cup (B \cap C)$, então $z \in (A \cup B) \cap (A \cup C)$.

Agora queremos apenas mostrar a outra direção, que todo **membro** de $(A \cup B) \cap (A \cup C)$ é **um membro** de $A \cup (B \cap C)$. Como antes, provamos essa afirmação universal supondo que temos um **membro** arbitrário do primeiro conjunto e mostrando que ele deve estar no segundo conjunto. Vamos declarar o que estamos prestes a fazer.

Agora, suponha que $z \in (A \cup B) \cap (A \cup C)$. Queremos mostrar que $z \in A \cup (B \cap C)$.

Estamos agora trabalhando a partir da hipótese de que $z \in (A \cup B) \cap (A \cup C)$. Esperamos que não seja confuso demais estarmos usando o mesmo z aqui que na primeira parte da prova. Quando terminamos aquela parte, todas as suposições que fizemos ali deixaram de estar em vigor, então agora podemos fazer novas suposições sobre o que z é. Se isso for confuso para você, basta substituir z por uma variável diferente no que segue.

Sabemos que z está tanto em $A \cup B$ quanto em $A \cup C$, pela definição de $\cap$. E, pela definição de $\cup$, podemos desdobrar isso ainda mais em: ou $z \in A$ ou $z \in B$, e também ou $z \in A$ ou $z \in C$. Isso parece uma prova por casos de novo—exceto que o “e” a torna confusa. Você poderia pensar que isso equivale a haver três possibilidades: z está ou em A , B ou C . Mas isso seria um erro. Temos que ter cuidado, então vamos considerar cada disjunção por vez.

Pela definição de $\cap$, $z \in A \cup B$ e $z \in A \cup C$. Pela definição de $\cup$, $z \in A$ ou $z \in B$. Distinguimos casos.

Como estamos nos concentrando na primeira disjunção, ainda não obtivemos nossa segunda disjunção (do desdobramento de $A \cup C$). De fato, ainda não precisamos dela. O primeiro caso é $z \in A$, e **um membro** de um conjunto é também **um membro** da união desse conjunto com qualquer outro. Então o caso 1 é fácil:

Caso 1: Suponha que $z \in A$. Segue-se que $z \in A \cup (B \cap C)$.

Agora, para o segundo caso, $z \in B$. Aqui desdobramos o segundo $\cup$ e faremos outra prova por casos:

Caso 2: Suponha que $z \in B$. Como $z \in A \cup C$, ou $z \in A$ ou $z \in C$. Distinguimos os casos ainda mais:

Caso 2a: $z \in A$. Então, novamente, $z \in A \cup (B \cap C)$.

Ok, isso foi um pouco estranho. Na verdade, não precisamos da suposição de que $z \in B$ para este caso, mas tudo bem.

Caso 2b: $z \in C$. Então $z \in B$ e $z \in C$, de modo que $z \in B \cap C$, e consequentemente, $z \in A \cup (B \cap C)$.

74.6. OUTRO EXEMPLO

Isso conclui ambas as provas por casos, e assim terminamos a segunda metade.

Então, se $z \in (A \cup B) \cap (A \cup C)$, então $z \in A \cup (B \cap C)$. □

content/methods/proofs/example-2.tex

74.6 Outro Exemplo

math:prf:ex2:
sec

Proposição 74.9. *Se $A \subseteq C$, então $A \cup (C \setminus A) = C$.*

Prova. Suponha que $A \subseteq C$. Queremos mostrar que $A \cup (C \setminus A) = C$.

Começamos observando que esta é uma afirmação condicional. Ela está tacitamente quantificada universalmente: a proposição vale para todos os conjuntos A e C . Assim, A e C são variáveis para conjuntos arbitrários. Para provar tal afirmação, supomos o antecedente e provamos o conseqüente.

Continuamos usando a suposição de que $A \subseteq C$. Vamos desdobrar a definição de $\subseteq$: a suposição significa que todos os **membros** de A são também **membros** de C . Vamos anotar isto—é um fato importante que usaremos ao longo da prova.

Pela definição de $\subseteq$, como $A \subseteq C$, para todo z , se $z \in A$, então $z \in C$.

Desdobramos todas as definições que nos são dadas na suposição. Agora podemos passar à conclusão. Queremos mostrar que $A \cup (C \setminus A) = C$, e por isso montamos uma prova de modo semelhante ao último exemplo: mostramos que todo **membro** de $A \cup (C \setminus A)$ é também **um membro** de C e, reciprocamente, todo **membro** de C é **um membro** de $A \cup (C \setminus A)$. Podemos abreviar isso para: $A \cup (C \setminus A) \subseteq C$ e $C \subseteq A \cup (C \setminus A)$. (Aqui estamos fazendo o oposto de desdobrar uma definição, mas isso torna a prova um pouco mais fácil de ler.) Como isso é uma conjunção, temos que provar ambas as partes. Para mostrar a primeira parte, isto é, que todo **membro** de $A \cup (C \setminus A)$ é também **um membro** de C , supomos que $z \in A \cup (C \setminus A)$ para um z arbitrário e mostramos que $z \in C$. Pela definição de $\cup$, podemos concluir que $z \in A$ ou $z \in C \setminus A$ a partir de $z \in A \cup (C \setminus A)$. Você já deve estar pegando o jeito disso.

$A \cup (C \setminus A) = C$ sse $A \cup (C \setminus A) \subseteq C$ e $C \subseteq (A \cup (C \setminus A))$. Primeiro provamos que $A \cup (C \setminus A) \subseteq C$. Seja $z \in A \cup (C \setminus A)$. Então, ou $z \in A$ ou $z \in (C \setminus A)$.

Chegamos a uma disjunção, e a partir dela queremos provar que $z \in C$. Fazemos isso usando a prova por casos.

Caso 1: $z \in A$. Como, para todo z , se $z \in A$, $z \in C$, temos que $z \in C$.

Aqui usamos o fato registrado anteriormente, que decorreu da hipótese da proposição de que $A \subseteq C$. O primeiro caso está completo, e passamos ao segundo caso, $z \in (C \setminus A)$. Lembre-se de que $C \setminus A$ denota a *diferença* dos dois conjuntos, isto é, o conjunto de todos os **mem-bros** de C que não são **membros** de A . Mas qualquer **membro** de C que não esteja em A é, em particular, **um membro** de C .

Caso 2: $z \in (C \setminus A)$. Isso significa que $z \in C$ e $z \notin A$. Então, em particular, $z \in C$.

Ótimo, provamos a primeira direção. Agora a segunda direção. Aqui provamos que $C \subseteq A \cup (C \setminus A)$. Então supomos que $z \in C$ e provamos que $z \in A \cup (C \setminus A)$.

Agora seja $z \in C$. Queremos mostrar que $z \in A$ ou $z \in C \setminus A$.

Como todos os **membros** de A são também **membros** de C , e $C \setminus A$ é o conjunto de todas as coisas que são **membros** de C mas não de A , segue-se que z está ou em A ou em $C \setminus A$. Isso pode ser um pouco obscuro se você ainda não souber por que o resultado é verdadeiro. Seria melhor prová-lo passo a passo. Ajudará usar um fato simples que podemos enunciar sem prova: $z \in A$ ou $z \notin A$. Isso é chamado de “princípio do terceiro excluído:” para qualquer afirmação p , ou p é verdadeira ou sua negação é verdadeira. (Aqui, p é a afirmação de que $z \in A$.) Como isso é uma disjunção, podemos novamente usar a prova por casos.

Ou $z \in A$ ou $z \notin A$. No primeiro caso, $z \in A \cup (C \setminus A)$. No segundo caso, $z \in C$ e $z \notin A$, de modo que $z \in C \setminus A$. Mas então $z \in A \cup (C \setminus A)$.

Nossa prova está completa: mostramos que $A \cup (C \setminus A) = C$. □

`content/methods/proofs/proof-by-contradiction.tex`

74.7 Prova por Contradição

Em primeira instância, a prova por contradição é um padrão de inferência usado para provar afirmações negativas. Suponha que você queira mostrar que alguma afirmação p é *falsa*, isto é, você quer mostrar $\neg p$. A estratégia mais promissora é (a) supor que p é verdadeira, e (b) mostrar que essa suposição leva a algo que você sabe ser falso. “Algo que se sabe ser falso” pode ser um resultado que entra em conflito com—contradiz—o próprio p , ou alguma outra hipótese da afirmação geral que você está considerando. Por exemplo, uma prova de “se q então $\neg p$ ” envolve supor que q é verdadeira e provar $\neg p$ a partir disso. Se você prova $\neg p$ por contradição, isso significa supor p além

math:prf:con:
sec

74.7. PROVA POR CONTRADIÇÃO

de q . Se você consegue provar $\neg q$ a partir de p , mostrou que a suposição p leva a algo que contradiz sua outra suposição q , já que q e $\neg q$ não podem ser ambas verdadeiras. É claro que você tem que usar outros padrões de inferência em sua prova da contradição, bem como desdobrar definições. Vamos considerar um exemplo.

Proposição 74.10. *Se $A \subseteq B$ e $B = \emptyset$, então A não tem membros.*

Prova. Suponha $A \subseteq B$ e $B = \emptyset$. Queremos mostrar que A não tem membros.

Como esta é uma afirmação condicional, supomos o antecedente e queremos provar o consequente. O consequente é: A não tem membros. Podemos tornar isso um pouco mais explícito: não é o caso de que haja um $x \in A$.

A não tem membros sse não é o caso de que haja um x tal que $x \in A$.

Então determinamos que o que queremos provar é, na verdade, uma afirmação negativa $\neg p$, a saber: não é o caso de que haja um $x \in A$. Para usar a prova por contradição, temos que supor a afirmação positiva correspondente p , isto é, há um $x \in A$, e provar uma contradição a partir dela. Indicamos que estamos fazendo uma prova por contradição escrevendo “por absurdo, suponha” ou mesmo apenas “suponha que não,” e então enunciamos a suposição p .

Suponha que não: há um $x \in A$.

Esta é agora a nova suposição que usaremos para obter uma contradição. Temos mais duas suposições: que $A \subseteq B$ e que $B = \emptyset$. A primeira nos dá que $x \in B$:

Como $A \subseteq B$, $x \in B$.

Mas como $B = \emptyset$, todo membro de B (por exemplo, x) também deve ser um membro de $\emptyset$.

Como $B = \emptyset$, $x \in \emptyset$. Isso é uma contradição, já que, por definição, $\emptyset$ não tem membros.

Isso já completa a prova: chegamos ao que precisávamos (uma contradição) a partir das suposições que montamos, e isso significa que as suposições não podem ser todas verdadeiras. Como as duas primeiras suposições ($A \subseteq B$ e $B = \emptyset$) não são contestadas, deve ser a última suposição introduzida (há um $x \in A$) que tem que ser falsa. Mas, se quisermos ser minuciosos, podemos explicitar isso.

Assim, nossa suposição de que há um $x \in A$ tem que ser falsa, logo, A não tem membros, por prova por contradição. $\square$

Toda afirmação positiva é trivialmente equivalente a uma afirmação negativa: p sse $\neg\neg p$. Assim, as provas por contradição também podem ser usadas para estabelecer afirmações positivas “indiretamente,” como segue: Para provar p , leia-o como a afirmação negativa $\neg\neg p$. Se conseguirmos provar uma contradição a partir de $\neg p$, estabelecemos $\neg\neg p$ por prova por contradição, e portanto p .

No último exemplo, buscávamos provar uma afirmação negativa, a saber, que A não tem **membros**, e por isso a suposição que fizemos para os fins da prova por contradição (isto é, que há um $x \in A$) era uma afirmação positiva. Ela nos deu algo com que trabalhar, a saber, o hipotético $x \in A$ sobre o qual continuamos a raciocinar até chegarmos a $x \in \emptyset$.

Ao provar uma afirmação positiva indiretamente, a suposição que você faria para os fins da prova por contradição seria negativa. Mas muito frequentemente você pode facilmente reformular uma afirmação positiva como uma afirmação negativa, e uma afirmação negativa como uma afirmação positiva. Nossa prova anterior teria sido essencialmente a mesma se tivéssemos provado “ $A = \emptyset$ ” em vez do consequente negativo “ A não tem **membros**.” (Pela definição de $=$, “ $A = \emptyset$ ” é uma afirmação geral, já que ela se desdobra em “todo **membro** de A é **um membro** de $\emptyset$ e vice-versa”.) Mas é facilmente visto que ela é equivalente à afirmação negativa “não: há um $x \in A$.”

Então, às vezes é mais fácil trabalhar com $\neg p$ como suposição do que provar p diretamente. Mesmo quando uma prova direta é igualmente simples ou até mais simples (como nos próximos exemplos), algumas pessoas preferem proceder indiretamente. Se a dupla negação o confunde, pense em uma prova por contradição de alguma afirmação como uma prova de uma contradição a partir da afirmação *oposta*. Assim, uma prova por contradição de $\neg p$ é uma prova de uma contradição a partir da suposição p ; e a prova por contradição de p é uma prova de uma contradição a partir de $\neg p$.

Proposição 74.11. $A \subseteq A \cup B$.

Prova. Queremos mostrar que $A \subseteq A \cup B$.

À primeira vista, esta é uma afirmação positiva: todo $x \in A$ está também em $A \cup B$. A negação disso é: algum $x \in A$ está $\notin A \cup B$. Então podemos provar a afirmação indiretamente supondo essa afirmação negada, e mostrando que ela leva a uma contradição.

Suponha que não, isto é, $A \not\subseteq A \cup B$.

Temos uma definição de $A \subseteq A \cup B$: todo $x \in A$ está também $\in A \cup B$. Para entender o que $A \not\subseteq A \cup B$ significa, temos que usar alguma manipulação lógica elementar sobre a definição desdobrada: é falso que todo $x \in A$ esteja também $\in A \cup B$ sse há *algum* $x \in A$ que está $\notin A \cup B$. (Este é um ponto em que você quer ser muito cuidadoso: muitas tentativas de prova por contradição de estudantes falham porque eles analisam a negação de uma afirmação como “todos os

74.7. PROVA POR CONTRADIÇÃO

A são B ” incorretamente.) Em outras palavras, $A \not\subseteq A \cup B$ sse há um x tal que $x \in A$ e $x \notin A \cup B$. A partir daí, é fácil.

Então, há um $x \in A$ tal que $x \notin A \cup B$. Pela definição de $\cup$, $x \in A \cup B$ sse $x \in A$ ou $x \in B$. Como $x \in A$, temos $x \in A \cup B$. Isso contradiz a suposição de que $x \notin A \cup B$. $\square$

Problema 74.2. Prove *indiretamente* que $A \cap B \subseteq A$.

Proposição 74.12. Se $A \subseteq B$ e $B \subseteq C$ então $A \subseteq C$.

Prova. Suponha $A \subseteq B$ e $B \subseteq C$. Queremos mostrar $A \subseteq C$.

Vamos proceder indiretamente: supomos a negação do que queremos estabelecer.

Suponha que não, isto é, $A \not\subseteq C$.

Como antes, raciocinamos que $A \not\subseteq C$ sse nem todo $x \in A$ está também $\in C$, isto é, algum $x \in A$ está $\notin C$. Não se preocupe, com prática você não terá mais que pensar muito para desdobrar negações como esta.

Em outras palavras, há um x tal que $x \in A$ e $x \notin C$.

Agora podemos usar isso para chegar à nossa contradição. É claro que teremos que usar as outras duas suposições para fazê-lo.

Como $A \subseteq B$, $x \in B$. Como $B \subseteq C$, $x \in C$. Mas isso contradiz $x \notin C$. $\square$

Proposição 74.13. Se $A \cup B = A \cap B$ então $A = B$.

Prova. Suponha $A \cup B = A \cap B$. Queremos mostrar que $A = B$.

O começo é agora rotineiro:

Suponha, por absurdo, que $A \neq B$.

Nossa suposição para a prova por contradição é que $A \neq B$. Como $A = B$ sse $A \subseteq B$ e $B \subseteq A$, obtemos que $A \neq B$ sse $A \not\subseteq B$ ou $B \not\subseteq A$. (Note como é importante ter cuidado ao manipular negações!) Para provar uma contradição a partir dessa disjunção, usamos uma prova por casos e mostramos que, em cada caso, uma contradição se segue.

$A \neq B$ sse $A \not\subseteq B$ ou $B \not\subseteq A$. Distinguimos casos.

No primeiro caso, supomos $A \not\subseteq B$, isto é, para algum x , $x \in A$ mas $x \notin B$. $A \cap B$ é definido como aqueles **membros** que A e B têm em comum, então, se algo não está em um deles, não está na interseção. $A \cup B$ é A junto com B , então qualquer coisa em um ou outro está também na união. Isso nos diz que $x \in A \cup B$ mas $x \notin A \cap B$, e logo que $A \cap B \neq A \cup B$.

Caso 1: $A \not\subseteq B$. Então, para algum x , $x \in A$ mas $x \notin B$. Como $x \notin B$, então $x \notin A \cap B$. Como $x \in A$, $x \in A \cup B$. Então, $A \cap B \neq A \cup B$, contradizendo a suposição de que $A \cap B = A \cup B$.

Caso 2: $B \not\subseteq A$. Então, para algum y , $y \in B$ mas $y \notin A$. Como antes, temos $y \in A \cup B$ mas $y \notin A \cap B$, e assim $A \cap B \neq A \cup B$, novamente contradizendo $A \cap B = A \cup B$. $\square$

content/methods/proofs/reading-proofs.tex

74.8 Lendo Provas

As provas que você encontra em livros-texto e artigos muito raramente fornecem todos os detalhes que incluímos até agora em nossos exemplos. Os autores frequentemente não chamam a atenção para quando distinguem casos, quando dão uma prova indireta, ou não mencionam que usam uma definição. Assim, quando você lê uma prova em um livro-texto, muitas vezes terá de preencher esses detalhes por conta própria para entender a prova. Fazer isso também é um bom exercício para pegar o jeito dos vários passos que você tem de dar em uma prova. Vejamos um exemplo.

mth:prf:rea:
sec

Proposição 74.14 (Absorção). *Para todos os conjuntos A , B ,*

$$A \cap (A \cup B) = A$$

Prova. Se $z \in A \cap (A \cup B)$, então $z \in A$, logo $A \cap (A \cup B) \subseteq A$. Agora suponha $z \in A$. Então também $z \in A \cup B$, e portanto também $z \in A \cap (A \cup B)$. $\square$

A prova precedente da lei da absorção é muito condensada. Não há menção a quaisquer definições usadas, nenhum “temos de provar que” antes de prová-lo, etc. Vamos desempacotá-la. A proposição provada é uma afirmação geral sobre quaisquer conjuntos A e B , e quando a prova menciona A ou B , estes são variáveis para conjuntos arbitrários. As afirmações gerais que a prova estabelece são o que se exige para provar a identidade de conjuntos, isto é, que todo **membro** do lado esquerdo da identidade é **um membro** do direito e vice-versa.

