OLP: Example à la Enderton

Open Logic Project

Revision: fad6b55 (2017-08-26)

Contents

Part I

First-order Logic

Chapter 1

Syntax and Semantics

1.1 Introduction

In order to develop the theory and metatheory of first-order logic, we must first define the syntax and semantics of its expressions. The expressions of first-order logic are terms and wffs. Terms are formed from variables, constant symbols, and function symbols. Wffs, in turn, are formed from predicate symbols together with terms (these form the smallest, "atomic" wffs), and then from atomic wffs we can form more complex ones using logical connectives and quantifiers. There are many different ways to set down the formation rules; we give just one possible one. Other systems will chose different symbols, will select different sets of connectives as primitive, 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 terms and wffs *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 is that of satisfaction in a structure. A structure gives meaning to the building blocks of the language: a domain is a non-empty set of objects. The quantifiers are interpreted as ranging over this domain, constant symbols are assigned elements in the domain, function symbols are assigned functions from the domain to itself, and predicate symbols are assigned relations on the domain. The domain together with assignments to the basic vocabulary constitutes a structure. Variables may appear in wffs, and in order to give a semantics, we also have to assign elements of the domain to them—this is a variable assignment. The satisfaction relation, finally, brings these together. A wff may be satisfied in a structure $\mathfrak{A}$ relative to a variable assignment s, written as $\models_{\mathfrak{A}} \alpha[s]$. This relation is also defined by induction on the structure of α, using the truth tables for the logical connectives to define, say, satisfaction of $\alpha \wedge \beta$ in terms of satisfaction (or not) of α and β. It then turns out that

the variable assignment is irrelevant if the wff α is a sentence, i.e., has no free variables, and so we can talk of sentences being simply satisfied (or not) in structures.

On the basis of the satisfaction relation $\models_{\mathfrak{A}} \alpha$ for sentences we can then define the basic semantic notions of validity, entailment, and satisfiability. A sentence is valid, $\models \alpha$, if every structure satisfies it. It is entailed by a set of sentences, $\Gamma \models \alpha$, if every structure that satisfies all the sentences in Γ also satisfies α. And a set of sentences is satisfiable if some structure satisfies all sentences in it at the same time. Because wffs are inductively defined, and satisfaction is in turn defined by induction on the structure of wffs, we can use induction to prove properties of our semantics and to relate the semantic notions defined.

1.2 First-Order Languages

Expressions of first-order logic are built up from a basic vocabulary containing *variables*, *constant symbols*, *predicate symbols* and sometimes *function symbols*. From them, together with logical connectives, quantifiers, and punctuation symbols such as parentheses and commas, *terms* and *wffs* are formed.

Informally, predicate symbols are names for properties and relations, constant symbols are names for individual objects, and function symbols are names for mappings. These, except for the equality symbol $=$, are the *non-logical symbols* and together make up a language. Any first-order language $\mathcal{L}$ is determined by its non-logical symbols. In the most general case, $\mathcal{L}$ contains infinitely many symbols of each kind.

In the general case, we make use of the following symbols in first-order logic:

1. Logical symbols
   a) Logical connectives: $\neg$ (negation), $\rightarrow$ (conditional), $\forall$ (universal quantifier).
   b) The two-place equality symbol $=$.
   c) A denumerable set of variables: v_0, v_1, v_2, ...
2. Non-logical symbols, making up the *standard language* of first-order logic
   a) A denumerable set of n-place predicate symbols for each $n > 0$: A^n_0, A^n_1, A^n_2, ...
   b) A denumerable set of constant symbols: c_0, c_1, c_2, ...
   c) A denumerable set of n-place function symbols for each $n > 0$: f^n_0, f^n_1, f^n_2, ...
3. Punctuation marks: (,), and the comma.

Most of our definitions and results will be formulated for the full standard language of first-order logic. However, depending on the application, we may

also restrict the language to only a few predicate symbols, constant symbols, and function symbols.

Example 1.2.1. The language $\mathcal{L}_A$ of arithmetic contains a single two-place predicate symbol $<$, a single constant symbol 0, one one-place function symbol $'$, and two two-place function symbols $+$ and $\times$.

Example 1.2.2. The language of set theory $\mathcal{L}_Z$ contains only the single two-place predicate symbol $\in$.

Example 1.2.3. The language of orders $\mathcal{L}_\leq$ contains only the two-place predicate symbol $\leq$.

Again, these are conventions: officially, these are just aliases, e.g., $<$, $\in$, and $\leq$ are aliases for A^2_0, 0 for c_0, $'$ for f^1_0, $+$ for f^2_0, $\times$ for f^2_1.

In addition to the primitive connectives and quantifier introduced above, we also use the following *defined* symbols: $\wedge$ (conjunction), $\vee$ (disjunction), $\leftrightarrow$ (biconditional), $\exists$ (existential quantifier), falsity $\bot$, truth $\top$

A defined symbol is not officially part of the language, but is introduced as an informal abbreviation: it allows us to abbreviate formulas which would, if we only used primitive symbols, get quite long. This is obviously an advantage. The bigger advantage, however, is that proofs become shorter. If a symbol is primitive, it has to be treated separately in proofs. The more primitive symbols, therefore, the longer our proofs.

You may be familiar with different terminology and symbols than the ones 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 $\bot$ symbol is variously called "falsity," "falsum,", "absurdity,", or "bottom." The $\top$ symbol is variously called "truth," "verum,", or "top."