“Se $z \in A \cap (A \cup B)$, então $z \in A$, logo $A \cap (A \cup B) \subseteq A$.”

Esta é a primeira metade da prova da identidade: ela estabelece que, se um z arbitrário é **um membro** do lado esquerdo, ele também é **um membro** do direito, isto é, $A \cap (A \cup B) \subseteq A$. Assuma que $z \in A \cap (A \cup B)$. Como z é **um membro** da interseção de dois conjuntos se, e somente se, é **um membro** de ambos os conjuntos, podemos concluir que $z \in A$ e também $z \in A \cup B$. Em particular, $z \in A$, que é o que queríamos mostrar. Como isso é tudo o que precisa ser feito para a primeira metade, sabemos que o restante da prova deve ser uma prova da segunda metade, isto é, uma prova de que $A \subseteq A \cap (A \cup B)$.

74.9. NÃO CONSIGO FAZER ISSO!

“Agora suponha $z \in A$. Então também $z \in A \cup B$, e portanto também $z \in A \cap (A \cup B)$.”

Começamos assumindo que $z \in A$, já que estamos mostrando que, para qualquer z , se $z \in A$ então $z \in A \cap (A \cup B)$. Para mostrar que $z \in A \cap (A \cup B)$, temos de mostrar (pela definição de “ $\cap$ ”) que (i) $z \in A$ e também (ii) $z \in A \cup B$. Aqui (i) é justamente nossa suposição, então não há mais nada a provar, e é por isso que a prova não a menciona novamente. Para (ii), lembre-se de que z é **um membro** de uma união de conjuntos se, e somente se, é **um membro** de pelo menos um desses conjuntos. Como $z \in A$, e $A \cup B$ é a união de A e B , este é o caso aqui. Então $z \in A \cup B$. Mostramos ambos (i) $z \in A$ e (ii) $z \in A \cup B$, logo, pela definição de “ $\cap$,” $z \in A \cap (A \cup B)$. A prova não menciona essas definições; presume-se que o leitor já as tenha internalizado. Se você não as internalizou, terá de voltar e relembrar quais são. Então também terá de reconhecer por que decorre de $z \in A$ que $z \in A \cup B$, e de $z \in A$ e $z \in A \cup B$ que $z \in A \cap (A \cup B)$.

Eis outra versão da prova acima, com tudo tornado explícito:

Prova. [Pela definição de $=$ para conjuntos, $A \cap (A \cup B) = A$, temos de mostrar (a) $A \cap (A \cup B) \subseteq A$ e (b) $A \cap (A \cup B) \subseteq A$. (a): Pela definição de $\subseteq$, temos de mostrar que se $z \in A \cap (A \cup B)$, então $z \in A$.] Se $z \in A \cap (A \cup B)$, então $z \in A$ [já que pela definição de $\cap$, $z \in A \cap (A \cup B)$ se, e somente se, $z \in A$ e $z \in A \cup B$], logo $A \cap (A \cup B) \subseteq A$. [(b): Pela definição de $\subseteq$, temos de mostrar que se $z \in A$, então $z \in A \cap (A \cup B)$.] Agora suponha [(1)] $z \in A$. Então também [(2)] $z \in A \cup B$ [já que por (1) $z \in A$ ou $z \in B$, o que pela definição de $\cup$ significa $z \in A \cup B$], e portanto também $z \in A \cap (A \cup B)$ [já que a definição de $\cap$ requer que $z \in A$, isto é, (1), e $z \in A \cup B$, isto é, (2)]. $\square$

Problema 74.3. Expanda a seguinte prova de $A \cup (A \cap B) = A$, mencionando todos os padrões de inferência usados, por que cada passo decorre de suposições ou afirmações estabelecidas antes dele, e onde temos de apelar para quais definições.

Prova. Se $z \in A \cup (A \cap B)$ então $z \in A$ ou $z \in A \cap B$. Se $z \in A \cap B$, $z \in A$. Qualquer $z \in A$ está também $\in A \cup (A \cap B)$. $\square$

content/methods/proofs/cant-do-it.tex

74.9 Não Consigo Fazer Isso!

math:prf:cnt:
sec

Todos nós chegamos a um ponto em que sentimos vontade de desistir. Mas você *consegue* fazer isso. Seu professor e o monitor, bem como seus colegas, podem ajudar. Peça ajuda a eles! Aqui estão algumas dicas para ajudá-lo a evitar uma crise, e o que fazer se você sentir vontade de desistir.

Para garantir que você consiga resolver os problemas com sucesso, faça o seguinte:

1. *Comece com a maior antecedência possível.* Ficamos ocupados ao longo do semestre e muitos de nós lutam com a procrastinação; uma das melhores coisas que você pode fazer é começar as tarefas de casa cedo. Dessa forma, se você travar, terá tempo para procurar uma solução (que não seja chorar).
2. *Converse com seus colegas de turma.* Você não está sozinho. Outros na turma também podem ter dificuldades—mas podem ter dificuldades com coisas diferentes. Conversar sobre o assunto com seus colegas pode lhe dar uma perspectiva diferente sobre o problema que pode levar a um avanço. É claro, não copie simplesmente a solução deles: peça uma dica, ou explique onde você travou e peça o próximo passo. E quando você conseguir, retribua. Ajudar outra pessoa, e explicar as coisas, ajudará você a entender melhor também.
3. *Peça ajuda.* Você tem muitos recursos à sua disposição—seu professor e o monitor estão lá por você e *querem* que você tenha sucesso. Eles devem ser capazes de ajudá-lo a resolver um problema e identificar em que ponto do processo você está com dificuldade.
4. *Faça uma pausa.* Se você travar, *pode* ser porque ficou olhando para o problema por tempo demais. Faça uma pausa curta, tome uma xícara de chá, ou trabalhe em um problema diferente por um tempo, depois volte ao problema com a mente renovada. Durma sobre o assunto.

Percebe como essas estratégias exigem que você tenha começado a trabalhar na prova com bastante antecedência? Se você começou a prova às 2 da madrugada do dia anterior à entrega, elas talvez não sejam tão úteis.

Isso pode soar como pessimismo e desgraça, mas resolver uma prova é um desafio que compensa no final. Algumas pessoas fazem isso como profissão—portanto deve haver algo de prazeroso nisso. Como praticamente tudo, resolver problemas e fazer provas é algo que exige prática. Você pode ver colegas que acham isso fácil: provavelmente eles apenas já tiveram muita prática. Tente não ceder com facilidade demais.

Se você ficar mesmo sem tempo (ou sem paciência) em um problema específico: tudo bem. Isso não significa que você seja burro ou que nunca vá conseguir. Descubra (com seu professor ou outro estudante) como se faz, e identifique onde você errou ou travou, para poder evitar fazer isso da próxima vez que encontrar uma questão semelhante. Depois tente fazê-lo sem olhar a solução. E na próxima vez, comece (e peça ajuda) mais cedo.

`content/methods/proofs/resources.tex`

74.10 Outros Recursos

Há muitos livros sobre como fazer provas em matemática que podem ser úteis. Confira em particular *How to Read and do Proofs: An Introduction to Mathe-*

mathematical Thought Processes (Solow, 2013) e *How to Prove It: A Structured Approach* (Velleman, 2019). O *Book of Proof* (Hammack, 2013) e o *Mathematical Reasoning* (Sandstrum, 2019) são livros sobre provas que estão disponíveis gratuitamente online. Filósofos podem achar *More Precisely: The Math you need to do Philosophy* (Steinhart, 2018) uma boa introdução ao raciocínio matemático.

Há também vários guias mais curtos sobre provas disponíveis na internet; por exemplo, “Introduction to Mathematical Arguments” (Hutchings, 2003) e “How to write proofs” (Cheng, 2004).

Vídeos Motivacionais

Sente que não tem motivação para fazer sua tarefa de casa? Desanimado? Estes vídeos podem ajudar!

- https://www.youtube.com/watch?v=ZXsQAXx_ao0
- <https://www.youtube.com/watch?v=BQ4yd2W50No>
- <https://www.youtube.com/watch?v=StTqXEQ21-Y>

Capítulo 75

Indução

`content/methods/induction/introduction.tex`

75.1 Introdução

`math:ind:int:`
`sec`

A indução é uma técnica de prova importante, usada, em diferentes formas, em quase todas as áreas da lógica, da ciência da computação teórica e da matemática. Ela é necessária para provar muitos dos resultados em lógica.

A indução é frequentemente contrastada com a dedução, e caracterizada como a inferência do particular para o geral. Por exemplo, se observamos muitas esmeraldas verdes, e nada que chamaríamos de esmeralda que não seja verde, poderíamos concluir que todas as esmeraldas são verdes. Essa é uma

inferência indutiva, no sentido de que procede de muitos casos particulares (esta esmeralda é verde, aquela esmeralda é verde, etc.) a uma afirmação geral (todas as esmeraldas são verdes). A indução *matemática* também é uma inferência que conclui uma afirmação geral, mas é de um tipo muito diferente dessa “indução simples.”

De modo bem aproximado, uma prova indutiva em matemática conclui que todos os objetos matemáticos de certo tipo têm uma certa propriedade. No caso mais simples, os objetos matemáticos com que uma prova indutiva se ocupa são os números naturais. Nesse caso, uma prova indutiva é usada para estabelecer que todos os números naturais têm alguma propriedade, e faz isso mostrando que

1. 0 tem a propriedade, e
2. sempre que um número k tem a propriedade, também $k + 1$ a tem.

A indução sobre os números naturais pode então, também, ser frequentemente usada para provar afirmações gerais sobre objetos matemáticos aos quais se podem atribuir números. Por exemplo, conjuntos finitos têm, cada um, um número finito n de elementos, e se pudermos usar a indução para mostrar que todo número n tem a propriedade “todos os conjuntos finitos de tamanho n são ...” então teremos mostrado algo sobre todos os conjuntos finitos.

A indução também pode ser generalizada para objetos matemáticos que são *definidos indutivamente*. Por exemplo, as expressões de uma linguagem formal, como as da lógica de primeira ordem, são definidas indutivamente. A *indução estrutural* é um modo de provar resultados sobre todas essas expressões. A indução estrutural, em particular, é muito útil—e amplamente usada—em lógica.

`content/methods/induction/induction-on-N.tex`

75.2 Indução em $\mathbb{N}$

Em sua forma mais simples, a indução é uma técnica usada para provar resultados para todos os números naturais. Ela usa o fato de que, começando em 0 e somando 1 repetidamente, eventualmente alcançamos todo número natural. Assim, para provar que algo é verdadeiro para todo número, podemos (1) estabelecer que é verdadeiro para 0 e (2) mostrar que, sempre que é verdadeiro para um número n , também é verdadeiro para o número seguinte $n + 1$. Se abreviarmos “o número n tem a propriedade P ” por $P(n)$ (e “o número k tem a propriedade P ” por $P(k)$, etc.), então uma prova por indução de que $P(n)$ para todo $n \in \mathbb{N}$ consiste em:

1. uma prova de $P(0)$, e
2. uma prova de que, para qualquer k , se $P(k)$ então $P(k + 1)$.

mth:ind:inN:
sec

75.2. INDUÇÃO EM $\mathbb{N}$

Para deixar isso bem claro, suponha que tenhamos tanto (1) quanto (2). Então (1) nos diz que $P(0)$ é verdadeiro. Se também tivermos (2), sabemos em particular que, se $P(0)$, então $P(0+1)$, isto é, $P(1)$. Isso decorre da afirmação geral “para qualquer k , se $P(k)$ então $P(k+1)$ ” ao pôr 0 no lugar de k . Então, por modus ponens, temos que $P(1)$. De (2) novamente, agora tomando 1 por n , temos: se $P(1)$ então $P(2)$. Como acabamos de estabelecer $P(1)$, por modus ponens, temos $P(2)$. E assim por diante. Para qualquer número n , depois de fazer isso n vezes, eventualmente chegamos a $P(n)$. Então (1) e (2) juntos estabelecem $P(n)$ para qualquer $n \in \mathbb{N}$.

Vejamos um exemplo. Suponha que queiramos descobrir quantas somas diferentes podemos lançar com n dados. Embora possa parecer bobo, vamos começar com 0 dados. Se você não tem dados, há apenas uma soma possível que você pode “lançar”: nenhum ponto, que soma 0. Assim, o número de lançamentos possíveis diferentes é 1. Se você tem apenas um dado, isto é, $n = 1$, há seis valores possíveis, de 1 a 6. Com dois dados, podemos lançar qualquer soma de 2 a 12, ou seja, 11 possibilidades. Com três dados, podemos lançar qualquer número de 3 a 18, isto é, 16 possibilidades diferentes. 1, 6, 11, 16: parece um padrão: talvez a resposta seja $5n + 1$? É claro que $5n + 1$ é o máximo possível, pois há apenas $5n + 1$ números entre n , o menor valor que você pode lançar com n dados (todos 1) e $6n$, o maior que você pode lançar (todos 6).

Teorema 75.1. *Com n dados pode-se lançar todos os $5n + 1$ valores possíveis entre n e $6n$.*

Prova. Seja $P(n)$ a afirmação: “É possível lançar qualquer número entre n e $6n$ usando n dados.” Para usar a indução, provamos:

1. A *base da indução* $P(1)$, isto é, com apenas um dado, você pode lançar qualquer número entre 1 e 6.
2. O *passo da indução*, para todo k , se $P(k)$ então $P(k+1)$.

(1) é provado inspecionando um dado de 6 lados. Ele tem todos os 6 lados, e todo número entre 1 e 6 aparece em um dos lados. Então é possível lançar qualquer número entre 1 e 6 usando um único dado.

Para provar (2), supomos o antecedente do condicional, isto é, $P(k)$. Essa suposição é chamada de *hipótese indutiva*. Nós a usamos para provar $P(k+1)$. A parte difícil é encontrar um modo de pensar sobre os valores possíveis de um lançamento de $k+1$ dados em termos dos valores possíveis de lançamentos de k dados, somados aos lançamentos do dado extra, o $(k+1)$ -ésimo—é isso, porém, que temos que fazer, se quisermos usar a hipótese indutiva.

A hipótese indutiva diz que podemos obter qualquer número entre k e $6k$ usando k dados. Se lançamos um 1 com nosso $(k+1)$ -ésimo dado, isso acrescenta 1 ao total. Então podemos lançar qualquer valor entre $k+1$ e $6k+1$ lançando k dados e depois rolando um 1 com o $(k+1)$ -ésimo dado. O que falta? Os valores de $6k+2$ a $6k+6$. Podemos obtê-los rolando k dados em

6 e depois um número entre 2 e 6 com nosso $(k + 1)$ -ésimo dado. Juntos, isso significa que, com $k + 1$ dados, podemos lançar qualquer um dos números entre $k + 1$ e $6(k + 1)$, isto é, provamos $P(k + 1)$ usando a suposição $P(k)$, a hipótese indutiva. $\square$

Muito frequentemente usamos a indução quando queremos provar algo sobre uma série de objetos (números, conjuntos, etc.) que é, ela própria, definida “indutivamente,” isto é, definindo o $(n + 1)$ -ésimo objeto em termos do n -ésimo. Por exemplo, podemos definir a soma s_n dos números naturais até n por

$$\begin{aligned}s_0 &= 0 \\ s_{n+1} &= s_n + (n + 1)\end{aligned}$$

Esta definição dá:

$$\begin{aligned}s_0 &= 0, \\ s_1 &= s_0 + 1 &= 1, \\ s_2 &= s_1 + 2 &= 1 + 2 = 3 \\ s_3 &= s_2 + 3 &= 1 + 2 + 3 = 6, \text{ etc.}\end{aligned}$$

Agora podemos provar, por indução, que $s_n = n(n + 1)/2$.

Proposição 75.2. $s_n = n(n + 1)/2$.

Prova. Temos que provar (1) que $s_0 = 0 \cdot (0 + 1)/2$ e (2) se $s_k = k(k + 1)/2$ então $s_{k+1} = (k + 1)(k + 2)/2$. (1) é óbvio. Para provar (2), supomos a hipótese indutiva: $s_k = k(k + 1)/2$. Usando-a, temos que mostrar que $s_{k+1} = (k + 1)(k + 2)/2$.

O que é s_{k+1} ? Pela definição, $s_{k+1} = s_k + (k + 1)$. Pela hipótese indutiva, $s_k = k(k + 1)/2$. Podemos substituir isso na equação anterior, e então só precisamos de um pouco de aritmética de frações:

$$\begin{aligned}s_{k+1} &= \frac{k(k + 1)}{2} + (k + 1) = \\ &= \frac{k(k + 1)}{2} + \frac{2(k + 1)}{2} = \\ &= \frac{k(k + 1) + 2(k + 1)}{2} = \\ &= \frac{(k + 2)(k + 1)}{2}.\end{aligned}\quad \square$$

A lição importante aqui é que, se você está provando algo sobre alguma sequência definida indutivamente a_n , a indução é o caminho óbvio a seguir. E, mesmo quando não é (como no caso das possibilidades de lançamentos de dados), você pode usar a indução se conseguir, de alguma forma, relacionar o caso para $k + 1$ ao caso para k .