It is conventional to use lower case letters (e.g., a, b, c) from the beginning of the Latin alphabet for constant symbols (sometimes called names), and lower case letters from the end (e.g., x, y, z) for variables. Quantifiers combine with variables, e.g., x; notational variations include $\forall x$, $(\forall x)$, (x), Πx, $\bigwedge_x$ for the universal quantifier and $\exists x$, $(\exists x)$, (Ex), Σx, $\bigvee_x$ for the existential quantifier.

We might treat all the propositional operators and both quantifiers as primitive symbols of the language. We might instead choose a smaller stock of primitive symbols and treat the other logical operators as defined. "Truth functionally complete" sets of Boolean operators include $\{\neg, \vee\}$, $\{\neg, \wedge\}$, and $\{\neg, \rightarrow\}$—these can be combined with either quantifier for an expressively complete first-order language.

You may be familiar with two other logical operators: the Sheffer stroke $|$ (named after Henry Sheffer), and Peirce's arrow $\downarrow$, also known as Quine's dagger. When given their usual readings of "nand" and "nor" (respectively), these operators are truth functionally complete by themselves.

1.3 Terms and Wffs

Once a first-order language $\mathcal{L}$ is given, we can define expressions built up from the basic vocabulary of $\mathcal{L}$. These include in particular *terms* and *wffs*.

DEFINITION 13A (TERMS) The set of *terms* $\mathrm{Trm}(\mathcal{L})$ of $\mathcal{L}$ is defined inductively by:

1. Every variable is a term.
2. Every constant symbol of $\mathcal{L}$ is a term.
3. If f is an n-place function symbol and $t_1, \ldots, t_n$ are terms, then $ft_1 \ldots t_n$ is a term.

A term containing no variables is a *closed term*.

The constant symbols appear in our specification of the language and the terms as a separate category of symbols, but they could instead have been included as zero-place function symbols. We could then do without the second clause in the definition of terms. We just have to understand $ft_1 \ldots t_n$ as just f by itself if $n = 0$.

DEFINITION 13B (FORMULA) The set of *wffs* $\mathrm{Frm}(\mathcal{L})$ of the language $\mathcal{L}$ is defined inductively as follows:

1. If R is an n-place predicate symbol of $\mathcal{L}$ and $t_1, \ldots, t_n$ are terms of $\mathcal{L}$, then $Rt_1 \ldots t_n$ is an atomic wff.
2. If t_1 and t_2 are terms of $\mathcal{L}$, then $= t_1 t_2$ is an atomic wff.
3. If α is a wff, then $\neg\alpha$ is wff.
4. If α and β are wffs, then $(\alpha \to \beta)$ is a wff.
5. If α is a wff and x is a variable, then $\forall x\, \alpha$ is a wff.

The definitions of the set of terms and that of wffs are *inductive definitions*. Essentially, we construct the set of wffs 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, i.e., the cases for $Rt_1 \ldots t_n$ and $= t_1 t_2$. "Atomic wff" thus means any wff of this form.

The other cases of the definition give rules for constructing new wffs out of wffs already constructed. At the second stage, we can use them to construct wffs out of atomic wffs. At the third stage, we construct new formulas from the atomic formulas and those obtained in the second stage, and so on. A wff is anything that is eventually constructed at such a stage, and nothing else.

By convention, we write $=$ between its arguments and leave out the parentheses: $t_1 = t_2$ is an abbreviation for $= t_1 t_2$. Moreover, $\neg = t_1 t_2$ is abbreviated as $t_1 \neq t_2$. When writing a formula $(\beta * \gamma)$ constructed from β, γ using a two-place connective $*$, we will often leave out the outermost pair of parentheses and write simply $\beta * \gamma$.

Some logic texts require that the variable x must occur in α in order for $\forall x\,\alpha$ to count as a wff. Nothing bad happens if you don't require this, and it makes things easier.

Definition 13C Formulas constructed using the defined operators are to be understood as follows:

1. $\top$ abbreviates $(\alpha \vee \neg\alpha)$ for some fixed atomic wff α.
2. $\bot$ abbreviates $(\alpha \wedge \neg\alpha)$ for some fixed atomic wff α.
3. $\alpha \vee \beta$ abbreviates $\neg\alpha \to \beta$.
4. $\alpha \wedge \beta$ abbreviates $\neg(\alpha \to \neg\beta)$.
5. $\alpha \leftrightarrow \beta$ abbreviates $(\alpha \to \beta) \wedge (\beta \to \alpha)$.
6. $\exists x\,\alpha$ abbreviates $\neg\forall x\,\neg\alpha$.

If we work in a language for a specific application, we will often write two-place predicate symbols and function symbols between the respective terms, e.g., $t_1 < t_2$ and $(t_1 + t_2)$ in the language of arithmetic and $t_1 \in t_2$ in the language of set theory. The successor function in the language of arithmetic is even written conventionally *after* its argument: t'. Officially, however, these are just conventional abbreviations for $A^2_0(t_1, t_2)$, $f^2_0(t_1, t_2)$, $A^2_0(t_1, t_2)$ and $f^1_0(t)$, respectively.

Definition 13D (Syntactic identity) The symbol $\equiv$ expresses syntactic identity between strings of symbols, i.e., $\alpha \equiv \beta$ iff α and β 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, e.g., $\alpha \equiv (\beta \vee \gamma)$ means: the string of symbols α is the same string as the one obtained by concatenating an opening parenthesis, the string β, the $\vee$ symbol, the string γ, and a closing parenthesis, in this order. If this is the case, then we know that the first symbol of α is an opening parenthesis, α contains β as a substring (starting at the second symbol), that substring is followed by $\vee$, etc.

1.4 Unique Readability

The way we defined wffs guarantees that every wff has a *unique reading*, i.e., there is essentially only one way of constructing it according to our formation rules for wffs and only one way of "interpreting" it. If this were not so, we would have ambiguous wffs, i.e., wffs that have more than one reading or intepretation—and that is clearly something we want to avoid. But more importantly, without this property, most of the definitions and proofs we are going to give will not go through.

Perhaps the best way to make this clear is to see what would happen if we had given bad rules for forming wffs that would not guarantee unique readability. For instance, we could have forgotten the parentheses in the formation rules for connectives, e.g., we might have allowed this:

If α and β are wffs, then so is $\alpha \to \beta$.

Starting from an atomic formula δ, this would allow us to form $\delta \to \delta$. From this, together with δ, we would get $\delta \to \delta \to \delta$. But there are two ways to do this:

1. We take δ to be α and $\delta \to \delta$ to be β.

2. We take α to be $\delta \to \delta$ and β is δ.

Correspondingly, there are two ways to "read" the wff $\delta \to \delta \to \delta$. It is of the form $\beta \to \gamma$ where β is δ and γ is $\delta \to \delta$, but *it is also* of the form $\beta \to \gamma$ with β being $\delta \to \delta$ and γ being δ.

If this happens, our definitions will not always work. For instance, when we define the main operator of a formula, we say: in a formula of the form $\beta \to \gamma$, the main operator is the indicated occurrence of $\to$. But if we can match the formula $\delta \to \delta \to \delta$ with $\beta \to \gamma$ in the two different ways mentioned above, then in one case we get the first occurrence of $\to$ as the main operator, and in the second case the second occurrence. But we intend the main operator to be a *function* of the wff, i.e., every wff must have exactly one main operator occurrence.

LEMMA 14A The number of left and right parentheses in a wff α are equal.

Proof. We prove this by induction on the way α is constructed. This requires two things: (a) We have to prove first that all atomic formulas have the property in question (the induction basis). (b) Then we have to prove that when we construct new formulas out of given formulas, the new formulas have the property provided the old ones do.

Let $l(\alpha)$ be the number of left parentheses, and $r(\alpha)$ the number of right parentheses in α, and $l(t)$ and $r(t)$ similarly the number of left and right parentheses in a term t. We leave the proof that for any term t, $l(t) = r(t)$ as an exercise.

1. $\alpha \equiv Rt_1 \dots t_n$: $l(\alpha) = 1+l(t_1)+\dots+l(t_n) = 1+r(t_1)+\dots+r(t_n) = r(\alpha)$. Here we make use of the fact, left as an exercise, that $l(t) = r(t)$ for any term t.

2. $\alpha \equiv t_1 = t_2$: $l(\alpha) = l(t_1) + l(t_2) = r(t_1) + r(t_2) = r(\alpha)$.

3. $\alpha \equiv \neg\beta$: By induction hypothesis, $l(\beta) = r(\beta)$. Thus $l(\alpha) = l(\beta) = r(\beta) = r(\alpha)$.

4. $\alpha \equiv (\beta * \gamma)$: By induction hypothesis, $l(\beta) = r(\beta)$ and $l(\gamma) = r(\gamma)$. Thus $l(\alpha) = 1 + l(\beta) + l(\gamma) = 1 + r(\beta) + r(\gamma) = r(\alpha)$.

5. $\alpha \equiv \forall x\, \beta$: By induction hypothesis, $l(\beta) = r(\beta)$. Thus, $l(\alpha) = l(\beta) = r(\beta) = r(\alpha)$.

□

Definition 14B (Proper prefix) A string of symbols β is a *proper prefix* of a string of symbols α if concatenating β and a non-empty string of symbols yields α.

Lemma 14C If α is a wff, and β is a proper prefix of α, then β is not a wff.

Proof. Exercise. □

Proposition 14D If α is an atomic wff, then it satisfes one, and only one of the following conditions.

1. $\alpha \equiv Rt_1 \dots t_n$ where R is an n-place predicate symbol, t_1, ..., t_n are terms, and each of R, t_1, ..., t_n is uniquely determined.

2. $\alpha \equiv t_1 = t_2$ where t_1 and t_2 are uniquely determined terms.

Proof. Exercise. □

Proposition 14E (Unique Readability) Every wff satisfies one, and only one of the following conditions.

1. α is atomic.

2. α is of the form $\neg\beta$.

3. α is of the form $(\beta \to \gamma)$.

4. α is of the form $\forall x\, \beta$.

Moreover, in each case β, or β and γ, are uniquely determined. This means that, e.g., there are no different pairs β, γ and β', γ' so that α is both of the form $(\beta \to \gamma)$ and $(\beta' \to \gamma')$.

Proof. The formation rules require that if a wff is not atomic, it must start with an opening parenthesis (, $\neg$, or with a quantifier. On the other hand, every wff that start with one of the following symbols must be atomic: a predicate symbol, a function symbol, a constant symbol.

So we really only have to show that if α is of the form $(\beta * \gamma)$ and also of the form $(\beta' *' \gamma')$, then $\beta \equiv \beta'$, $\gamma \equiv \gamma'$, and $* = *'$.