75.3. INDUÇÃO FORTE

75.3 Indução Forte

meth:ind:str:
sec No princípio de indução discutido acima, provamos $P(0)$ e também que, se $P(k)$, então $P(k+1)$. Na segunda parte, assumimos que $P(k)$ é verdadeiro e usamos essa suposição para provar $P(k+1)$. De forma equivalente, é claro, poderíamos assumir $P(k-1)$ e usá-la para provar $P(k)$ —o importante é que sejamos capazes de realizar a inferência de qualquer número para seu sucessor; que possamos provar a afirmação em questão para qualquer número sob a suposição de que ela vale para seu antecessor.

Há uma variante do princípio de indução em que não apenas assumimos que a afirmação vale para o antecessor $k-1$ de k , mas para todos os números menores que k , e usamos essa suposição para estabelecer a afirmação para k . Isso também nos dá a afirmação $P(n)$ para todo $n \in \mathbb{N}$. Pois, uma vez que tenhamos estabelecido $P(0)$, teremos com isso estabelecido que P vale para todos os números menores que 1. E se sabemos que, se $P(l)$ para todo $l < k$, então $P(k)$, sabemos isso em particular para $k=1$. Então podemos concluir $P(1)$. Com isso provamos $P(0)$ e $P(1)$, isto é, $P(l)$ para todo $l < 2$, e como temos também o condicional, se $P(l)$ para todo $l < 2$, então $P(2)$, podemos concluir $P(2)$, e assim por diante.

De fato, se conseguirmos estabelecer o condicional geral “para todo k , se $P(l)$ para todo $l < k$, então $P(k)$,” não temos mais de estabelecer $P(0)$, já que ele decorre disso. Pois lembre-se de que uma afirmação geral como “para todo $l < k$, $P(l)$ ” é verdadeira se não houver nenhum $l < k$. Este é um caso de quantificação vazia: “todos os As são Bs ” é verdadeiro se não houver nenhum A , $\forall x (\varphi(x) \rightarrow \psi(x))$ é verdadeiro se nenhum x satisfaz $\varphi(x)$. Neste caso, a versão formalizada seria “ $\forall l (l < k \rightarrow P(l))$ ”—e isso é verdadeiro se não houver nenhum $l < k$. E se $k=0$ esse é exatamente o caso: não há nenhum $l < 0$, logo “para todo $l < 0$, $P(l)$ ” é verdadeiro, qualquer que seja P . Uma prova de “se $P(l)$ para todo $l < k$, então $P(k)$ ” estabelece assim automaticamente $P(0)$.

Esta variante é útil se estabelecer a afirmação para k não puder ser feito de modo a apenas se apoiar na afirmação para $k-1$, mas puder exigir a suposição de que ela é verdadeira para um ou mais $l < k$.

`content/methods/induction/inductive-definitions.tex`

75.4 Definições Indutivas

meth:ind:idf:
sec Em lógica, muito frequentemente definimos tipos de objetos *indutivamente*, isto é, especificando regras para o que conta como um objeto do tipo a ser definido, que explicam como obter novos objetos desse tipo a partir de objetos antigos desse tipo. Por exemplo, frequentemente definimos tipos especiais de seqüências de símbolos, como os termos e **fórmulas** de uma linguagem, por indução. Para um exemplo simples, considere cadeias compostas das letras a, b, c, d , do símbolo $\circ$, e dos colchetes $[$ e $]$, como “ $[[c \circ d]]$ ”, “ $[a[]\circ]$ ”, “ a ” ou “ $[[a \circ b] \circ d]$ ”. Você provavelmente sente que há algo “errado” com as duas

primeiras cadeias: os colchetes não “se equilibram” de modo algum na primeira, e você pode sentir que o “o” deveria “conectar” expressões que elas próprias façam sentido. A terceira e a quarta cadeia parecem melhores: para todo “[” há um “]” de fechamento (se houver algum), e para qualquer o podemos encontrar expressões “bonitas” de cada lado, cercadas por um par de parênteses.

Gostaríamos de especificar precisamente o que conta como um “termo bonito.” Em primeiro lugar, toda letra isolada é bonita. Qualquer coisa que não seja apenas uma letra isolada deve ser da forma “[$t \circ s$]” em que s e t são, eles próprios, bonitos. Reciprocamente, se t e s são bonitos, então podemos formar um novo termo bonito pondo um $\circ$ entre eles e cercando-os por um par de colchetes. Poderíamos usar essas operações para *definir* o conjunto dos termos bonitos. Esta é uma *definição indutiva*.

Definição 75.3 (Termos bonitos). O conjunto dos *termos bonitos* é definido indutivamente como segue:

1. Qualquer letra a, b, c, d é um termo bonito.
2. Se s_1 e s_2 são termos bonitos, então também o é $[s_1 \circ s_2]$.
3. Nada mais é um termo bonito.

Esta definição nos diz que algo conta como um termo bonito sse ele pode ser construído de acordo com as duas condições (1) e (2) em algum número finito de passos. No primeiro passo, construímos todos os termos bonitos compostos apenas por letras isoladas, isto é,

$$a, b, c, d$$

No segundo passo, aplicamos (2) aos termos que construímos. Obteremos

$$[a \circ a], [a \circ b], [b \circ a], \dots, [d \circ d]$$

para todas as combinações de duas letras. No terceiro passo, aplicamos (2) novamente, a quaisquer dois termos bonitos que construímos até agora. Obtemos novos termos bonitos como $[a \circ [a \circ a]]$ —em que t é a do passo 1 e s é $[a \circ a]$ do passo 2—e $[[b \circ c] \circ [d \circ b]]$ construído a partir dos dois termos $[b \circ c]$ e $[d \circ b]$ do passo 2. E assim por diante. A cláusula (3) impede que qualquer coisa não construída desse modo se infiltre no conjunto dos termos bonitos.

Note que ainda não provamos que toda sequência de símbolos que “parece” bonita é bonita de acordo com esta definição. No entanto, deve ficar claro que tudo o que podemos construir de fato “parece bonito”: os colchetes estão equilibrados, e $\circ$ conecta partes que são, elas próprias, bonitas.

A característica-chave das definições indutivas é que, se você quer provar algo sobre todos os termos bonitos, a definição lhe diz quais casos você deve considerar. Por exemplo, se lhe é dito que t é um termo bonito, a definição indutiva lhe diz com o que t pode se parecer: t pode ser uma letra, ou pode ser $[s_1 \circ s_2]$ para algum par de termos bonitos s_1 e s_2 . Por causa da cláusula (3), essas são as únicas possibilidades.

75.4. DEFINIÇÕES INDUTIVAS

Ao provar afirmações sobre todo um conjunto definido indutivamente, a forma forte da indução torna-se particularmente importante. Por exemplo, suponha que queiramos provar que, para todo termo bonito de comprimento n , o número de $[$ nele é $< n/2$. Isso pode ser visto como uma afirmação sobre todos os n : para todo n , o número de $[$ em qualquer termo bonito de comprimento n é $< n/2$.

Proposição 75.4. *Para qualquer n , o número de $[$ em um termo bonito de comprimento n é $< n/2$.*

Prova. Para provar esse resultado por indução (forte), temos que mostrar que a seguinte afirmação condicional é verdadeira:

Se, para todo $l < k$, qualquer termo bonito de comprimento l tem $< l/2$ $[$, então qualquer termo bonito de comprimento k tem $< k/2$ $[$.

Para mostrar este condicional, suponha que seu antecedente é verdadeiro, isto é, suponha que, para qualquer $l < k$, termos bonitos de comprimento l contêm $< l/2$ $[$. Chamamos essa suposição de hipótese indutiva. Queremos mostrar que o mesmo vale para termos bonitos de comprimento k .

Então suponha que t é um termo bonito de comprimento k . Como os termos bonitos são definidos indutivamente, temos dois casos: (1) t é uma letra isolada, ou (2) t é $[s_1 \circ s_2]$ para alguns termos bonitos s_1 e s_2 .

1. t é uma letra. Então $k = 1$, e o número de $[$ em t é 0. Como $0 < 1/2$, a afirmação vale.
2. t é $[s_1 \circ s_2]$ para alguns termos bonitos s_1 e s_2 . Sejam l_1 o comprimento de s_1 e l_2 o comprimento de s_2 . Então o comprimento k de t é $l_1 + l_2 + 3$ (os comprimentos de s_1 e s_2 mais três símbolos $[$, $\circ$, $]$). Como $l_1 + l_2 + 3$ é sempre maior que l_1 , $l_1 < k$. De modo semelhante, $l_2 < k$. Isso significa que a hipótese de indução se aplica aos termos s_1 e s_2 : o número m_1 de $[$ em s_1 é $< l_1/2$, e o número m_2 de $[$ em s_2 é $< l_2/2$.

O número de $[$ em t é o número de $[$ em s_1 , mais o número de $[$ em s_2 , mais 1, isto é, é $m_1 + m_2 + 1$. Como $m_1 < l_1/2$ e $m_2 < l_2/2$, temos:

$$m_1 + m_2 + 1 < \frac{l_1}{2} + \frac{l_2}{2} + 1 = \frac{l_1 + l_2 + 2}{2} < \frac{l_1 + l_2 + 3}{2} = k/2.$$

Em cada caso, mostramos que o número de $[$ em t é $< k/2$ (com base na hipótese indutiva). Por indução forte, a proposição se segue. $\square$

Problema 75.1. Defina o conjunto dos termos superbonitos por

1. Qualquer letra a , b , c , d é um termo superbonito.
2. Se s é um termo superbonito, então também o é $[s]$.

3. Se s_1 e s_2 são termos superbonitos, então também o é $[s_1 \circ s_2]$.
4. Nada mais é um termo superbonito.

Mostre que o número de $[$ em um termo superbonito t de comprimento n é $\leq n/2 + 1$.

content/methods/induction/structural-induction.tex

75.5 Indução Estrutural

Até aqui usamos a indução para estabelecer resultados sobre todos os números naturais. Mas um princípio correspondente pode ser usado diretamente para provar resultados sobre todos os **membros** de um conjunto definido indutivamente. Isso é frequentemente chamado de indução *estrutural*, porque depende da estrutura dos objetos definidos indutivamente. mth:ind:sti:
sec

Em geral, uma definição indutiva é dada por (a) uma lista de **membros** “iniciais” do conjunto e (b) uma lista de operações que produzem novos **membros** do conjunto a partir de antigos. No caso dos termos bonitos, por exemplo, os objetos iniciais são as letras. Temos apenas uma operação: as operações são

$$o(s_1, s_2) = [s_1 \circ s_2]$$

Você pode até pensar nos próprios números naturais $\mathbb{N}$ como sendo dados por uma definição indutiva: o objeto inicial é 0, e a operação é a função sucessor $x + 1$.

Para provar algo sobre todos os elementos de um conjunto definido indutivamente, isto é, que todo **membro** do conjunto tem uma propriedade P , devemos:

1. Provar que os objetos iniciais têm P
2. Provar que, para cada operação o , se os argumentos têm P , o resultado também tem.

Por exemplo, para provar algo sobre todos os termos bonitos, nós provaríamos que é verdadeiro para todas as letras, e que é verdadeiro para $[s_1 \circ s_2]$ desde que seja verdadeiro de s_1 e s_2 individualmente.

Proposição 75.5. *O número de $[$ é igual ao número de $]$ em qualquer termo bonito t .*

Prova. Usamos a indução estrutural. Os termos bonitos são definidos indutivamente, com as letras como objetos iniciais e a operação o para construir novos termos bonitos a partir de antigos.

1. A afirmação é verdadeira para toda letra, já que o número de $[$ em uma letra isolada é 0 e o número de $]$ nela também é 0.

75.5. INDUÇÃO ESTRUTURAL

2. Suponha que o número de $[$ em s_1 é igual ao número de $]$, e o mesmo é verdadeiro para s_2 . O número de $[$ em $o(s_1, s_2)$, isto é, em $[s_1 \circ s_2]$, é a soma do número de $[$ em s_1 e s_2 mais um. O número de $]$ em $o(s_1, s_2)$ é a soma do número de $]$ em s_1 e s_2 mais um. Assim, o número de $[$ em $o(s_1, s_2)$ é igual ao número de $]$ em $o(s_1, s_2)$. $\square$

Problema 75.2. Prove, por indução estrutural, que nenhum termo bonito começa com $]$.

Vamos dar outra prova por indução estrutural: um segmento inicial próprio de uma cadeia t de símbolos é qualquer cadeia s que concorda com t símbolo por símbolo, lida da esquerda, mas t é mais longa. Assim, por exemplo, $[a \circ$ é um segmento inicial próprio de $[a \circ b]$, mas nem $[b \circ$ (discordam no segundo símbolo) nem $[a \circ b]$ (têm o mesmo comprimento) o são.

meth:ind:sti: **Proposição 75.6.** *prop:initial* *Todo segmento inicial próprio de um termo bonito t tem mais $[$ do que $]$.*

Prova. Por indução em t :

1. t é uma letra isolada: Então t não tem segmentos iniciais próprios.
2. $t = [s_1 \circ s_2]$ para alguns termos bonitos s_1 e s_2 . Se r é um segmento inicial próprio de t , há várias possibilidades:
 - a) r é apenas $[$: Então r tem um $[$ a mais do que $]$.
 - b) r é $[r_1$ em que r_1 é um segmento inicial próprio de s_1 : Como s_1 é um termo bonito, por hipótese de indução, r_1 tem mais $[$ do que $]$, e o mesmo é verdadeiro para $[r_1$.
 - c) r é $[s_1$ ou $[s_1 \circ$: Pelo resultado anterior, o número de $[$ e $]$ em s_1 é igual; então o número de $[$ em $[s_1$ ou $[s_1 \circ$ é um a mais do que o número de $]$.
 - d) r é $[s_1 \circ r_2$ em que r_2 é um segmento inicial próprio de s_2 : Por hipótese de indução, r_2 contém mais $[$ do que $]$. Pelo resultado anterior, o número de $[$ e de $]$ em s_1 é igual. Então o número de $[$ em $[s_1 \circ r_2$ é maior do que o número de $]$.
 - e) r é $[s_1 \circ s_2$: Pelo resultado anterior, o número de $[$ e $]$ em s_1 é igual, e o mesmo para s_2 . Então há um $[$ a mais em $[s_1 \circ s_2$ do que há $]$. $\square$

75.6 Relações e Funções

Quando definimos um conjunto de objetos (como os números naturais ou os termos bonitos) indutivamente, também podemos definir *relações sobre* esses objetos por indução. Por exemplo, considere a seguinte ideia: um termo bonito t_1 é um subtermo de um termo bonito t_2 se ocorre como parte dele. Vamos usar um símbolo para isso: $t_1 \sqsubseteq t_2$. Todo termo bonito é, é claro, um subtermo de si mesmo: $t \sqsubseteq t$. Podemos dar uma definição indutiva dessa relação como segue:

Definição 75.7. A relação de um termo bonito t_1 ser um subtermo de t_2 , $t_1 \sqsubseteq t_2$, é definida por indução em t_2 como segue:

1. Se t_2 é uma letra, então $t_1 \sqsubseteq t_2$ sse $t_1 = t_2$.
2. Se t_2 é $[s_1 \circ s_2]$, então $t_1 \sqsubseteq t_2$ sse $t_1 = t_2$, $t_1 \sqsubseteq s_1$, ou $t_1 \sqsubseteq s_2$.

Esta definição, por exemplo, nos dirá que $a \sqsubseteq [b \circ a]$. Pois (2) diz que $a \sqsubseteq [b \circ a]$ sse $a = [b \circ a]$, ou $a \sqsubseteq b$, ou $a \sqsubseteq a$. As duas primeiras são falsas: a claramente não é idêntico a $[b \circ a]$, e por (1), $a \sqsubseteq b$ sse $a = b$, o que também é falso. No entanto, também por (1), $a \sqsubseteq a$ sse $a = a$, o que é verdadeiro.

É importante observar que o sucesso desta definição depende de um fato que ainda não provamos: todo termo bonito t ou é uma letra isolada, ou existem termos bonitos *univocamente determinados* s_1 e s_2 tais que $t = [s_1 \circ s_2]$. “Univocamente determinados” aqui significa que, se $t = [s_1 \circ s_2]$, ele não é *também* $= [r_1 \circ r_2]$ com $s_1 \neq r_1$ ou $s_2 \neq r_2$. Se este fosse o caso, então a cláusula (2) poderia entrar em conflito consigo mesma: lendo t_2 como $[s_1 \circ s_2]$ poderíamos obter $t_1 \sqsubseteq t_2$, mas se lermos t_2 como $[r_1 \circ r_2]$ poderíamos obter não $t_1 \sqsubseteq t_2$. Antes de provarmos que isso não pode acontecer, vejamos um exemplo em que ele *pode* acontecer.

Definição 75.8. Defina *termos sem colchetes* indutivamente por

1. Toda letra é um termo sem colchetes.
2. Se s_1 e s_2 são termos sem colchetes, então $s_1 \circ s_2$ é um termo sem colchetes.
3. Nada mais é um termo sem colchetes.

Termos sem colchetes são, por exemplo, a , $b \circ d$, $b \circ a \circ b$. Agora, se definíssemos “subtermo” para termos sem colchetes do modo como fizemos acima, a segunda cláusula diria

Se $t_2 = s_1 \circ s_2$, então $t_1 \sqsubseteq t_2$ sse $t_1 = t_2$, $t_1 \sqsubseteq s_1$, ou $t_1 \sqsubseteq s_2$.

Agora, $b \circ a \circ b$ é da forma $s_1 \circ s_2$ com

$$s_1 = b \text{ e}$$

$$s_2 = a \circ b.$$

75.6. RELAÇÕES E FUNÇÕES

Ele também é da forma $r_1 \circ r_2$ com

$$r_1 = b \circ a \text{ e } r_2 = b.$$

Agora, $a \circ b$ é um subtermo de $b \circ a \circ b$? A resposta é sim, se seguirmos a primeira leitura, e não, se seguirmos a segunda.

A propriedade de que o modo como um termo bonito é construído a partir de outros termos bonitos é único é chamada de *leitura única*. Como definições indutivas de relações para tais objetos definidos indutivamente são importantes, temos que provar que ela vale.