So suppose both $\alpha \equiv (\beta * \gamma)$ and $\alpha \equiv (\beta' *' \gamma')$. Then either $\beta \equiv \beta'$ or not. If it is, clearly $* = *'$ and $\gamma \equiv \gamma'$, since they then are substrings of α that begin in the same place and are of the same length. The other case is $\beta \not\equiv \beta'$. Since β and β' are both substrings of α that begin at the same place, one must be a proper prefix of the other. But this is impossible by Lemma 14C. □

1.5 Main operator of a Formula

It is often useful to talk about the last operator used in constructing a wff α. This operator is called the *main operator* of α. Intuitively, it is the "outermost" operator of α. For example, the main operator of $\neg\alpha$ is $\neg$, the main operator of $(\alpha \vee \beta)$ is $\vee$, etc.

Definition 15A (Main operator) The *main operator* of a wff α is defined as follows:

1. α is atomic: α has no main operator.
2. $\alpha \equiv \neg\beta$: the main operator of α is $\neg$.
3. $\alpha \equiv (\beta \to \gamma)$: the main operator of α is $\to$.
4. $\alpha \equiv \forall x\, \beta$: the main operator of α is $\forall$.

In each case, we intend the specific indicated *occurrence* of the main operator in the formula. For instance, since the formula $((\delta \to \phi) \to (\phi \to \delta))$ is of the form $(\beta \to \gamma)$ where β is $(\delta \to \phi)$ and γ is $(\phi \to \delta)$, the second occurrence of $\to$ is the main operator.

This is a *recursive* definition of a function which maps all non-atomic wffs to their main operator occurrence. Because of the way wffs are defined inductively, every wff α satisfies one of the cases in Definition 15A. This guarantees that for each non-atomic wff α a main operator exists. Because each wff satisfies only one of these conditions, and because the smaller wffs from which α is constructed are uniquely determined in each case, the main operator occurrence of α is unique, and so we have defined a function.

We call wffs by the following names depending on which symbol their main operator is:

Main operator	Type of wff	Example
none	atomic (wff)	$Rt_1 \dots t_n$, $t_1 = t_2$
$\neg$	negation	$\neg\alpha$
$\wedge$	conjunction	$(\alpha \wedge \beta)$
$\vee$	disjunction	$(\alpha \vee \beta)$
$\to$	conditional	$(\alpha \to \beta)$
$\forall$	universal (wff)	$\forall x\, \alpha$
$\exists$	existential (wff)	$\exists x\, \alpha$

1.6 Subformulas

It is often useful to talk about the wffs that "make up" a given wff. We call these its *subformulas*. Any wff counts as a subformula of itself; a subformula of α other than α itself is a *proper subformula*.

Definition 16A (Immediate Subformula) If α is a wff, the *immediate subformulas* of α are defined inductively as follows:

1. Atomic wffs have no immediate subformulas.
2. $\alpha \equiv \neg\beta$: The only immediate subformula of α is β.
3. $\alpha \equiv (\beta * \gamma)$: The immediate subformulas of α are β and γ ($*$ is any one of the two-place connectives).
4. $\alpha \equiv \forall x\, \beta$: The only immediate subformula of α is β.

Definition 16B (Proper Subformula) If α is a wff, the *proper subformulas* of α are recursively as follows:

1. Atomic wffs have no proper subformulas.
2. $\alpha \equiv \neg\beta$: The proper subformulas of α are β together with all proper subformulas of β.
3. $\alpha \equiv (\beta * \gamma)$: The proper subformulas of α are β, γ, together with all proper subformulas of β and those of γ.
4. $\alpha \equiv \forall x\, \beta$: The proper subformulas of α are β together with all proper subformulas of β.

Definition 16C (Subformula) The subformulas of α are α itself together with all its proper subformulas.

Note the subtle difference in how we have defined immediate subformulas and proper subformulas. In the first case, we have directly defined the immediate subformulas of a formula α for each possible form of α. It is an explicit definition by cases, and the cases mirror the inductive definition of the set of wffs. In the second case, we have also mirrored the way the set of all wffs is defined, but in each case we have also included the proper subformulas of the smaller wffs β, γ in addition to these wffs themselves. This makes the definition *recursive*. In general, a definition of a function on an inductively defined set (in our case, wffs) is recursive if the cases in the definition of the function make use of the function itself. To be well defined, we must make sure, however, that we only ever use the values of the function for arguments that come "before" the one we are defining—in our case, when defining "proper subformula" for $(\beta * \gamma)$ we only use the proper subformulas of the "earlier" wffs β and γ.

1.7 Free Variables and Sentences

Definition 17A (Free occurrences of a variable) The *free* occurrences of a variable in a wff are defined inductively as follows:

1. α is atomic: all variable occurrences in α are free.
2. $\alpha \equiv \neg\beta$: the free variable occurrences of α are exactly those of β.
3. $\alpha \equiv (\beta * \gamma)$: the free variable occurrences of α are those in β together with those in γ.

4. $\alpha \equiv \forall x\, \beta$: the free variable occurrences in α are all of those in β except for occurrences of x.

DEFINITION 17B (BOUND VARIABLES) An occurrence of a variable in a formula α is *bound* if it is not free.

DEFINITION 17C (SCOPE) If $\forall x\, \beta$ is an occurrence of a subformula in a formula α, then the corresponding occurrence of β in α is called the *scope* of the corresponding occurrence of $\forall x$.