Proposição 75.9. *Suponha que t é um termo bonito. Então ou t é uma letra isolada, ou existem termos bonitos univocamente determinados s_1, s_2 tais que $t = [s_1 \circ s_2]$.*

Prova. Se t é uma letra isolada, a condição é satisfeita. Então suponha que t não é uma letra isolada. Podemos perceber, pela definição indutiva, que então t deve ser da forma $[s_1 \circ s_2]$ para alguns termos bonitos s_1 e s_2 . Resta mostrar que estes são univocamente determinados, isto é, se $t = [r_1 \circ r_2]$, então $s_1 = r_1$ e $s_2 = r_2$.

Então suponha $t = [s_1 \circ s_2]$ e também $t = [r_1 \circ r_2]$ para termos bonitos s_1, s_2, r_1, r_2 . Temos que mostrar que $s_1 = r_1$ e $s_2 = r_2$. Primeiro, s_1 e r_1 devem ser idênticos, pois, caso contrário, um é um segmento inicial próprio do outro. Mas, por **Proposição 75.6**, isso é impossível se s_1 e r_1 são ambos termos bonitos. Mas se $s_1 = r_1$, então claramente também $s_2 = r_2$. $\square$

Também podemos definir funções indutivamente: por exemplo, podemos definir a função f que mapeia qualquer termo bonito na profundidade máxima de aninhamento de $[\dots]$ nele como segue:

Definição 75.10. A *profundidade* de um termo bonito, $f(t)$, é definida indutivamente como segue:

$$f(t) = \begin{cases} 0 & \text{se } t \text{ é uma letra} \\ \max(f(s_1), f(s_2)) + 1 & \text{se } t = [s_1 \circ s_2]. \end{cases}$$

Por exemplo,

$$\begin{aligned} f([a \circ b]) &= \max(f(a), f(b)) + 1 = \\ &= \max(0, 0) + 1 = 1, \text{ e} \\ f([([a \circ b] \circ c)]) &= \max(f([a \circ b]), f(c)) + 1 = \\ &= \max(1, 0) + 1 = 2. \end{aligned}$$

Aqui, é claro, supomos que s_1 e s_2 são termos bonitos, e fazemos uso do fato de que todo termo bonito ou é uma letra ou é da forma $[s_1 \circ s_2]$. É novamente importante que ele possa ser dessa forma de apenas um modo. Para ver por

quê, considere novamente os termos sem colchetes que definimos antes. A “definição” correspondente seria:

$$g(t) = \begin{cases} 0 & \text{se } t \text{ é uma letra} \\ \max(g(s_1), g(s_2)) + 1 & \text{se } t = s_1 \circ s_2. \end{cases}$$

Agora considere o termo sem colchetes $a \circ b \circ c \circ d$. Ele pode ser lido de mais de um modo, por exemplo, como $s_1 \circ s_2$ com

$$s_1 = a \text{ e } s_2 = b \circ c \circ d,$$

ou como $r_1 \circ r_2$ com

$$r_1 = a \circ b \text{ e } r_2 = c \circ d.$$

Calcular g de acordo com o primeiro modo de lê-lo daria

$$\begin{aligned} g(s_1 \circ s_2) &= \max(g(a), g(b \circ c \circ d)) + 1 = \\ &= \max(0, 2) + 1 = 3 \end{aligned}$$

enquanto, de acordo com a outra leitura, obtemos

$$\begin{aligned} g(r_1 \circ r_2) &= \max(g(a \circ b), g(c \circ d)) + 1 = \\ &= \max(1, 1) + 1 = 2 \end{aligned}$$

Mas uma função deve sempre produzir um valor único; então nossa “definição” de g não define função alguma.

Problema 75.3. Dê uma definição indutiva da função l , em que $l(t)$ é o número de símbolos no termo bonito t .

Problema 75.4. Prove, por indução estrutural sobre termos bonitos t , que $f(t) < l(t)$ (em que $l(t)$ é o número de símbolos em t e $f(t)$ é a profundidade de t tal como definida em [Definição 75.10](#)).

Parte XVII

History

Capítulo 76

Biografias

<content/history/biographies/georg-cantor.tex>

76.1 Georg Cantor

his:bio:can:
sec

Uma biografia inicial de Georg Cantor (GAY-org KAHN-tor) afirmava que ele tinha nascido e sido encontrado num navio que seguia para São Petersburgo, na Rússia, e que seus pais eram desconhecidos. Isto, porém, não é verdade; embora ele tivesse nascido em São Petersburgo em 1845.

Cantor se doutorou em matemática na Universidade de Berlim em 1867. É conhecido por seu trabalho em teoria dos conjuntos e é considerado o fundador da teoria dos conjuntos como disciplina de pesquisa distinta. Foi o primeiro a provar que existem conjuntos infinitos de tamanhos diferentes. Suas teorias, e sobretudo a teoria dos infinitos, provocaram muito debate entre os matemáticos da época, e seu trabalho foi controverso.

As convicções religiosas de Cantor e seu trabalho matemático estavam inextricavelmente ligados; ele chegou a afir-



Figura 76.1: Georg Cantor

mar que a teoria dos números transfinitos lhe tinha sido comunicada diretamente por Deus. Mais tarde, Cantor sofreu de doença mental. A partir de 1894, e com maior frequência nos últimos anos, Cantor foi internado. As fortes críticas a seu trabalho, incluindo um rompimento com o matemático Leopold Kronecker, levaram à depressão e à perda de interesse pela matemática. Durante episódios depressivos, Cantor se dedicava à filosofia e à literatura, e chegou a publicar uma teoria de que Francis Bacon era o autor das peças de Shakespeare.

Cantor morreu em 6 de janeiro de 1918, num sanatório em Halle.

Leitura complementar Para biografias completas de Cantor, ver [Dauben \(1990\)](#) e [Grattan-Guinness \(1971\)](#). As visões radicais de Cantor são também descritas no programa da BBC Radio 4 *A Brief History of Mathematics* ([du Sautoy, 2014](#)). Se quiser ouvir as teorias de Cantor em forma de rap, ver [Rose \(2012\)](#).

[content/history/biographies/alonzo-church.tex](#)

76.2 Alonzo Church

Alonzo Church nasceu em Washington, DC, em 14 de junho de 1903. Na primeira infância, um acidente com uma arma de ar comprimido deixou Church cego de um olho. Ele concluiu o ensino preparatório em Connecticut em 1920 e iniciou sua formação universitária em Princeton naquele mesmo ano. Concluiu seus estudos de doutorado em 1927. Após alguns anos no exterior, Church retornou a Princeton. Church era conhecido por ser extremamente educado e cuidadoso. Sua escrita no quadro-negro era impecável, e ele preservava artigos importantes cobrindo-os cuidadosamente com cimento Duco (uma cola transparente). Fora de suas atividades acadêmicas, gostava de ler revistas de ficção científica e não tinha receio de escrever aos editores caso identificasse quaisquer imprecisões nos textos.



his:bio:chu:
sec

Figura 76.2: Alonzo Church

As realizações acadêmicas de Church foram notáveis. Junto com seus alunos Stephen Kleene e Barkley Rosser, ele desenvolveu uma teoria da calculabilidade efetiva, o cálculo lambda, independentemente do desenvolvimento da máquina de Turing por Alan Turing. As duas definições de computabilidade são equivalentes e dão origem ao que hoje é conhecido como a *Tese de Church-Turing*: a de que uma função dos números naturais é efetivamente computável

76.3. GERHARD GENTZEN

se e somente se for computável por meio de uma máquina de Turing (ou do cálculo lambda). Ele também provou o que hoje é conhecido como o *Teorema de Church*: o problema da decisão para a validade de fórmulas de primeira ordem é insolúvel.

Church continuou seu trabalho até a velhice. Em 1967, deixou Princeton para ir à UCLA, onde foi professor até se aposentar, em 1990. Church faleceu em 1º de agosto de 1995, aos 92 anos.

Leitura complementar Para uma breve biografia de Church, veja [Enderton \(2019\)](#). Os escritos originais de Church sobre o cálculo lambda e o Entscheidungsproblem (Tese de Church) são [Church \(1936a,b\)](#). [Aspray \(1984\)](#) registra uma entrevista com Church sobre a comunidade matemática de Princeton na década de 1930. Church escreveu uma série de resenhas de livros para o *Journal of Symbolic Logic* de 1936 até 1979. Todas estão arquivadas no site de John MacFarlane ([MacFarlane, 2015](#)).

[content/history/biographies/gerhard-gentzen.tex](#)

76.3 Gerhard Gentzen

his:bio:gen:
sec

Gerhard Gentzen é conhecido principalmente como o criador da teoria estrutural da prova e, especificamente, pela criação dos sistemas de *derivação* de dedução natural e de cálculo de seqüentes. Ele nasceu em 24 de novembro de 1909 em Greifswald, na Alemanha. Gerhard foi educado em casa por três anos antes de frequentar o ensino preparatório, onde estava atrás da maioria de seus colegas em termos de instrução. Apesar disso, era um aluno brilhante e demonstrava forte aptidão para a matemática. Seus interesses eram variados e ele, por exemplo, também escrevia poemas para a mãe e peças para o teatro da escola.

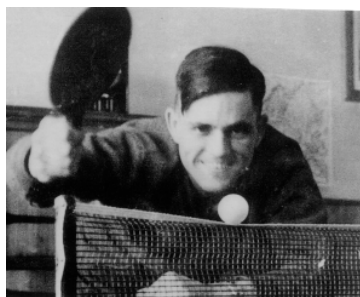


Figura 76.3: Gerhard Gentzen

Gentzen começou seus estudos universitários na Universidade de Greifswald, mas passou por Göttingen, Munique e Berlim. Recebeu seu doutorado em 1933 pela Universidade de Göttingen, sob a orientação de Hermann Weyl. (Paul Bernays supervisionou a maior parte de seu trabalho, mas foi demitido da universidade pelos nazistas.) Em 1934, Gentzen começou a trabalhar como assistente de David Hilbert. Nesse mesmo ano, desenvolveu os sistemas de *derivação* de cálculo de seqüentes e de dedução natural, em seus artigos *Untersuchungen über das logische Schließen I–II* [*Investigações sobre a Dedução Lógica I–II*]. Ele provou a consistência dos axiomas de Peano em 1936.

A relação de Gentzen com os nazistas é complicada. Ao mesmo tempo em que seu mentor Bernays foi forçado a deixar a Alemanha, Gentzen ingressou no braço universitário da SA, a organização paramilitar nazista. Como muitos alemães, foi membro do partido nazista. Durante a guerra, serviu como oficial de telecomunicações na unidade de inteligência aérea. No entanto, em 1942 foi dispensado do serviço devido a um colapso nervoso. Não está claro se as lealdades de Gentzen estavam com o partido nazista ou se ele ingressou no partido a fim de garantir sucesso acadêmico.

Em 1943, foi oferecida a Gentzen uma posição acadêmica no Instituto Matemático da Universidade Alemã de Praga, que ele aceitou. No entanto, em 1945 os cidadãos de Praga se revoltaram contra a ocupação alemã. As forças soviéticas chegaram à cidade e prenderam todos os professores da universidade. Por causa de sua filiação a organizações nazistas, Gentzen foi levado a um campo de trabalhos forçados. Ele morreu de desnutrição em sua cela em 4 de agosto de 1945, aos 35 anos.

Leitura complementar Para uma biografia completa de Gentzen, veja [Menzler-Trott \(2007\)](#). Uma leitura interessante sobre matemáticos sob o regime nazista, que traz uma breve nota sobre a vida de Gentzen, é dada por [Segal \(2014\)](#). Os artigos de Gentzen sobre dedução lógica estão disponíveis no alemão original ([Gentzen, 1935a,b](#)). Traduções para o inglês dos artigos de Gentzen foram reunidas em um único volume por [Szabo \(1969\)](#), que também inclui um esboço biográfico.

[content/history/biographies/kurt-goedel.tex](#)

76.4 Kurt Gödel

Kurt Gödel (GER-dle) nasceu em 28 de abril de 1906 em Brünn, no império austro-húngaro (hoje Brno, na República Tcheca). Devido a seu espírito inquisitivo e brilhante, o jovem Kurtele era muitas vezes chamado “Der kleine Herr Warum” (o Pequeno Senhor Porquê) pela família. Destacou-se nos estudos desde a escola primária, tendo obtido menos do que a nota mais alta apenas em matemática. Gödel faltava frequentemente à escola por causa da frágil saúde e estava isento de educação física. Foi diagnosticado com febre reumática durante a infância. Ao longo da vida, acreditou que isso tinha afetado permanentemente seu coração, apesar dos pareceres médicos em contrário.

his:bio:god:
sec

Gödel começou a estudar na Universidade de Viena em 1924 e concluiu o doutorado em 1929. Inicialmente pretendia estudar física, mas seus interesses se deslocaram rapidamente para a matemática e sobretudo para a lógica, em parte devido à influência do filósofo Rudolf Carnap. Sua dissertação, orientada por Hans Hahn, provava o teorema de completude da lógica de predicados de primeira ordem com identidade ([Gödel, 1929](#)). Apenas um ano depois, obteve seus resultados mais famosos—o primeiro e o segundo teorema da incompletude (publicados em [Gödel 1931](#)). Durante o período em Viena, Gödel participou

76.4. KURT GÖDEL

intensamente no Círculo de Viena, um grupo de filósofos de orientação científica que incluía Carnap, cujo trabalho foi especialmente influenciado pelos resultados de Gödel.

Em 1938, Gödel se casou com Adele Nimbursky. Os pais não ficaram contentes: não só ela era seis anos mais velha e já divorciada, como trabalhava como bailarina num clube noturno. As pressões sociais não afetaram Gödel, contudo, e permaneceram felizmente casados até sua morte.

Depois de a Alemanha Nazi ter anexo a Áustria em 1938, Gödel e Adele emigraram para os Estados Unidos, onde ele assumiu um lugar no Institute for Advanced Study em Princeton, Nova Jersey. Apesar da introversão e do caráter excêntrico, o período de Gödel em Princeton foi colaborativo e frutuoso. Publicou ensaios em teoria dos conjuntos, filosofia e física. Notavelmente, estabeleceu uma amizade particularmente forte com o colega no IAS, Albert Einstein.

Nos últimos anos, a saúde mental de Gödel se deteriorou. A internação da esposa em 1977 fez com que ela não pudesse mais cozinhar as refeições dele. Tendo sofrido de problemas de saúde mental ao longo da vida, sucumbiu à paranoia. Com medo mortal de ser envenenado, Gödel se recusou a comer. Morreu de inanição em 14 de janeiro de 1978, em Princeton.



Figura 76.4: Kurt Gödel

Leitura complementar Para uma biografia completa da vida de Gödel, ver [John Dawson \(1997\)](#). Para outros textos biográficos, bem como ensaios sobre as contribuições de Gödel para a lógica e a filosofia, ver [Wang \(1990\)](#), [Baaz et al. \(2011\)](#), [Takeuti et al. \(2003\)](#), e [Sigmund et al. \(2007\)](#).

A tese de doutorado de Gödel está disponível no alemão original ([Gödel, 1929](#)). O texto original dos teoremas da incompletude é ([Gödel, 1931](#)). Todas as publicações e textos não publicados de Gödel, bem como uma seleção de correspondência, estão disponíveis em inglês em seus *Collected Papers* [Feferman et al. \(1986, 1990\)](#).

Para um tratamento detalhado dos teoremas da incompletude de Gödel, ver [Smith \(2013\)](#). Para uma discussão informal e filosófica dos teoremas de Gödel, ver o podcast de Mark Linsenmayer ([Linsenmayer, 2014](#)).

76.5 Emmy Noether

Emmy Noether (NER-ter) nasceu em Erlangen, na Alemanha, em 23 de março de 1882, em uma família erudita de classe média alta. Aclamada como a “mãe da álgebra moderna”, Noether fez contribuições inovadoras tanto para a matemática quanto para a física, apesar das significativas barreiras à educação das mulheres. Na Alemanha daquela época, esperava-se que as meninas fossem educadas nas artes e elas não tinham permissão para frequentar escolas preparatórias para a faculdade. No entanto, após assistir como ouvinte às aulas nas Universidades de Göttingen e Erlangen (onde seu pai era professor de matemática), Noether finalmente conseguiu se matricular como aluna em Erlangen em 1904, quando a política da instituição foi atualizada para permitir estudantes do sexo feminino. Ela recebeu seu doutorado em matemática em 1907.



his:bio:noe:
sec

Figura 76.5: Emmy Noether

Apesar de suas qualificações, Noether enfrentou muita resistência durante sua carreira. De 1908 a 1915, ela lecionou em Erlangen sem remuneração. Durante esse período, chamou a atenção de David Hilbert, um dos mais eminentes matemáticos do mundo na época, que a convidou para Göttingen. No entanto, as mulheres eram proibidas de obter cátedras, e ela só pôde lecionar sob o nome de Hilbert, novamente sem remuneração. Foi nesse período que ela provou o que hoje é conhecido como o teorema de Noether, que ainda é usado na física teórica atualmente. Noether finalmente recebeu o direito de lecionar em 1919. A resposta de Hilbert à resistência contínua de seus colegas de universidade teria sido: “Senhores, o conselho da faculdade não é uma casa de banhos.”

No final da década de 1920, ela concentrou-se no trabalho em álgebra abstrata, e suas contribuições revolucionaram o campo. Em suas provas, ela frequentemente fazia uso da chamada condição de cadeia ascendente, que afirma que não há cadeia infinita estritamente crescente de certos conjuntos. Por exemplo, certas estruturas algébricas hoje conhecidas como anéis noetherianos têm a propriedade de que não há sequências infinitas de ideais $I_1 \subsetneq I_2 \subsetneq \dots$. A condição pode ser generalizada para qualquer ordem parcial (na álgebra, ela diz respeito ao caso especial de ideais ordenados pela relação de subconjunto), e também podemos considerar a condição dual de cadeia descendente, na qual toda sequência estritamente decrescente em uma ordem parcial eventualmente termina. Se uma ordem parcial satisfaz a condição de cadeia descendente, é possível usar indução ao longo dessa ordem de maneira semelhante àquela em

76.6. RÓZSA PÉTER

que podemos usar indução ao longo da ordem $<$ sobre $\mathbb{N}$. Tais ordens são chamadas *bem fundadas* ou *noetherianas*, e o princípio de prova correspondente, *indução noetheriana*.

Noether era judia e, quando os nazistas chegaram ao poder em 1933, ela foi demitida de seu cargo. Felizmente, Noether conseguiu emigrar para os Estados Unidos para ocupar uma posição temporária em Bryn Mawr, na Pensilvânia. Durante seu tempo lá, ela também lecionou em Princeton, embora considerasse a universidade pouco acolhedora para as mulheres (Dick, 1981, 81). Em 1935, Noether passou por uma operação para remover um tumor uterino. Ela morreu de uma infecção decorrente da cirurgia e foi enterrada em Bryn Mawr.

Leitura complementar Para uma biografia de Noether, veja Dick (1981). O Perimeter Institute for Theoretical Physics disponibiliza on-line suas palestras sobre a vida e a influência de Noether (Institute, 2015). Se você estiver cansado de ler, *Stuff You Missed in History Class* tem um podcast sobre a vida e a influência de Noether (Frey and Wilson, 2015). As obras reunidas de Noether estão disponíveis no alemão original (Jacobson, 1983).

-NoValue-

[content/history/biographies/rozs-peter.tex](#)

76.6 Rózsa Péter

his:bio:pet:
sec

Rózsa Péter was born Rózsa Politzer, in Budapest, Hungary, on February 17, 1905. She is best known for her work on recursive functions, which was essential for the creation of the field of recursion theory.

Péter was raised during harsh political times—WWI raged when she was a teenager—but was able to attend the affluent Maria Terezia Girls' School in Budapest, from where she graduated in 1922. She then studied at Pázmány Péter University (later renamed Loránd Eötvös University) in Budapest. She began studying chemistry at the insistence of her father, but later switched to mathematics, and graduated in 1927. Although she had the credentials to teach high school mathematics, the economic situation at the time was dire as the Great Depression affected the world economy. During this time, Péter took odd jobs as a tutor and private teacher of mathematics. She eventually returned to university to take up graduate



Figura 76.6: Rózsa Péter

studies in mathematics. She had originally planned to work in number theory, but after finding out that her results had already been proven, she almost gave up on mathematics altogether. She was encouraged to work on Gödel's incompleteness theorems, and unknowingly proved several of his results in different ways. This restored her confidence, and Péter went on to write her first papers on recursion theory, inspired by David Hilbert's foundational program. She received her PhD in 1935, and in 1937 she became an editor for the *Journal of Symbolic Logic*.

Péter's early papers are widely credited as founding contributions to the field of recursive function theory. In Péter (1935a), she investigated the relationship between different kinds of recursion. In Péter (1935b), she showed that a certain recursively defined function is not primitive recursive. This simplified an earlier result due to Wilhelm Ackermann. Péter's simplified function is what's now often called the Ackermann function—and sometimes, more properly, the Ackermann–Péter function. She wrote the first book on recursive function theory (Péter, 1951).

Despite the importance and influence of her work, Péter did not obtain a full-time teaching position until 1945. During the Nazi occupation of Hungary during World War II, Péter was not allowed to teach due to anti-Semitic laws. In 1944 the government created a Jewish ghetto in Budapest; the ghetto was cut off from the rest of the city and attended by armed guards. Péter was forced to live in the ghetto until 1945 when it was liberated. She then went on to teach at the Budapest Teachers Training College, and from 1955 onward at Eötvös Loránd University. She was the first female Hungarian mathematician to become an Academic Doctor of Mathematics, and the first woman to be elected to the Hungarian Academy of Sciences.

Péter was known as a passionate teacher of mathematics, who preferred to explore the nature and beauty of mathematical problems with her students rather than to merely lecture. As a result, she was affectionately called “Aunt Rosa” by her students. Péter died in 1977 at the age of 71.

Leitura complementar For more biographical reading, see (O'Connor and Robertson, 2014) and (Andrásfai, 1986). Tamassy (1994) conducted a brief interview with Péter. For a fun read about mathematics, see Péter's book *Playing With Infinity* (Péter, 2010).

-NoValue-

[content/history/biographies/julia-robinson.tex](#)

76.7 Julia Robinson

Julia Bowman Robinson was an American mathematician. She is known mainly for her work on decision problems, and most famously for her contributions to the solution of Hilbert's tenth problem. Robinson was born in St. Louis, Missouri, on December 8, 1919. Robinson recalls being intrigued by numbers

his:bio:rob:
sec

76.7. JULIA ROBINSON

already as a child (Reid, 1986, 4). At age nine she contracted scarlet fever and suffered from several recurrent bouts of rheumatic fever. This forced her to spend much of her time in bed, putting her behind in her education. Although she was able to catch up with the help of private tutors, the physical effects of her illness had a lasting impact on her life.

Despite her childhood struggles, Robinson graduated high school with several awards in mathematics and the sciences. She started her university career at San Diego State College, and transferred to the University of California, Berkeley, as a senior. There she was influenced by the mathematician Raphael Robinson. They became good friends, and married in 1941. As a spouse of a faculty member, Robinson was barred from teaching in the mathematics department at Berkeley. Although she continued to audit mathematics classes, she hoped to leave university and start a family. Not long after her wedding, however, Robinson contracted pneumonia. She was told that there was substantial scar tissue build up on her heart due to the rheumatic fever she suffered as a child. Due to the severity of the scar tissue, the doctor predicted that she would not live past forty and she was advised not to have children (Reid, 1986, 13).

Robinson was depressed for a long time, but eventually decided to continue studying mathematics. She returned to Berkeley and completed her PhD in 1948 under the supervision of Alfred Tarski. The first-order theory of the real numbers had been shown to be decidable by Tarski, and from Gödel's work it followed that the first-order theory of the natural numbers is undecidable. It was a major open problem whether the first-order theory of the rationals is decidable or not. In her thesis (1949), Robinson proved that it was not.

Interested in decision problems, Robinson next attempted to find a solution to Hilbert's tenth problem. This problem was one of a famous list of 23 mathematical problems posed by David Hilbert in 1900. The tenth problem asks whether there is an algorithm that will answer, in a finite amount of time, whether or not a polynomial equation with integer coefficients, such as $3x^2 - 2y + 3 = 0$, has a solution in the integers. Such questions are known as *Diophantine problems*. After some initial successes, Robinson joined forces with Martin Davis and Hilary Putnam, who were also working on the problem. They succeeded in showing that exponential Diophantine problems (where the unknowns may also appear as exponents) are undecidable, and showed that a certain conjecture (later called "J.R.") implies that Hilbert's tenth problem is undecidable (Davis et al., 1961). Robinson continued to work on the problem

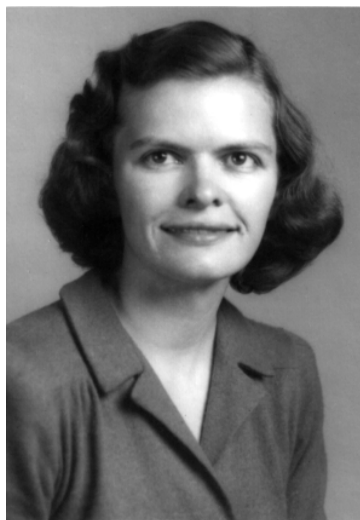


Figura 76.7: Julia Robinson

throughout the 1960s. In 1970, the young Russian mathematician Yuri Matijasevich finally proved the J.R. hypothesis. The combined result is now called the Matijasevich–Robinson–Davis–Putnam theorem, or MRDP theorem for short. Matijasevich and Robinson became friends and collaborated on several papers. In a letter to Matijasevich, Robinson once wrote that “actually I am very pleased that working together (thousands of miles apart) we are obviously making more progress than either one of us could alone” (Matijasevich, 1992, 45).

Robinson was the first female president of the American Mathematical Society, and the first woman to be elected to the National Academy of Science. She died on July 30, 1985 at the age of 65 after being diagnosed with leukemia.

Leitura complementar Robinson’s mathematical papers are available in her *Collected Works* (Robinson, 1996), which also includes a reprint of her National Academy of Sciences biographical memoir (Feferman, 1994). Robinson’s older sister Constance Reid published an “Autobiography of Julia,” based on interviews (Reid, 1986), as well as a full memoir (Reid, 1996). A short documentary about Robinson and Hilbert’s tenth problem was directed by George Csicsery (Csicsery, 2016). For a brief memoir about Yuri Matijasevich’s collaborations with Robinson, and her influence on his work, see (Matijasevich, 1992).

content/history/biographies/bertrand-russell.tex

76.8 Bertrand Russell

Bertrand Russell é celebrado como um dos fundadores da filosofia analítica moderna. Nascido em 18 de maio de 1872, Russell não era conhecido apenas pelo trabalho em filosofia e lógica, mas escreveu muitos livros de divulgação em diversas áreas. Foi também um ardente ativista político ao longo da vida.

his:bio:rus:
sec

Russell nasceu em Trellech, Monmouthshire, no País de Gales. Os pais pertenciam à nobreza britânica. Eram livre-pensadores e chegaram a travar amizade com os radicais de Boston da época. Infelizmente, os pais de Russell morreram quando ele era ainda criança, e Russell foi viver com os avós. Ali recebeu uma educação religiosa (algo que os pais tinham querido evitar a todo custo). A avó era muito rigorosa em todas as questões de moralidade. Na adolescência foi sobretudo educado em casa por professores particulares.

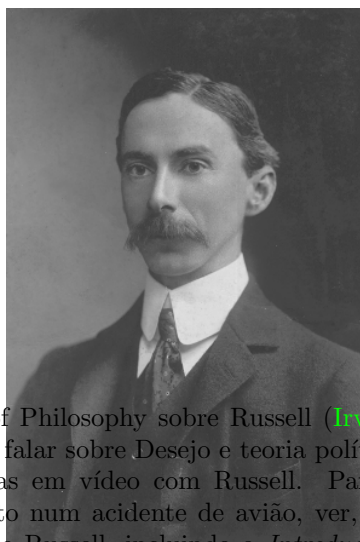
A influência de Russell na filosofia analítica, e sobretudo na lógica, é enorme. Estudou matemática e filosofia no Trinity College, em Cambridge, onde foi influenciado pelo matemático e filósofo Alfred North Whitehead. Em 1910, Russell e Whitehead publicaram o primeiro volume de *Principia Mathematica*, defendendo a ideia de que a matemática é redutível à lógica. Seguiu-se a publicação de centenas de livros, ensaios e panfletos políticos. Em 1950, ganhou o Prêmio Nobel de Literatura.

Russell se encontrava profundamente envolvido na política e no ativismo social. Durante a Primeira Guerra Mundial foi preso e condenado a seis meses

76.9. ALFRED TARSKI

de prisão por atividades pacifistas e protesto. Na prisão pôde escrever e ler, e afirma ter achado a experiência “bastante agradável.” Permaneceu pacifista ao longo da vida, e voltou a ser detido por participar num comício pelo desarmamento nuclear em 1961. Sobreviveu também a um acidente de avião em 1948, no qual os únicos sobreviventes eram os que estavam sentados na área de fumantes. Por isso, Russell declarou que devia a vida ao tabagismo. Russell se casou quatro vezes, mas tinha fama de manter relações extraconjugais. Morreu em 2 de fevereiro de 1970, aos 97 anos, em Penrhyndeudraeth, no País de Gales.

Leitura complementar Russell escreveu uma autobiografia em três partes, que abrange sua vida de 1872 a 1967 (Russell, 1967, 1968, 1969). O Bertrand Russell Research Centre na McMaster University alberga os arquivos de Bertrand Russell. Ver o site em Duncan (2015), para informação sobre os volumes de suas obras reunidas (incluindo índices pesquisáveis) e projetos de arquivo. O artigo de Russell *On Denoting* (Russell, 1905) é um clássico da filosofia analítica do século XX.



A entrada da Stanford Encyclopedia of Philosophy sobre Russell (Irvine, 2015) inclui excertos de áudio de Russell a falar sobre Desejo e teoria política. Estão disponíveis online muitas entrevistas em vídeo com Russell. Para o ouvir falar de tabagismo e do envolvimento num acidente de avião, ver, por exemplo, Russell (n.d.). Algumas obras de Russell, incluindo a *Introduction to Mathematical Philosophy*, estão disponíveis como audiolivros gratuitos em LibriVox (n.d.).

`content/history/biographies/alfred-tarski.tex`

76.9 Alfred Tarski

his:bio:tar:
sec

Alfred Tarski nasceu em 14 de janeiro de 1901 em Varsóvia, na Polónia (então parte do Império Russo). Descrito como “napoleônico”, Tarski era ruidoso, falador e intenso. Sua energia se refletia muitas vezes nas aulas—chegou a incendiar um cesto de lixo ao jogar fora um cigarro durante uma aula, e foi proibido de lecionar novamente naquele edifício.

Tarski tinha sede de conhecimento desde cedo. Mais tarde diria aos estudantes que estudara lógica porque era a única disciplina em que tivera um B, mas os registros do colégio mostram que tinha notas máximas em tudo—até em lógica. Estudou na Universidade de Varsóvia de 1918 a 1924. Tarski pretendia inicialmente estudar biologia, mas se interessou por matemática, filosofia e lógica, pois a universidade era o centro da Escola de Lógica e Filosofia de Varsóvia. Tarski se doutorou em 1924 sob orientação de Stanisław Leśniewski.



Figura 76.9: Alfred Tarski

Antes de emigrar para os Estados Unidos em 1939, Tarski completou parte de seu trabalho mais importante enquanto professor de ensino secundário em Varsóvia. O trabalho sobre consequência lógica e verdade lógica foi escrito nesse período. Em 1939, Tarski se encontrava nos Estados Unidos numa turnê de palestras. Durante a visita, a Alemanha invadiu a Polônia, e, por causa da ascendência judaica, Tarski não pôde retornar. A esposa e os filhos permaneceram na Polônia até o fim da guerra, mas depois também puderam emigrar para os Estados Unidos. Tarski lecionou em Harvard, no College of the City of New York, no Institute for Advanced Study em Princeton, e finalmente na Universidade da Califórnia, em Berkeley. Ali fundou o programa multidisciplinar em Lógica e Metodologia da Ciência. Tarski morreu em 26 de outubro de 1983, aos 82 anos.

Leitura complementar Para mais sobre a vida de Tarski, ver a biografia *Alfred Tarski: Life and Logic* (Feferman and Feferman, 2004). As obras fundadoras de Tarski sobre consequência lógica e verdade estão disponíveis em inglês em (Corcoran, 1983). Todas as obras originais de Tarski foram reunidas numa série de quatro volumes, (Tarski, 1981).

[content/history/biographies/alan-turing.tex](#)

76.10 Alan Turing

Alan Turing nasceu em Maida Vale, Londres, em 23 de junho de 1912. É considerado o pai da ciência da computação teórica. O interesse de Turing pelas ciências físicas e pela matemática começou em tenra idade. No entanto, quando menino, seus interesses não eram bem contemplados nas escolas que frequentava, onde a ênfase recaía sobre a literatura e os clássicos. Consequentemente,

[his:bio:turing](#)
[sec](#)

76.10. ALAN TURING

ele teve um desempenho fraco na escola e foi repreendido por muitos de seus professores.

Turing estudou no King's College, em Cambridge, na graduação, onde estudou matemática. Em 1936, Turing desenvolveu (o que hoje se chama) a máquina de Turing como uma tentativa de definir precisamente a noção de função computável e de provar a indecidibilidade do problema da decisão. Ele foi superado no resultado por Alonzo Church, que o provou por meio de seu próprio cálculo lambda. Ainda assim, o artigo de Turing foi publicado com referência ao resultado de Church. Church convidou Turing para Princeton, onde ele passou de 1936 a 1938 e obteve um doutorado sob a orientação de Church.



Figura 76.10: Alan Turing

Apesar de seu interesse pela lógica, os interesses anteriores de Turing pelas ciências físicas permaneceram predominantes. Suas habilidades práticas foram postas em prática durante seu serviço no departamento britânico de criptanálise em Bletchley Park durante a Segunda Guerra Mundial. Turing foi uma figura central na quebra da cifra usada nas comunicações navais alemãs—o código Enigma. A expertise de Turing em estatística e criptografia, juntamente com a introdução de maquinário eletrônico, deu à equipe a capacidade de quebrar o código por meio da criação de uma máquina de descriptografia chamada “bombe”. Suas ideias também ajudaram na criação do primeiro computador eletrônico programável do mundo, o Colossus, também usado em Bletchley Park para quebrar a cifra alemã Lorenz.

Turing era homossexual. Mesmo assim, em 1942 ele pediu Joan Clarke em casamento, uma de suas colegas de equipe em Bletchley Park, mas depois rompeu o noivado e confessou a ela que era homossexual. Teve vários amantes ao longo da vida, embora os atos homossexuais fossem, então, crimes no Reino Unido. Em 1952, a casa de Turing foi assaltada por um amigo de seu amante da época e, ao prestar queixa à polícia, Turing admitiu ter um relacionamento homossexual, sob a impressão de que o governo estava a caminho de legalizar os atos homossexuais. Isso não era verdade, e ele foi acusado de grave indecência. Em vez de ir para a prisão, Turing optou por um tratamento hormonal que reduzia a libido. Turing foi encontrado morto em 8 de junho de 1954, de uma overdose de cianeto—muito provavelmente suicídio. Ele recebeu um perdão real da Rainha Elizabeth II em 2013.

Leitura complementar Para uma biografia abrangente de Alan Turing, veja [Hodges \(2014\)](#). A vida e a obra de Turing inspiraram uma peça, *Breaking*

the Code, que foi produzida em 1996 para a TV, com Derek Jacobi no papel de Turing. *The Imitation Game*, um filme indicado ao Oscar, com Benedict Cumberbatch e Kiera Knightley, também é baseado livremente na vida de Alan Turing e em seu período em Bletchley Park (Tyldum, 2014).

Radiolab (2012) tem vários podcasts sobre a vida e a obra de Turing. O documentário *The Strange Life and Death of Dr. Turing*, da BBC Horizon, está disponível para assistir on-line (Sykes, 1992). (Theelen, 2012) é um vídeo curto de uma Máquina de Turing de LEGO em funcionamento—feito para homenagear o centenário de Turing em 2012.