If β is the scope of a quantifier occurrence $\forall x$ in α, then all occurrences of x which are free in β are said to be *bound by* the mentioned quantifier occurrence.

Example 1.7.4. Consider the following formula:

$$\exists v_0 \underbrace{A_0^2 v_0 v_1}_{\beta}$$

β represents the scope of $\exists v_0$. The quantifier binds the occurence of v_0 in β, but does not bind the occurence of v_1. So v_1 is a free variable in this case.

We can now see how this might work in a more complicated wff α:

$$\forall v_0 \underbrace{(A_0^1 v_0 \to A_0^2 v_0 v_1)}_{\beta} \to \exists v_1 \underbrace{(A_1^2 v_0 v_1 \vee \forall v_0 \overbrace{\neg A_1^1 v_0}^{\delta})}_{\gamma}$$

β is the scope of the first $\forall v_0$, γ is the scope of $\exists v_1$, and δ is the scope of the second $\forall v_0$. The first $\forall v_0$ binds the occurrences of v_0 in β, $\exists v_1$ the occurrence of v_1 in γ, and the second $\forall v_0$ binds the occurrence of v_0 in δ. The first occurrence of v_1 and the fourth occurrence of v_0 are free in α. The last occurrence of v_0 is free in δ, but bound in γ and α.

DEFINITION 17E (SENTENCE) A wff α is a *sentence* iff it contains no free occurrences of variables.

1.8 Substitution

DEFINITION 18A (SUBSTITUTION IN A TERM) We define $s[t/x]$, the result of *substituting* t for every occurrence of x in s, recursively:

1. $s \equiv c$: $s[t/x]$ is just s.
2. $s \equiv y$: $s[t/x]$ is also just s, provided y is a variable and $y \not\equiv x$.
3. $s \equiv x$: $s[t/x]$ is t.
4. $s \equiv f t_1 \dots t_n$: $s[t/x]$ is $f t_1[t/x] \dots t_n[t/x]$.

DEFINITION 18B A term t is *free for* x in α if none of the free occurrences of x in α occur in the scope of a quantifier that binds a variable in t.

Example 1.8.3.

1. v_8 is free for v_1 in $\exists v_3 A_4^2 v_3 v_1$
2. $f_1^2(v_1, v_2)$ is *not* free for v_o in $\forall v_2 A_4^2 v_0 v_2$

DEFINITION 18D (SUBSTITUTION IN A WFF) If α is a wff, x is a variable, and t is a term free for x in α, then $\alpha[t/x]$ is the result of substituting t for all free occurrences of x in α.

1. $\alpha \equiv Pt_1 \dots t_n$: $\alpha[t/x]$ is $Pt_1[t/x] \dots t_n[t/x]$.
2. $\alpha \equiv t_1 = t_2$: $\alpha[t/x]$ is $t_1[t/x] = t_2[t/x]$.
3. $\alpha \equiv \neg\beta$: $\alpha[t/x]$ is $\neg\beta[t/x]$.
4. $\alpha \equiv (\beta \to \gamma)$: $\alpha[t/x]$ is $(\beta[t/x] \to \gamma[t/x])$.
5. $\alpha \equiv \forall y\, \beta$: $\alpha[t/x]$ is $\forall y\, \beta[t/x]$, provided y is a variable other than x; otherwise $\alpha[t/x]$ is just α.

Note that substitution may be vacuous: If x does not occur in α at all, then $\alpha[t/x]$ is just α.

The restriction that t must be free for x in α is necessary to exclude cases like the following. If $\alpha \equiv \exists y\, x < y$ and $t \equiv y$, then $\alpha[t/x]$ would be $\exists y\, y < y$. In this case the free variable y is "captured" by the quantifier $\exists y$ upon substitution, and that is undesirable. For instance, we would like it to be the case that whenever $\forall x\, \beta$ holds, so does $\beta[t/x]$. But consider $\forall x \exists y\, x < y$ (here β is $\exists y\, x < y$). It is sentence that is true about, e.g., the natural numbers: for every number x there is a number y greater than it. If we allowed y as a possible substitution for x, we would end up with $\beta[y/x] \equiv \exists y\, y < y$, which is false. We prevent this by requiring that none of the free variables in t would end up being bound by a quantifier in α.

We often use the following convention to avoid cumbersume notation: If α is a formula with a free variable x, we write $\alpha(x)$ to indicate this. When it is clear which α and x we have in mind, and t is a term (assumed to be free for x in $\alpha(x)$), then we write $\alpha(t)$ as short for $\alpha(x)[t/x]$.

1.9 Structures for First-order Languages

First-order languages are, by themselves, *uninterpreted:* the constant symbols, function symbols, and predicate symbols have no specific meaning attached to them. Meanings are given by specifying a *structure*. It specifies the *domain*, i.e., the objects which the constant symbols pick out, the function symbols operate on, and the quantifiers range over. In addition, it specifies which constant symbols pick out which objects, how a function symbol maps objects to objects, and which objects the predicate symbols apply to. Structures are the basis for *semantic* notions in logic, e.g., the notion of consequence, validity, satisfiablity. They are variously called "structures," "interpretations," or "models" in the literature.

Definition 19A (Structures) A *structure* $\mathfrak{A}$, for a language $\mathcal{L}$ of first-order logic consists of the following elements:

1. *Domain:* a non-empty set, $|\mathfrak{A}|$
2. *Interpretation of constant symbols:* for each constant symbol c of $\mathcal{L}$, an element $c^{\mathfrak{A}} \in |\mathfrak{A}|$
3. *Interpretation of predicate symbols:* for each n-place predicate symbol R of $\mathcal{L}$ (other than $=$), an n-place relation $R^{\mathfrak{A}} \subseteq |\mathfrak{A}|^n$
4. *Interpretation of function symbols:* for each n-place function symbol f of $\mathcal{L}$, an n-place function $f^{\mathfrak{A}} \colon |\mathfrak{A}|^n \to |\mathfrak{A}|$

Example 1.9.2. A structure $\mathfrak{A}$ for the language of arithmetic consists of a set, an element of $|\mathfrak{A}|$, $0^{\mathfrak{A}}$, as interpretation of the constant symbol 0, a one-place function $\prime^{\mathfrak{A}} \colon |\mathfrak{A}| \to |\mathfrak{A}|$, two two-place functions $+^{\mathfrak{A}}$ and $\times^{\mathfrak{A}}$, both $|\mathfrak{A}|^2 \to |\mathfrak{A}|$, and a two-place relation $<^{\mathfrak{A}} \subseteq |\mathfrak{A}|^2$.

An obvious example of such a structure is the following:

1. $|\mathfrak{B}| = \mathbb{N}$
2. $0^{\mathfrak{B}} = 0$
3. $\prime^{\mathfrak{B}}(n) = n + 1$ for all $n \in \mathbb{N}$
4. $+^{\mathfrak{B}}(n, m) = n + m$ for all $n, m \in \mathbb{N}$
5. $\times^{\mathfrak{B}}(n, m) = n \cdot m$ for all $n, m \in \mathbb{N}$
6. $<^{\mathfrak{B}} = \{\langle n, m\rangle : n \in \mathbb{N}, m \in \mathbb{N}, n < m\}$

The structure $\mathfrak{B}$ for $\mathcal{L}_A$ so defined is called the *standard model of arithmetic*, because it interprets the non-logical constants of $\mathcal{L}_A$ exactly how you would expect.

However, there are many other possible structures for $\mathcal{L}_A$. For instance, we might take as the domain the set $\mathbb{Z}$ of integers instead of $\mathbb{N}$, and define the interpretations of 0, $\prime$, $+$, $\times$, $<$ accordingly. But we can also define structures for $\mathcal{L}_A$ which have nothing even remotely to do with numbers.

Example 1.9.3. A structure $\mathfrak{A}$ for the language $\mathcal{L}_Z$ of set theory requires just a set and a single-two place relation. So technically, e.g., the set of people plus the relation "x is older than y" could be used as a structure for $\mathcal{L}_Z$, as well as $\mathbb{N}$ together with $n \geq m$ for $n, m \in \mathbb{N}$.

A particularly interesting structure for $\mathcal{L}_Z$ in which the elements of the domain are actually sets, and the interpretation of $\in$ actually is the relation "x is an element of y" is the structure $\mathfrak{HF}$ of *hereditarily finite sets*:

1. $|\mathfrak{HF}| = \emptyset \cup \wp(\emptyset) \cup \wp(\wp(\emptyset)) \cup \wp(\wp(\wp(\emptyset))) \cup \ldots$;
2. $\in^{\mathfrak{HF}} = \{\langle x, y\rangle : x, y \in |\mathfrak{HF}|, x \in y\}$.

The stipulations we make as to what counts as a structure impact our logic. For example, the choice to prevent empty domains ensures, given the usual account of satisfaction (or truth) for quantified sentences, that $\exists x\,(\alpha(x) \vee \neg\alpha(x))$ is valid—that is, a logical truth. And the stipulation that all constant symbols must refer to an object in the domain ensures that the existential generalization is a sound pattern of inference: $\alpha(a)$, therefore $\exists x\,\alpha(x)$. If we allowed names to refer outside the domain, or to not refer, then we would be on our way to a *free logic*, in which existential generalization requires an additional premise: $\alpha(a)$ and $\exists x\, x = a$, therefore $\exists x\,\alpha(x)$.

1.10 Covered Structures for First-order Languages

Recall that a term is *closed* if it contains no variables.

Definition 110A (Value of closed terms) If t is a closed term of the language $\mathcal{L}$ and $\mathfrak{A}$ is a structure for $\mathcal{L}$, the *value* $t^{\mathfrak{A}}$ is defined as follows:

1. If t is just the constant symbol c, then $c^{\mathfrak{A}} = c^{\mathfrak{A}}$.
2. If t is of the form $ft_1 \dots t_n$, then
$$t^{\mathfrak{A}} = f^{\mathfrak{A}}({t_1}^{\mathfrak{A}}, \dots, {t_n}^{\mathfrak{A}}).$$

Definition 110B (Covered structure) A structure is *covered* if every element of the domain is the value of some closed term.