O artigo original de Turing sobre máquinas de Turing e o problema da decisão é Turing (1937).

[content/history/biographies/ernst-zermelo.tex](#)

76.11 Ernst Zermelo

Ernst Zermelo nasceu em 27 de julho de 1871 em Berlim, na Alemanha. Tinha cinco irmãs, embora a família sofresse de frágil saúde e apenas três sobrevivessem à idade adulta. Os pais também faleceram quando ele era jovem, deixando-o órfão, com os irmãos, aos dezessete anos. Zermelo tinha um profundo interesse pelas artes, e sobretudo pela poesia. Era conhecido por ser perspicaz, espirituoso e crítico. Entre suas conquistas matemáticas mais celebradas estão a introdução do axioma da escolha (em 1904) e sua axiomatização da teoria dos conjuntos (em 1908).

Os interesses de Zermelo na universidade eram variados. Frequentou cursos de física, matemática e filosofia. Sob orientação de Hermann Schwarz, Zermelo concluiu a dissertação *Investigations in the Calculus of Variations* em 1894 na Universidade de Berlim. Em 1897, decidiu prosseguir estudos na Universidade de Göttingen, onde foi fortemente influenciado pelo trabalho fundacional de David Hilbert. Em 1899 tornou-se elegível para uma cátedra, mas só obteve uma onze anos depois—possivelmente devido a seu comportamento estranho e à “pressa nervosa.”

Zermelo obteve finalmente uma cátedra remunerada na Universidade de Zurique em 1910, mas foi forçado a se aposentar em 1916 por causa da tuberculose. Depois da recuperação, foi nomeado professor honorário na Universidade de Freiburg em 1921. Nesse período dedicou-se à matemática fundacional.



his:bio:zer:
sec

Figura 76.11: Ernst Zermelo

Irritou-se com os trabalhos de Thoralf Skolem e Kurt Gödel, e criticou publicamente suas abordagens em seus artigos. Foi demitido do cargo em Freiburg em 1935, devido a sua impopularidade e à oposição à ascensão de Hitler ao poder na Alemanha.

Os últimos anos da vida de Zermelo foram marcados pelo isolamento. Depois da demissão em 1935, abandonou a matemática. Mudou-se para o campo, onde viveu modestamente. Casou-se em 1944 e ficou completamente dependente da esposa à medida que ia ficando cego. Zermelo perdeu totalmente a visão por volta de 1951. Faleceu em Günterstal, na Alemanha, em 21 de maio de 1953.

Leitura complementar Para uma biografia completa de Zermelo, ver [Ebbinghaus \(2015\)](#). Os artigos fundadores de 1904 e 1908 de Zermelo podem ser lidos no alemão original ([Zermelo, 1904, 1908b](#)). As obras reunidas de Zermelo, incluindo os escritos sobre física, estão disponíveis em tradução inglesa em [Ebbinghaus et al. \(2010\)](#); [Ebbinghaus and Kanamori \(2013\)](#).

Capítulo 77

História e Mitologia da Teoria dos Conjuntos

This chapter includes the historical prelude from Tim Button’s Open Set Theory text.

`content/history/set-theory/infininitesimals.tex`

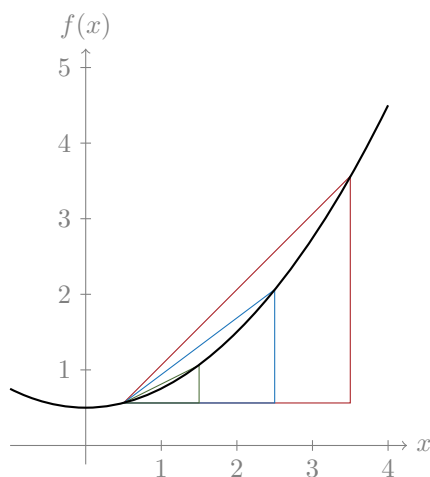
77.1 Infinitesimais e Diferenciação

`his:set:infininitesimals:
sec`

Newton e Leibniz descobriram o cálculo (independentemente) no fim do século XVII. Uma aplicação particularmente importante do cálculo era a *diferenciação*. Grosso modo, a diferenciação pretende dar uma noção da “taxa de variação”, ou gradiente, de uma função num ponto.

CAPÍTULO 77. HISTÓRIA E MITOLOGIA DA TEORIA DOS CONJUNTOS

Eis uma forma vívida de ilustrar a ideia. Considere a função $f(x) = x^2/4 + 1/2$, representada a preto abaixo:



Suponhamos que queremos encontrar o gradiente da função em $c = 1/2$. Começamos por desenhar um triângulo cuja hipotenusa aproxima o gradiente nesse ponto, talvez o triângulo vermelho acima. Quando β é o comprimento da base do nosso triângulo, a sua altura é $f(1/2 + \beta) - f(1/2)$, de modo que o gradiente da hipotenusa é:

$$\frac{f(1/2 + \beta) - f(1/2)}{\beta}.$$

Assim, o gradiente do nosso triângulo **vermelho**, com base de comprimento 3, é exatamente 1. A hipotenusa de um triângulo menor, o triângulo **azul** com base de comprimento 2, dá uma melhor aproximação; o seu gradiente é $3/4$. Um triângulo ainda menor, o **verde** com base de comprimento 1, dá uma aproximação ainda melhor, com gradiente $1/2$.

Triângulos cada vez menores dão-nos aproximações cada vez melhores. Por isso poderíamos dizer algo deste gênero: a hipotenusa de um triângulo com comprimento de base *infinitesimal* nos dá o próprio gradiente em $c = 1/2$. Deste modo, obteríamos uma fórmula para a (primeira) derivada da função f no ponto c :

$$f'(c) = \frac{f(c + \beta) - f(c)}{\beta} \text{ onde } \beta \text{ é infinitesimal.}$$

E, grosso modo, foi isto que Newton e Leibniz disseram.

Contudo, uma vez que o disseram, devemos perguntar-lhes: o que é um *infinitesimal*? Surge um dilema sério. Se $\beta = 0$, então f' fica mal definida, pois envolve dividir por 0. Mas se $\beta > 0$, obtemos apenas uma *aproximação* ao gradiente, e não o próprio gradiente.

Esta não é uma preocupação anacrônica. Eis Berkeley, a criticar os seguidores de Newton:

77.2. DEFINIÇÃO RIGOROSA DE LIMITES

Admito que os signos possam ser feitos para denotar qualquer coisa ou nada: e consequentemente que, na notação original $c + \beta$, β poderia ter significado um incremento ou nada. Mas então, seja qual for o significado atribuído a β , é preciso argumentar de modo consistente com essa significação, e não proceder com duplo sentido: o que seria um sofismo manifesto. (Berkeley 1734, §XIII, variáveis alteradas para coincidir com o texto precedente)

Para defender o cálculo infinitesimal face a Berkeley, poder-se-ia responder que a linguagem dos “infinitesimais” é meramente figurada. Poder-se-ia dizer que, desde que tomemos um triângulo *bem pequeno*, obteremos uma aproximação *suficientemente boa* à tangente. Berkeley tinha também uma resposta para isto: embora isso pudesse bastar para a engenharia, minava o *estatuto* da matemática, pois

dizem-nos que *in rebus mathematicis errores quàm minimi non sunt contemnendi*. [No caso da matemática, os menores erros não devem ser desprezados.] (Berkeley, 1734, §IX)

A passagem em itálico é uma citação quase literal do próprio Newton na *Quadrature of Curves* (1704).

As objeções filosóficas de Berkeley são profundamente incisivas. Não obstante, o cálculo foi uma empresa enormemente bem-sucedida, e os matemáticos continuaram a usá-lo sem incorrer em erro.

content/history/set-theory/limits.tex

77.2 Definição Rigorosa de Limites

his:set:limits:sec Hoje em dia, a solução padrão ao problema anterior é livrar-se dos infinitesimais. Eis como.

Já vimos que, à medida que β fica menor, obtemos melhores aproximações do gradiente. De fato, à medida que β se aproxima arbitrariamente de 0, o valor de $f'(c)$ “tende sem limite” para o gradiente que queremos. Assim, em vez de considerar o que acontece *em* $\beta = 0$, precisamos apenas considerar a *tendência* de $f'(c)$ quando β se aproxima de 0.

Posto deste modo, o desafio geral é dar sentido a afirmações desta forma:

À medida que x se aproxima de c , $g(x)$ tende sem limite para ℓ .

que podemos escrever de forma mais compacta assim:

$$\lim_{x \rightarrow c} g(x) = \ell.$$

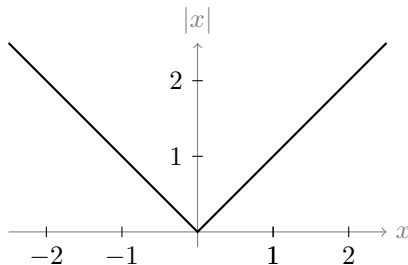
No século XIX, sobre trabalho anterior de Cauchy, Weierstrass deu uma definição perfeitamente rigorosa desta expressão. A ideia é, de fato, que podemos

CAPÍTULO 77. HISTÓRIA E MITOLOGIA DA TEORIA DOS CONJUNTOS

tornar $g(x)$ tão próxima quanto queiramos de ℓ , tornando x adequadamente próximo de c . Mais precisamente, estipulamos que $\lim_{x \rightarrow c} g(x) = \ell$ significará:

$$(\forall \varepsilon > 0)(\exists \delta > 0) \forall x (|x - c| < \delta \rightarrow |g(x) - \ell| < \varepsilon).$$

As barras verticais indicam aqui valor absoluto. Isto é, $|x| = x$ quando $x \geq 0$, e $|x| = -x$ quando $x < 0$; pode representar-se *essa* função como se segue:



A definição diz portanto, grosso modo, o seguinte: pode tornar o seu “erro” inferior a ε (isto é, $|g(x) - \ell| < \varepsilon$) escolhendo argumentos que distem no máximo δ de c (isto é, $|x - c| < \delta$).

Tendo definido a noção de limite, podemos usá-la para evitar por completo os infinitesimais, estipulando que o gradiente de f em c é dado por:

$$f'(c) = \lim_{x \rightarrow 0} \left(\frac{f(c+x) - f(c)}{x} \right) \text{ quando um limite existe.}$$

É importante, no entanto, compreender por que a nossa definição precisa da ressalva “quando um limite existe”. Para um exemplo simples, considere $f(x) = |x|$, cujo gráfico acabamos de ver. Evidentemente, $f'(0)$ está mal definida: se nos aproximamos de 0 “pela direita”, o gradiente é sempre 1; se nos aproximamos “pela esquerda”, o gradiente é sempre -1 ; logo o limite não está definido. Como tal, podemos acrescentar que uma função f é *diferenciável* em x se e só se tal limite existe.

Já vimos como tratar a diferenciação usando a noção de *limite*. Podemos usar a mesma noção para definir a ideia de função *contínua*. (Bolzano tinha, com efeito, percebido isto já por 1817.) O tratamento Cauchy–Weierstrass da continuidade é o seguinte. Grosso modo: uma função f é contínua (num ponto) desde que, ao exigir um certo grau de precisão relativamente à saída da função, se possa garantir essa precisão insistindo num certo grau de precisão relativamente à entrada da função. Mais precisamente: f é contínua em c desde que, à medida que x tende para zero, a diferença entre $f(c+x)$ e $f(c)$ tenda para 0. De outro modo: f é *contínua* em c se e só se $f(c) = \lim_{x \rightarrow c} f(x)$.

Ir mais longe levar-nos-ia apenas para a análise real, quando o nosso tema é a teoria dos conjuntos. Por isso, agora devemos fazer uma pausa e enunciar a moral. Durante o século XIX, os matemáticos aprenderam a passar sem infinitesimais, invocando uma noção rigorosamente definida de *limite*.

content/history/set-theory/pathologies.tex

77.3 Patologias

his:set:pathology: Contudo, a definição de um *limite* acabou por permitir construções bastante
sec “patológicas”.

Por volta da década de 1830, Bolzano descobriu uma função que era *contínua em todo o lado*, mas *não diferenciável em parte nenhuma*. (Infelizmente, Bolzano nunca publicou isto; a ideia foi pela primeira vez encontrada pelos matemáticos em 1872, graças à descoberta independente de Weierstrass da mesma ideia.)¹ Isto era, no mínimo, bastante surpreendente. É fácil encontrar funções, como $|x|$, que são contínuas em todo o lado mas não diferenciáveis num dado ponto. Mas uma função contínua em todo o lado e *não diferenciável em parte nenhuma* é uma criatura muito diferente. Considere, por um momento, como tentaria desenhar tal função. Para garantir que é contínua, deve consegui-la desenhar sem nunca levantar a caneta da página; mas para garantir que não é diferenciável em nenhum ponto, teria de mudar abruptamente a direção da caneta, constantemente.

Seguiram-se mais “patologias”. A 5 de janeiro de 1874, Cantor escreveu uma carta a Dedekind, colocando o problema:

Poderá uma superfície (digamos um quadrado incluindo o seu contorno) estar correlacionada biunivocamente com uma linha (digamos uma reta incluindo as extremidades) de modo que a cada ponto da superfície corresponda um ponto da linha e, reciprocamente, a cada ponto da linha corresponda um ponto da superfície?

Ainda me parece, de momento, que a resposta a esta questão é muito difícil—embora aqui também se esteja tão impelido a dizer *não* que se queira considerar quase supérflua a prova. [Citado em Gouvêa 2011]

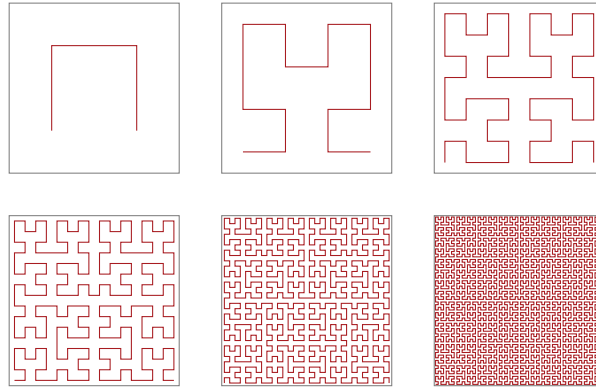
Mas, em 1877, Cantor provou que estivera errado. De fato, uma linha e um quadrado têm exatamente o mesmo número de pontos. Escreveu a 29 de junho de 1877 a Dedekind “*je le vois, mais je ne le crois pas*”; isto é, “eu vejo, mas não acredito”. Na “história recebida” da matemática, isto é frequentemente tomado como indicador de quão *literalmente incríveis* eram estes novos resultados para os matemáticos da época. (A correspondência é apresentada em Gouvêa (2011), e voltaremos a ela em Seção 77.4. A prova de Cantor está esboçada em Seção 77.5.)

Inspirado pelo resultado de Cantor, Peano começou a considerar se seria possível mapear suavemente uma linha sobre um plano. Isto seria uma *curva que preenche o espaço*. Em 1890, Peano construiu precisamente tal curva. Isto é verdadeiramente contraintuitivo: Euclides tinha definido uma linha como “comprimento sem largura” (Livro I, Definição 2), mas Peano mostrara que, enrolando adequadamente uma linha, o seu comprimento pode transformar-se em largura. Em 1891, Hilbert descreveu uma curva preenchedora de espaço

¹A história está documentada em notas de rodapé extremamente minuciosas do artigo da Wikipédia sobre a [função de Weierstrass](#).

CAPÍTULO 77. HISTÓRIA E MITOLOGIA DA TEORIA DOS CONJUNTOS

ligeiramente mais intuitiva, juntamente com algumas figuras que a ilustram. A curva é construída em sequência, e eis as primeiras seis fases da construção:



No limite—uma noção que, entretanto, recebera definição rigorosa—todo o quadrado fica inteiramente preenchido na cor **vermelho**. E, de passagem, a curva de Hilbert é contínua em todo o lado mas não diferenciável em parte nenhuma; intuitivamente porque, no limite infinito, a função muda abruptamente de direção a cada instante. (Esboçaremos a construção de Hilbert com mais pormenor na **Seção 77.6**.)

Para o bem ou para o mal, estas construções geométricas “patológicas” foram tratadas como motivo para duvidar de apelos à intuição geométrica. Tornaram-se algo próximo de *propaganda* por uma nova forma de fazer matemática, que culminaria na teoria dos conjuntos. Na construção posterior do mito em torno do assunto, repetia-se, frequentemente, que estes resultados eram ao mesmo tempo perfeitamente rigorosos e perfeitamente chocantes. Serviam portanto um duplo propósito: como aviso contra confiar na intuição geométrica, e como uma demonstração da fertilidade de novas formas de pensar.

`content/history/set-theory/mythology.tex`

77.4 Mais Mito do que História?

Com mais de um século de perspectiva, ao olhar para estes acontecimentos, devemos ter cuidado para não tomar estes veredictos por adquiridos. Os resultados eram certamente novos, entusiasmantes e surpreendentes. Mas quão verdadeiramente chocantes eram? E demonstraram realmente que não devemos confiar na intuição geométrica?

`his:set:mythology:sec`

Quanto ao choque, **Gouvêa (2011)** observa que a célebre nota de Cantor a Dedekind, “*je le vois, mais je ne le crois pas*” é tirada um tanto fora de contexto. Eis mais desse contexto (citado de **Gouvêa**):

982 Peço desculpa pelo meu zelo pelo assunto se lhe faço tantas exigências à sua bondade e paciência; as comunicações que ultimamente lhe enviei são mesmo para mim tão inesperadas, tão novas, que não posso ter paz de espírito até obter de si, amigo honrado, uma decisão sobre a sua correcção. Enquanto não concordar comigo, só posso dizer: *je le vois, mais je ne le crois pas*.

Release : 51c2271 (2026-07-12)

77.5. CANTOR NA RETA E NO PLANO

Cantor sabia que o seu resultado era “tão inesperado, tão novo”. Mas é duvidoso que alguma vez achasse o seu resultado *inacreditável*. Como Gouvêa observa, estava simplesmente a pedir a Dedekind que verificasse a prova que oferecera.