Example 1.10.3. Let $\mathcal{L}$ be the language with constant symbols *zero*, *one*, *two*, ..., the binary predicate symbol $<$, and the binary function symbols $+$ and $\times$. Then a structure $\mathfrak{A}$ for $\mathcal{L}$ is the one with domain $|\mathfrak{A}| = \{0, 1, 2, \dots\}$ and assignments $zero^{\mathfrak{A}} = 0$, $one^{\mathfrak{A}} = 1$, $two^{\mathfrak{A}} = 2$, and so forth. For the binary relation symbol $<$, the set $<^{\mathfrak{A}}$ is the set of all pairs $\langle c_1, c_2\rangle \in |\mathfrak{A}|^2$ such that c_1 is less than c_2: for example, $\langle 1, 3\rangle \in <^{\mathfrak{A}}$ but $\langle 2, 2\rangle \notin <^{\mathfrak{A}}$. For the binary function symbol $+$, define $+^{\mathfrak{A}}$ in the usual way—for example, $+^{\mathfrak{A}}(2, 3)$ maps to 5, and similarly for the binary function symbol $\times$. Hence, the value of $four$ is just 4, and the value of $\times(two, +(three, zero))$ (or in infix notation, $two \times (three + zero)$) is

$$\begin{aligned}
\times(two, +(three, zero)^{\mathfrak{A}} = & \\
&= \times^{\mathfrak{A}}(two^{\mathfrak{A}}, two, +(three, zero)^{\mathfrak{A}}) \\
&= \times^{\mathfrak{A}}(two^{\mathfrak{A}}, +^{\mathfrak{A}}(three^{\mathfrak{A}}, zero^{\mathfrak{A}})) \\
&= \times^{\mathfrak{A}}(two^{\mathfrak{A}}, +^{\mathfrak{A}}(three^{\mathfrak{A}}, zero^{\mathfrak{A}})) \\
&= \times^{\mathfrak{A}}(2, +^{\mathfrak{A}}(3, 0)) \\
&= \times^{\mathfrak{A}}(2, 3) \\
&= 6
\end{aligned}$$

1.11 Satisfaction of a Wff in a Structure

The basic notion that relates expressions such as terms and wffs, on the one hand, and structures on the other, are those of *value* of a term and *satisfaction* of a wff. Informally, the value of a term is an element of a structure—if the term is just a constant, its value is the object assigned to the constant by the structure, and if it is built up using function symbols, the value is computed from the values of constants and the functions assigned to the functions in the term. A wff is *satisfied* in a structure if the interpretation given to the predicates makes the wff true in the domain of the structure. This notion of satisfaction is specified inductively: the specification of the structure directly states when atomic wffs are satisfied, and we define when a complex wff is satisfied depending on the main connective or quantifier and whether or not the immediate subformulas are satisfied. The case of the quantifiers here is a bit tricky, as the immediate subformula of a quantified wff has a free variable, and structures don't specify the values of variables. In order to deal with this difficulty, we also introduce *variable assignments* and define satisfaction not with respect to a structure alone, but with respect to a structure plus a variable assignment.

Definition 111A (Variable Assignment) A *variable assignment* s for a structure $\mathfrak{A}$ is a function which maps each variable to an element of $|\mathfrak{A}|$, i.e., $s\colon \mathrm{Var} \to |\mathfrak{A}|$.

A structure assigns a value to each constant symbol, and a variable assignment to each variable. But we want to use terms built up from them to also name elements of the domain. For this we define the value of terms inductively. For constant symbols and variables the value is just as the structure or the variable assignment specifies it; for more complex terms it is computed recursively using the functions the structure assigns to the function symbols.

Definition 111B (Value of Terms) If t is a term of the language $\mathcal{L}$, $\mathfrak{A}$ is a structure for $\mathcal{L}$, and s is a variable assignment for $\mathfrak{A}$, the *value* $\bar{s}(t)$ is defined as follows:

1. $t \equiv c$: $\bar{s}(t) = c^{\mathfrak{A}}$.
2. $t \equiv x$: $\bar{s}(t) = s(x)$.
3. $t \equiv f t_1 \dots t_n$:
$$\bar{s}(t) = f^{\mathfrak{A}}(\bar{s}(t_1), \dots, \bar{s}(t_n)).$$

Definition 111C (x-Variant) If s is a variable assignment for a structure $\mathfrak{A}$, then any variable assignment s' for $\mathfrak{A}$ 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 $s \sim_x s'$.

Note that an x-variant of an assignment s does not *have* to assign something different to x. In fact, every assignment counts as an x-variant of itself.

DEFINITION 111D (SATISFACTION) Satisfaction of a wff α in a structure $\mathfrak{A}$ relative to a variable assignment s, in symbols: $\models_{\mathfrak{A}} \alpha[s]$, is defined recursively as follows. (We write $\not\models_{\mathfrak{A}} \alpha[s]$ to mean "not $\models_{\mathfrak{A}} \alpha[s]$.")

1. $\alpha \equiv Rt_1 \dots t_n$: $\models_{\mathfrak{A}} \alpha[s]$ iff $\langle \bar{s}(t_1), \dots, \bar{s}(t_n) \rangle \in R^{\mathfrak{A}}$.
2. $\alpha \equiv t_1 = t_2$: $\models_{\mathfrak{A}} \alpha[s]$ iff $\bar{s}(t_1) = \bar{s}(t_2)$.
3. $\alpha \equiv \neg\beta$: $\models_{\mathfrak{A}} \alpha[s]$ iff $\not\models_{\mathfrak{A}} \beta[s]$.
4. $\alpha \equiv (\beta \to \gamma)$: $\models_{\mathfrak{A}} \alpha[s]$ iff $\not\models_{\mathfrak{A}} \beta[s]$ or $\models_{\mathfrak{A}} \gamma[s]$ (or both).
5. $\alpha \equiv \forall x\, \beta$: $\models_{\mathfrak{A}} \alpha[s]$ iff for every x-variant s' of s, $\models_{\mathfrak{A}} \beta[s']$.