Quanto à intuição geométrica: Peano publicou a sua curva preenchedora de espaço sem incluir quaisquer diagramas. Mas quando Hilbert publicou a sua curva, explicou o seu propósito: daria aos leitores um modo claro de compreender o resultado de Peano, se “se servirem da seguinte intuição geométrica”; após o que incluiu uma série de *diagramas* tal como os fornecidos em [Seção 77.3](#).

De modo mais geral: embora os diagramas tenham caído um tanto em desuso nas provas publicadas, não há como escapar ao fato de que os matemáticos *com frequência* usam diagramas ao provar resultados. (Grosso modo: bons matemáticos sabem quando podem confiar na intuição geométrica.)

Em suma: não acredite na *exaltação*; ou, pelo menos, não a aceite por adquirido. Para mais sobre isto, pode ler [Giaquinto \(2007\)](#).

[content/history/set-theory/cantor-plane.tex](#)

77.5 Cantor na Reta e no Plano

[his:set:cantorplane:](#)
[sec](#)

Algumas circunstâncias em torno da prova de Schröder-Bernstein relacionam-se com a história que discutimos em [Seção 77.3](#). Recordemos que, em 1877, Cantor provou que há tantos pontos num quadrado quanto num dos seus lados. Aqui apresentaremos a sua prova (primeira tentativa).

Seja L o intervalo unitário, isto é, o conjunto de pontos $[0, 1]$. Seja S o quadrado unitário, isto é, o conjunto de pontos $L \times L$. Nestes termos, Cantor provou que $L \approx S$. Escreveu uma nota a Dedekind, que continha essencialmente o seguinte argumento.

[his:set:cantorplane:](#)
[thm:cantorplane](#)

Teorema 77.1. $L \approx S$

Prova: primeira parte.. Fixemos $a, b \in L$. Escrevamos a e b em notação binária, de modo que temos sequências infinitas de 0s e 1s, $a_1, a_2, \dots$, e $b_1, b_2, \dots$, tais que:

$$a = 0.a_1a_2a_3a_4\dots$$

$$b = 0.b_1b_2b_3b_4\dots$$

Consideremos a função $f: S \rightarrow L$ dada por

$$f(a, b) = 0.a_1b_1a_2b_2a_3b_3a_4b_4\dots$$

A função f é [uma injeção](#), pois se $f(a, b) = f(c, d)$, então $a_n = c_n$ e $b_n = d_n$ para todo $n \in \mathbb{N}$, logo $a = c$ e $b = d$. $\square$

Infelizmente, como Dedekind apontou a Cantor, isto não responde à questão original. Considere-se $0.\dot{1}0 = 0.10101010\dots$. Para que $f(a, b) = 0.\dot{1}0$, se

preciso:

$$\begin{aligned}a &= 0.\dot{1}\dot{1} = 0.111111\dots \\b &= 0\end{aligned}$$

Mas $a = 0.\dot{1}\dot{1} = 1$. Assim, quando dizemos “escrever a e b em notação binária”, temos de escolher *qual* notação usar; e, como f deve ser uma *função*, só podemos usar *uma* das duas notações possíveis. Mas se, por exemplo, usarmos a notação simples e escrevermos a como “1.000...”, então não existe par $\langle a, b \rangle$ tal que $f(a, b) = 0.\dot{1}\dot{0}$.

Em resumo: Dedekind observou que, dada a possibilidade de certas expansões binárias periódicas, a função f de Cantor é **uma injeção** mas *não* é **uma sobrejeção**. Cantor tinha mostrado apenas que $S \preceq L$ e *não* que $S \approx L$.

Cantor respondeu a Dedekind quase de imediato, sugerindo essencialmente que a prova poderia completar-se assim:

Prova: completada.. Mostramos que $S \preceq L$. Mas é óbvio que existe **uma injeção** de L para S : basta “deitar” a reta sobre um lado do quadrado. Logo $L \preceq S$ e $S \preceq L$. Pelo teorema de Schröder–Bernstein (**Teorema 4.25**), $L \approx S$. $\square$

Claro que Cantor não podia justificar a última linha nestes termos, pois o teorema de Schröder–Bernstein ainda não estava provado. De fato, embora Cantor viesse a formulá-lo como conjectura geral, só viria a ser provado de modo satisfatório em 1897. (E assim, mais tarde em 1877, Cantor deu outra prova de **Teorema 77.1**, que não passava por Schröder–Bernstein.)

[content/history/set-theory/hilbert-curve.tex](#)

77.6 Apêndice: Curvas Preenchedoras de Espaço de Hilbert

No capítulo **Seção 77.3**, mencionamos que a prova de Cantor de que uma reta e um quadrado têm exatamente o mesmo número de pontos (**Teorema 77.1**) levou Peano a perguntar se poderia existir uma *curva* que preenchesse o espaço. Obteve resposta afirmativa em 1890. Nesta seção explicamos (de modo informal) como construir a curva preenchedora de espaço de Hilbert (com um pequeno ajuste).²

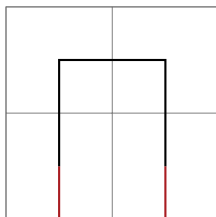
[his:set:hilbertcurve:sec](#)

Temos de definir uma função, h , como limite de uma sequência de funções $h_1, h_2, h_3, \dots$. Primeiro descrevemos a construção. Depois mostramos que preenche o espaço. Depois mostramos que é uma curva.

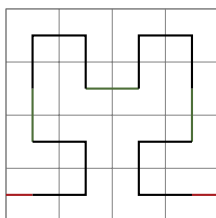
Tomaremos a imagem de h como sendo o quadrado unitário, S . Eis a nossa primeira aproximação a h , isto é, h_1 :

²Para uma explicação mais rigorosa, ver [Rose \(2010\)](#). O ajuste consiste em incluir as partes vermelhas das curvas abaixo. Isto torna ligeiramente mais fácil verificar que a curva é contínua.

77.6. APÊNDICE: CURVAS PREENCHEDORAS DE ESPAÇO DE HILBERT

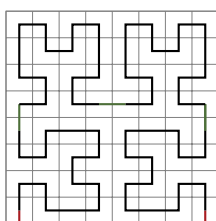


Para orientação, impusemos uma grelha 2×2 ao quadrado. Podemos pensar na curva como começando no quadrante inferior esquerdo, indo para o superior esquerdo, depois para o superior direito e finalmente para o inferior direito. Eis a segunda etapa da construção, isto é, h_2 :



As cores diferentes ajudam a explicar como h_2 foi construída. Colocamos primeiro cópias reduzidas do troço não-vermelho de h_1 no inferior esquerdo, superior esquerdo, superior direito e inferior direito do quadrado (traçadas a preto). Depois ligamos estas quatro figuras (com linhas verde). Por fim, ligamos a figura ao contorno do quadrado (com linhas vermelho).

Passamos a h_3 . Tal como h_2 foi feita a partir de quatro cópias ligadas e reduzidas do troço não vermelho de h_1 , h_3 compõe-se de quatro cópias reduzidas do troço não vermelho de h_2 (a preto), que depois se unem (com linhas verde) e se ligam ao contorno do quadrado (com linhas vermelho).



Agora vê-se o padrão geral para definir h_{n+1} a partir de h_n . Por fim definimos a curva h em si considerando o limite ponto a ponto destas funções sucessivas $h_1, h_2, \dots$. Isto é, para cada $x \in S$:

$$h(x) = \lim_{n \rightarrow \infty} h_n(x)$$

Mostramos agora que esta curva preenche o espaço. Ao traçar h_n , impomos uma grelha $2^n \times 2^n$ sobre S . Pelo teorema de Pitágoras, a diagonal de cada

célula da grelha tem comprimento:

$$\sqrt{(1/2^n)^2 + (1/2^n)^2} = 2^{(\frac{1}{2}-n)}$$

e é evidente que h_n passa por todas as células. Logo cada ponto de S está a uma distância no máximo $2^{(\frac{1}{2}-n)}$ de algum ponto de h_n . Mas h é definida como o limite das funções $h_1, h_2, h_3, \dots$. Assim, a distância máxima de qualquer ponto a h é dada por:

$$\lim_{n \rightarrow \infty} 2^{(\frac{1}{2}-n)} = 0.$$

Ou seja: todo o ponto de S está à distância 0 de h . Em outras palavras, todo o ponto de S situa-se *sobre* a curva. Logo h preenche o espaço!

Resta mostrar que h é, de fato, uma *curva*. Para isso é preciso definir o conceito. A definição moderna baseia-se na de Jordan, de 1887 (isto é, poucos anos antes da primeira curva preenchedora de espaço):

Definição 77.2. Uma curva é uma aplicação contínua de L para $\mathbb{R}^2$.

Isto é bastante intuitivo: uma curva é, intuitivamente, uma aplicação “suave” que leva uma reta canônica para o plano $\mathbb{R}^2$. A nossa função h é de fato uma aplicação de L para $\mathbb{R}^2$. Logo só precisamos de mostrar que h é contínua. Definimos continuidade em [Seção 77.2](#) com a notação ε/δ . Em linguagem corrente, queremos estabelecer o seguinte: *Se especificar um ponto p em S , juntamente com qualquer nível de precisão ε , podemos encontrar um intervalo aberto de L tal que, dado qualquer x nesse intervalo, $h(x)$ fica a uma distância inferior a ε de p .*

Assim: suponhamos que p e ε estão fixos. Isto corresponde, em efeito, a traçar um círculo com centro p e raio ε sobre S . (O círculo pode sair para fora de S , mas não importa.) Recordemos que, ao descrever h_n , traçamos uma grelha $2^n \times 2^n$ sobre S . É óbvio que, por pequeno que seja ε , existe algum n tal que alguma célula individual da grelha $2^n \times 2^n$ sobre S fica inteiramente dentro do círculo com centro p e raio ε .

Tomemos esse n , e seja I a maior parte aberta de L que h_n mapeia inteiramente para a célula relevante. (É claro que I existe, pois já notamos que h_n passa por todas as células da grelha $2^n \times 2^n$.) Basta agora mostrar que, sempre que $x \in I$, o ponto $h(x)$ pertence à mesma célula. E, para *isso*, basta mostrar que $h_m(x)$ pertence à mesma célula, para qualquer $m > n$. Mas isto é óbvio. Se considerarmos o que acontece com h_m para $m > n$, vemos que exatamente a “mesma parte” do intervalo unitário é mapeada para a mesma célula; apenas a descrevemos de modo cada vez mais esticado e sinuoso.

Parte XVIII

Reference

This part collects various lists of alphabets, notations, definitions, rules, etc., for inclusion as appendices in a textbook.

Capítulo 78

O Alfabeto Grego

Alfa	α	A	Niu	ν	N
Beta	β	B	Xi	ξ	Ξ
Gama	γ	Γ	Ômicron	o	O
Delta	δ	Δ	Pi	π	Π
Épsilon	ε	E	Rô	ρ	P
Zeta	ζ	Z	Sigma	σ	Σ
Eta	η	H	Tau	τ	T
Teta	θ	Θ	Ípsilon	v	Υ
Iota	ι	I	Fi	φ	Φ
Capa	κ	K	Qui	χ	X
Lambda	λ	Λ	Psi	ψ	Ψ
Mi	μ	M	Ômega	ω	Ω

Capítulo 79

The Fraktur Alphabet

A	𝔸	a	𝔞	N	𝔑	n	𝔫
B	𝔖	b	𝔞	O	𝔔	o	𝔔
C	𝔢	c	𝔢	P	𝔓	p	𝔓
D	𝔡	d	𝔡	Q	𝔚	q	𝔚
E	𝔢	e	𝔢	R	𝔖	r	𝔢
F	𝔣	f	𝔣	S	𝔢	s	𝔢
G	𝔢	g	𝔢	T	𝔗	t	𝔗
H	𝔥	h	𝔥	U	𝔘	u	𝔘
I	𝔦	i	𝔦	V	𝔙	v	𝔙
J	𝔧	j	𝔧	W	𝔚	w	𝔚
K	𝔦	k	𝔦	X	𝔗	x	𝔗
L	𝔡	l	𝔡	Y	𝔙	y	𝔙
M	𝔖	m	𝔖	Z	𝔗	z	𝔗

Créditos das Imagens

Georg Cantor, [p. 963](#): Retrato de Georg Cantor por Otto Zeth, cortesia do [Arquivo Universitário, Martin-Luther-Universität Halle-Wittenberg](#). UAHW Rep. 40-VI, Nr. 3 Bild 102.

Kurt Gödel, [p. 967](#): Retrato de Kurt Gödel, c. 1925, fotografia desconhecido. Do [Shelby White and Leon Levy Archives Center, Institute for Advanced Study](#), Princeton, NJ, EUA, em depósito na Biblioteca da Universidade de Princeton, [Divisão de Manuscritos, Departamento de Livros Raros e Coleções Especiais](#), Kurt Gödel Papers, (C0282), Box 14b, #110000. O *Open Logic Project* obteve permissão do Arquivo do Instituto para usar esta imagem em

materiais derivados do OLP não comerciais. É necessária autorização do Arquivo para qualquer outro uso.

Bertrand Russell, p. 973: Retrato de Bertrand Russell, c. 1907, cortesia da Divisão William Ready de Arquivos e Coleções de Investigação, Biblioteca da McMaster University. [Arquivos Bertrand Russell](#), Box 2, f. 4.

Alfred Tarski, p. 974: Fotografia de passaporte de Alfred Tarski, 1939. Recortada e restaurada a partir de digitalização do passaporte de Tarski por Joel Fuller. Original cortesia da [Biblioteca Bancroft, University of California, Berkeley](#). Alfred Tarski Papers, Banc MSS 84/49. O *Open Logic Project* obteve permissão para usar esta imagem em materiais derivados do OLP não comerciais. É necessária autorização da Biblioteca Bancroft para qualquer outro uso.

Ernst Zermelo, p. 976: Retrato de Ernst Zermelo, c. 1922, cortesia da [Abteilung für Handschriften und Seltene Drucke, Niedersächsische Staats- und Universitätsbibliothek Göttingen](#), Cod. Ms. D. Hilbert 754, Bl. 6 Nr. 25.

Referências Bibliográficas

Andrásfai, Béla. 1986. Rózsa (Rosa) Péter. *Periodica Polytechnica Electrical Engineering* 30(2-3): 139–145. URL <http://www.pp.bme.hu/ee/article/view/4651>.

Aspray, William. 1984. The Princeton mathematics community in the 1930s: Alonzo Church. URL http://www.princeton.edu/mudd/finding_aids/mathoral/pmc05.htm. Interview.

Baaz, Matthias, Christos H. Papadimitriou, Hilary W. Putnam, Dana S. Scott, and Charles L. Harper Jr. 2011. *Kurt Gödel and the Foundations of Mathematics: Horizons of Truth*. Cambridge: Cambridge University Press.

Banach, Stefan and Alfred Tarski. 1924. Sur la décomposition des ensembles de points en parties respectivement congruentes. *Fundamenta Mathematicae* 6: 244–77.

Benacerraf, Paul. 1965. What numbers could not be. *The Philosophical Review* 74(1): 47–73.

Berkeley, George. 1734. *The Analyst; or, a Discourse Addressed to an Infidel Mathematician*.

Boolos, George. 1971. The iterative conception of set. *The Journal of Philosophy* 68(8): 215–31.

REFERÊNCIAS BIBLIOGRÁFICAS

- Boolos, George. 1989. Iteration again. *Philosophical Topics* 17(2): 5–21.
- Boolos, George. 2000. Must we believe in set theory? In *Between Logic and Intuition: Essays in Honor of Charles Parsons*, eds. Gila Sher and Richard Tieszen, 257–68. Cambridge: Cambridge University Press.
- Burali-Forti, Cesare. 1897. Una questione sui numeri transfiniti. *Rendiconti del Circolo Matematico di Palermo* 11: 154–64.
- Button, Tim. 2021. Level theory, part 1: Axiomatizing the bare idea of a cumulative hierarchy of sets. *The Bulletin of Symbolic Logic* 27(4): 436–460.
- Cantor, Georg. 1878. Ein Beitrag zur Mannigfaltigkeitslehre. *Journal für die reine und angewandte Mathematik* 84: 242–58.
- Cantor, Georg. 1883. *Grundlagen einer allgemeinen Mannigfaltigkeitslehre. Ein mathematisch-philosophischer Versuch in der Lehre des Unendlichen*. Leipzig: Teubner.
- Cantor, Georg. 1892. Über eine elementare Frage der Mannigfaltigkeitslehre. *Jahresbericht der deutschen Mathematiker-Vereinigung* 1: 75–8.
- Cheng, Eugenia. 2004. How to write proofs: A quick guide. URL <https://eugeniacheng.com/wp-content/uploads/2017/02/cheng-proofguide.pdf>.
- Church, Alonzo. 1936a. A note on the Entscheidungsproblem. *The Journal of Symbolic Logic* 1: 40–41.
- Church, Alonzo. 1936b. An unsolvable problem of elementary number theory. *American Journal of Mathematics* 58: 345–363.
- Cohen, Paul J. 1963. The independence of the continuum hypothesis. *Proceedings of the National Academy of Sciences of the United States of America* 24: 556–557.
- Cohen, Paul J. 1966. *Set Theory and the Continuum Hypothesis*. Reading, MA: Benjamin.
- Conway, John. 2006. The power of mathematics. In *Power*, eds. Alan Blackwell and David MacKay, Darwin College Lectures. Cambridge: Cambridge University Press. URL <http://www.cs.toronto.edu/~mackay/conway.pdf>.
- Corcoran, John. 1983. *Logic, Semantics, Metamathematics*. Indianapolis: Hackett, 2nd ed.
- Csicsery, George. 2016. Zala films: Julia Robinson and Hilbert’s tenth problem. URL <http://www.zalafilms.com/films/juliarobinson.html>.
- Dauben, Joseph. 1990. *Georg Cantor: His Mathematics and Philosophy of the Infinite*. Princeton: Princeton University Press.

REFERÊNCIAS BIBLIOGRÁFICAS

- Davis, Martin, Hilary Putnam, and Julia Robinson. 1961. The decision problem for exponential Diophantine equations. *Annals of Mathematics* 74(3): 425–436. URL <http://www.jstor.org/stable/1970289>.
- Dedekind, Richard. 1888. *Was sind und was sollen die Zahlen?* Braunschweig: Vieweg.
- Dick, Auguste. 1981. *Emmy Noether 1882–1935*. Boston: Birkhäuser.
- du Sautoy, Marcus. 2014. A brief history of mathematics: Georg Cantor. URL <http://www.bbc.co.uk/programmes/b00ss1j0>. Audio Recording.
- Duncan, Arlene. 2015. The Bertrand Russell Research Centre. URL <http://russell.mcmaster.ca/>.
- Ebbinghaus, Heinz-Dieter. 2015. *Ernst Zermelo: An Approach to his Life and Work*. Berlin: Springer-Verlag.
- Ebbinghaus, Heinz-Dieter, Craig G. Fraser, and Akihiro Kanamori. 2010. *Ernst Zermelo. Collected Works*, vol. 1. Berlin: Springer-Verlag.
- Ebbinghaus, Heinz-Dieter and Akihiro Kanamori. 2013. *Ernst Zermelo: Collected Works*, vol. 2. Berlin: Springer-Verlag.
- Enderton, Herbert B. 2019. Alonzo Church: Life and Work. In *The Collected Works of Alonzo Church*, eds. Tyler Burge and Herbert B. Enderton. Cambridge, MA: MIT Press.
- Feferman, Anita and Solomon Feferman. 2004. *Alfred Tarski: Life and Logic*. Cambridge: Cambridge University Press.
- Feferman, Solomon. 1994. Julia Bowman Robinson 1919–1985. *Biographical Memoirs of the National Academy of Sciences* 63: 1–28. URL <http://www.nasonline.org/publications/biographical-memoirs/memoir-pdfs/robinson-julia.pdf>.
- Feferman, Solomon, John W. Dawson Jr., Stephen C. Kleene, Gregory H. Moore, Robert M. Solovay, and Jean van Heijenoort. 1986. *Kurt Gödel: Collected Works. Vol. 1: Publications 1929–1936*. Oxford: Oxford University Press.
- Feferman, Solomon, John W. Dawson Jr., Stephen C. Kleene, Gregory H. Moore, Robert M. Solovay, and Jean van Heijenoort. 1990. *Kurt Gödel: Collected Works. Vol. 2: Publications 1938–1974*. Oxford: Oxford University Press.
- Feferman, Solomon and Azriel Levy. 1963. Independence results in set theory by Cohen’s method II. *Notices of the American Mathematical Society* 10: 593.

REFERÊNCIAS BIBLIOGRÁFICAS

- Fraenkel, Abraham. 1922. Über den Begriff ‘definit’ und die Unabhängigkeit des Auswahlaxioms. *Sitzungsberichte der Preussischen Akademie der Wissenschaften, Physikalisch-mathematische Klasse* 253–257.
- Frege, Gottlob. 1884. *Die Grundlagen der Arithmetik: Eine logisch mathematische Untersuchung über den Begriff der Zahl*. Breslau: Wilhelm Koebner. Translation in Frege (1953).
- Frege, Gottlob. 1953. *Foundations of Arithmetic*, ed. J. L. Austin. Oxford: Basil Blackwell & Mott, 2nd ed.
- Frey, Holly and Tracy V. Wilson. 2015. Stuff you missed in history class: Emmy Noether, mathematics trailblazer. URL <https://www.iheart.com/podcast/stuff-you-missed-in-history-cl-21124503/episode/emmy-noether-mathematics-trailblazer-30207491/>. Podcast audio.
- Gentzen, Gerhard. 1935a. Untersuchungen über das logische Schließen I. *Mathematische Zeitschrift* 39: 176–210. English translation in Szabo (1969), pp. 68–131.
- Gentzen, Gerhard. 1935b. Untersuchungen über das logische Schließen II. *Mathematische Zeitschrift* 39: 176–210, 405–431. English translation in Szabo (1969), pp. 68–131.
- Giaquinto, Marcus. 2007. *Visual Thinking in Mathematics*. Oxford: Oxford University Press.
- Gödel, Kurt. 1929. Über die Vollständigkeit des Logikkalküls [On the completeness of the calculus of logic]. Dissertation, Universität Wien. Reprinted and translated in Feferman et al. (1986), pp. 60–101.
- Gödel, Kurt. 1931. über formal unentscheidbare Sätze der *Principia Mathematica* und verwandter Systeme I [On formally undecidable propositions of *Principia Mathematica* and related systems I]. *Monatshefte für Mathematik und Physik* 38: 173–198. Reprinted and translated in Feferman et al. (1986), pp. 144–195.
- Gödel, Kurt. 1938. The consistency of the axiom of choice and the generalized continuum hypothesis. *Proceedings of the National Academy of Sciences of the United States of America* 50: 1143–8.
- Gouvêa, Fernando Q. 2011. Was Cantor surprised? *American Mathematical Monthly* 118(3): 198–209.
- Grattan-Guinness, Ivor. 1971. Towards a biography of Georg Cantor. *Annals of Science* 27(4): 345–391.
- Hammack, Richard. 2013. *Book of Proof*. Richmond, VA: Virginia Commonwealth University. URL <http://www.people.vcu.edu/~rhammack/BookOfProof/BookOfProof.pdf>.

REFERÊNCIAS BIBLIOGRÁFICAS

- Hartogs, Friedrich. 1915. Über das Problem der Wohlordnung. *Mathematische Annalen* 76: 438–43.
- Hausdorff, Felix. 1914. Bemerkung über den Inhalt von Punktmengen. *Mathematische Annalen* 75: 428–34.
- Heck, Richard Kimberly. 2012. *Reading Frege's Grundgesetze*. Oxford: Oxford University Press.
- Heijenoort, Jean van. 1967. *From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931*. Cambridge, MA: Harvard University Press.
- Hilbert, David. 1891. Über die stetige Abbildung einer Linie auf ein Flächenstück. *Mathematische Annalen* 38(3): 459–460.
- Hilbert, David. 2013. *David Hilbert's Lectures on the Foundations of Arithmetic and Logic 1917–1933*, eds. William Bragg Ewald and Wilfried Sieg. Heidelberg: Springer.
- Hodges, Andrew. 2014. *Alan Turing: The Enigma*. London: Vintage.
- Hume, David. 1740. *A Treatise of Human Nature*. London.
- Hutchings, Michael. 2003. Introduction to mathematical arguments. URL <https://math.berkeley.edu/~hutching/teach/proofs.pdf>.
- Incurvati, Luca. 2020. *Conceptions of Set and the Foundations of Mathematics*. Cambridge: Cambridge University Press.
- Institute, Perimeter. 2015. Emmy Noether: Her life, work, and influence. URL <https://www.youtube.com/watch?v=tNNyAyMRsgE>. Video Lecture.
- Irvine, Andrew David. 2015. Sound clips of Bertrand Russell speaking. URL <http://plato.stanford.edu/entries/russell/russell-soundclips.html>.
- Jacobson, Nathan. 1983. *Emmy Noether: Gesammelte Abhandlungen—Collected Papers*. Berlin: Springer-Verlag.
- John Dawson, Jr. 1997. *Logical Dilemmas: The Life and Work of Kurt Gödel*. Boca Raton: CRC Press.
- Katz, Karin Usadi and Mikhail G. Katz. 2012. Stevin numbers and reality. *Foundations of Science* 17(2): 109–23.
- Kunen, Kenneth. 1980. *Set Theory: An Introduction to Independence Proofs*. New York: North Holland.
- Lévy, Azriel. 1960. Axiom schemata of strong infinity in axiomatic set theory. *Pacific Journal of Mathematics* 10(1): 223–38.

REFERÊNCIAS BIBLIOGRÁFICAS

- LibriVox. n.d. Bertrand Russell. URL https://librivox.org/author/1508?primary_key=1508&search_category=author&search_page=1&search_form=get_results. Collection of public domain audiobooks.
- Linnebo, Øystein. 2010. Predicative and impredicative definitions. *Internet Encyclopedia of Philosophy* URL <http://www.iep.utm.edu/predicat/>.
- Linsenmayer, Mark. 2014. The partially examined life: Gödel on math. URL <http://www.partiallyexaminedlife.com/2014/06/16/ep95-godel/>. Podcast audio.
- MacFarlane, John. 2015. Alonzo Church's JSL reviews. URL <http://johnmacfarlane.net/church.html>.
- Maddy, Penelope. 1988a. Believing the axioms I. *The Journal of Symbolic Logic* 53(2): 481–511.
- Maddy, Penelope. 1988b. Believing the axioms II. *The Journal of Symbolic Logic* 53(3): 736–64.
- Magnus, P. D., Tim Button, J. Robert Loftis, Aaron Thomas-Bolduc, Robert Trueman, and Richard Zach. 2021. *Forall x: Calgary. An Introduction to Formal Logic*. Calgary: Open Logic Project, f21 ed. URL <https://forallx.openlogicproject.org/>.
- Matijasevich, Yuri. 1992. My collaboration with Julia Robinson. *The Mathematical Intelligencer* 14(4): 38–45.
- Menzler-Trott, Eckart. 2007. *Logic's Lost Genius: The Life of Gerhard Gentzen*. Providence: American Mathematical Society.
- Montague, Richard. 1961. Semantic closure and non-finite axiomatizability I. In *Infinistic Methods: Proceedings of the Symposium on Foundations of Mathematics (Warsaw 1959)*, 45–69. New York: Pergamon.
- Montague, Richard. 1965. Set theory and higher-order logic. In *Formal systems and recursive functions*, eds. John Crossley and Michael Dummett, 131–48. Amsterdam: North-Holland. Proceedings of the Eight Logic Colloquium, July 1963.
- O'Connor, John J. and Edmund F. Robertson. 2005. The real numbers: Stevin to Hilbert URL http://www-history.mcs.st-and.ac.uk/HistTopics/Real_numbers_2.html.
- O'Connor, John J. and Edmund F. Robertson. 2014. Rózsa Péter. URL <http://www-groups.dcs.st-and.ac.uk/~history/Biographies/Peter.html>.
- Peano, Giuseppe. 1890. Sur une courbe, qui remplit toute une aire plane. *Mathematische Annalen* 36(1): 157–60.

REFERÊNCIAS BIBLIOGRÁFICAS

- Péter, Rózsa. 1935a. Über den Zusammenhang der verschiedenen Begriffe der rekursiven Funktion. *Mathematische Annalen* 110: 612–632.
- Péter, Rózsa. 1935b. Konstruktion nichtrekursiver Funktionen. *Mathematische Annalen* 111: 42–60.
- Péter, Rózsa. 1951. *Rekursive Funktionen*. Budapest: Akademiai Kiado. English translation in (Péter, 1967).
- Péter, Rózsa. 1967. *Recursive Functions*. New York: Academic Press.
- Péter, Rózsa. 2010. *Playing with Infinity*. New York: Dover. URL https://books.google.ca/books?id=6V3wNs4uv_4C&lpg=PP1&ots=BkQZaHcR99&lr&pg=PP1#v=onepage&q&f=false.
- Potter, Michael. 2004. *Set Theory and its Philosophy*. Oxford: Oxford University Press.
- Radiolab. 2012. The Turing problem. URL <http://www.radiolab.org/story/193037-turing-problem/>. Podcast audio.
- Ramsey, Frank Plumpton. 1925. The foundations of mathematics. *Proceedings of the London Mathematical Society* 25: 338–384.
- Reid, Constance. 1986. The autobiography of Julia Robinson. *The College Mathematics Journal* 17: 3–21.
- Reid, Constance. 1996. *Julia: A Life in Mathematics*. Cambridge: Cambridge University Press. URL <https://books.google.ca/books?id=1RtSzQyHf9UC&lpg=PP1&pg=PP1#v=onepage&q&f=false>.
- Robinson, Julia. 1949. Definability and decision problems in arithmetic. *The Journal of Symbolic Logic* 14(2): 98–114. URL <http://www.jstor.org/stable/2266510>.
- Robinson, Julia. 1996. *The Collected Works of Julia Robinson*. Providence: American Mathematical Society.
- Robinson, Raphael. 1947. On the decomposition of spheres. *Fundamenta Mathematicae* 34(1): 246–60.
- Rose, Daniel. 2012. A song about Georg Cantor. URL <https://www.youtube.com/watch?v=QUP5Z4Fb5k4>. Audio Recording.
- Rose, Nicholas J. 2010. Hilbert-type space-filling curves. URL <https://web.archive.org/web/20151010184939/http://www4.ncsu.edu/~njrose/pdfFiles/HilbertCurve.pdf>.
- Russell, Bertrand. 1905. On denoting. *Mind* 14: 479–493.
- Russell, Bertrand. 1919. *Introduction to Mathematical Philosophy*. London: Allen & Unwin.

REFERÊNCIAS BIBLIOGRÁFICAS

- Russell, Bertrand. 1967. *The Autobiography of Bertrand Russell*, vol. 1. London: Allen and Unwin.
- Russell, Bertrand. 1968. *The Autobiography of Bertrand Russell*, vol. 2. London: Allen and Unwin.
- Russell, Bertrand. 1969. *The Autobiography of Bertrand Russell*, vol. 3. London: Allen and Unwin.
- Russell, Bertrand. n.d. Bertrand Russell on smoking. URL https://www.youtube.com/watch?v=80oLTiVW_1c. Video Interview.
- Sandstrum, Ted. 2019. *Mathematical Reasoning: Writing and Proof*. Allendale, MI: Grand Valley State University. URL <https://scholarworks.gvsu.edu/books/7/>.
- Scott, Dana. 1974. Axiomatizing set theory. In *Axiomatic Set Theory II*, ed. Thomas Jech, 207–14. American Mathematical Society. Proceedings of the Symposium in Pure Mathematics of the American Mathematical Society, July–August 1967.
- Segal, Sanford L. 2014. *Mathematicians under the Nazis*. Princeton: Princeton University Press.
- Shoenfield, Joseph R. 1977. Axioms of set theory. In *Handbook of Mathematical Logic*, ed. Jon Barwise, 321–44. London: North-Holland.
- Sigmund, Karl, John Dawson, Kurt Mühlberger, Hans Magnus Enzensberger, and Juliette Kennedy. 2007. Kurt Gödel: Das Album–The Album. *The Mathematical Intelligencer* 29(3): 73–76.
- Skolem, Thoralf. 1922. Einige Bemerkungen zur axiomatischen Begründung der Mengenlehre. In *Wissenschaftliche Vorträge gehalten auf dem fünften Kongress der skandinavischen Mathematiker in Helsingfors vom 4. bis zum 7. Juli 1922*, 137–52. Akademiska Bokhandeln.
- Smith, Peter. 2013. *An Introduction to Gödel's Theorems*. Cambridge: Cambridge University Press.
- Smullyan, Raymond M. 1968. *First-Order Logic*. New York, NY: Springer. Corrected reprint, New York, NY: Dover, 1995.
- Solow, Daniel. 2013. *How to Read and Do Proofs*. Hoboken, NJ: Wiley.
- Steinhart, Eric. 2018. *More Precisely: The Math You Need to Do Philosophy*. Peterborough, ON: Broadview, 2nd ed.
- Sykes, Christopher. 1992. BBC Horizon: The strange life and death of Dr. Turing. URL <https://www.youtube.com/watch?v=gyusnGbBSHE>.

REFERÊNCIAS BIBLIOGRÁFICAS

- Szabo, Manfred E. 1969. *The Collected Papers of Gerhard Gentzen*. Amsterdam: North-Holland.
- Takeuti, Gaisi, Nicholas Passell, and Mariko Yasugi. 2003. *Memoirs of a Proof Theorist: Gödel and Other Logicians*. Singapore: World Scientific.
- Tamassy, Istvan. 1994. Interview with Róza Péter. *Modern Logic* 4(3): 277–280.
- Tarski, Alfred. 1981. *The Collected Works of Alfred Tarski*, vol. I–IV. Basel: Birkhäuser.
- Theelen, Andre. 2012. Lego turing machine. URL <https://www.youtube.com/watch?v=FTSAiF9AHN4>.
- Tomkowicz, Grzegorz and Stan Wagon. 2016. *The Banach-Tarski Paradox*. Cambridge: Cambridge University Press.
- Turing, Alan M. 1937. On computable numbers, with an application to the “Entscheidungsproblem”. *Proceedings of the London Mathematical Society, 2nd Series* 42: 230–265.
- Tyldum, Morten. 2014. The imitation game. Motion picture.
- Velleman, Daniel J. 2019. *How to Prove It: A Structured Approach*. Cambridge: Cambridge University Press, 3rd ed.
- Vitali, Giuseppe. 1905. *Sul problema della misura dei gruppi di punti di una retta*. Bologna: Gamberini e Parmeggiani.
- von Neumann, John. 1925. Eine Axiomatisierung der Mengenlehre. *Journal für die reine und angewandte Mathematik* 154: 219–40.
- Wang, Hao. 1990. *Reflections on Kurt Gödel*. Cambridge: MIT Press.
- Weston, Tom. 2003. The Banach-Tarski paradox URL <http://people.math.umass.edu/~weston/oldpapers/banach.pdf>.
- Whitehead, Alfred North and Bertrand Russell. 1910. *Principia Mathematica*, vol. 1. Cambridge: Cambridge University Press.
- Zermelo, Ernst. 1904. Beweis, daß jede Menge wohlgeordnet werden kann. *Mathematische Annalen* 59: 514–516. English translation in (Ebbinghaus et al., 2010, pp. 115–119).
- Zermelo, Ernst. 1908a. Untersuchungen über die Grundlagen der Mengenlehre I. *Mathematische Annalen* 65: 261–81.
- Zermelo, Ernst. 1908b. Untersuchungen über die Grundlagen der Mengenlehre I. *Mathematische Annalen* 65(2): 261–281. English translation in (Ebbinghaus et al., 2010, pp. 189–229).

REFERÊNCIAS BIBLIOGRÁFICAS

Zuckerman, Martin M. 1973. Formation sequences for propositional formulas.
Notre Dame Journal of Formal Logic 14(1): 134–138.