The variable assignments are important in the last clause. We cannot define satisfaction of $\forall x\, \beta(x)$ by "for all $a \in |\mathfrak{A}|$, $\models_{\mathfrak{A}} \beta(a)$." The reason is that a is not symbol of the language, and so $\beta(a)$ is not a wff (that is, $\beta[a/x]$ is undefined). We also cannot assume that we have constant symbols or terms available that name every element of $\mathfrak{A}$, since there is nothing in the definition of structures that requires it. Even in the standard language the set of constant symbols is denumerable, so if $|\mathfrak{A}|$ is not enumerable there aren't even enough constant symbols to name every object.

Example 1.11.5. Let $=\{a, b, f, R\}$ where a and b are constant symbols, f is a two-place function symbol, and A is a two-place predicate symbol. Consider the structure $\mathfrak{A}$ defined by:

1. $|\mathfrak{A}| = \{1, 2, 3, 4\}$
2. $a^{\mathfrak{A}} = 1$
3. $b^{\mathfrak{A}} = 2$
4. $f^{\mathfrak{A}}(x, y) = x + y$ if $x + y \leq 3$ and $= 3$ otherwise.
5. $R^{\mathfrak{A}} = \{\langle 1, 1\rangle, \langle 1, 2\rangle, \langle 2, 3\rangle, \langle 2, 4\rangle\}$

The function $s(x) = 1$ that assigns $1 \in |\mathfrak{A}|$ to every variable is a variable assignment for $\mathfrak{A}$.

Then

$$\bar{s}(f(a, b)) = f^{\mathfrak{A}}(\bar{s}(a), \bar{s}(b)).$$

Since a and b are constant symbols, $\bar{s}(a) = a^{\mathfrak{A}} = 1$ and $\bar{s}(b) = b^{\mathfrak{A}} = 2$. So

$$\bar{s}(f(a, b)) = f^{\mathfrak{A}}(1, 2) = 1 + 2 = 3.$$

To compute the value of $f(f(a, b), a)$ we have to consider

$$\bar{s}(f(f(a, b), a)) = f^{\mathfrak{A}}(\bar{s}(f(a, b)), \bar{s}(a)) = f^{\mathfrak{A}}(3, 1) = 3,$$

since $3 + 1 > 3$. Since $s(x) = 1$ and $\bar{s}(x) = s(x)$, we also have

$$\bar{s}(f(f(a,b),x)) = f^{\mathfrak{A}}(\bar{s}(f(a,b)), \bar{s}(x)) = f^{\mathfrak{A}}(3,1) = 3,$$

An atomic wff $R(t_1, t_2)$ is satisfied if the tuple of values of its arguments, i.e., $\langle \bar{s}(t_1), \bar{s}(t_2) \rangle$, is an element of $R^{\mathfrak{A}}$. So, e.g., we have $\models_{\mathfrak{A}} R(b, f(a,b))[s]$ since $\langle b^{\mathfrak{A}}, f(a,b)^{\mathfrak{A}} \rangle = \langle 2, 3 \rangle \in R^{\mathfrak{A}}$, but $\not\models_{\mathfrak{A}} R(x, f(a,b))$ since $\langle 1, 3 \rangle \notin R^{\mathfrak{A}}[s]$.

To determine if a non-atomic formula α is satisfied, you apply the clauses in the inductive definition that applies to the main connective. For instance, the main connective in $R(a,a) \to (R(b,x) \vee R(x,b)$ is the $\to$, and

$$\begin{aligned} &\models_{\mathfrak{A}} R(a,a) \to (R(b,x) \vee R(x,b))[s] \text{ iff} \\ &\quad \not\models_{\mathfrak{A}} R(a,a)[s] \text{ or } \models_{\mathfrak{A}} R(b,x) \vee R(x,b)[s] \end{aligned}$$

Since $\models_{\mathfrak{A}} R(a,a)[s]$ (because $\langle 1, 1 \rangle \in R^{\mathfrak{A}}$) we can't yet determine the answer and must first figure out if $\models_{\mathfrak{A}} R(b,x) \vee R(x,b)[s]$:

$$\begin{aligned} &\models_{\mathfrak{A}} R(b,x) \vee R(x,b)[s] \text{ iff} \\ &\quad \models_{\mathfrak{A}} R(b,x)[s] \text{ or } \models_{\mathfrak{A}} R(x,b)[s] \end{aligned}$$

And this is the case, since $\models_{\mathfrak{A}} R(x,b)[s]$ (because $\langle 1, 2 \rangle \in R^{\mathfrak{A}}$).

Recall that an x-variant of s is a variable assignment that differs from s at most in what it assigns to x. For every element of $|\mathfrak{A}|$, there is an x-variant of s: $s_1(x) = 1$, $s_2(x) = 2$, $s_3(x) = 3$, $s_4(s) = 4$, and with $s_i(y) = s(y) = 1$ for all variables y other than x are all the x-variants of s for the structure $\mathfrak{A}$. Note, in particular, that $s_1 = s$ is also an x-variant of s, i.e., s is an x-variant of itself.

To determine if a universally quantified wff $\forall x\, \alpha(x)$ is satisfied, we have to determine if $\models_{\mathfrak{A}} \alpha(x)[s']$ for all x-variants s' of s. So,

$$\models_{\mathfrak{A}} \forall x\, (R(x,a) \to R(a,x))[s],$$

since $\models_{\mathfrak{A}} R(x,a) \to R(a,x)[s_i]$ for all s_i ($\models_{\mathfrak{A}} R(a,x)[s_1]$ and $\not\models_{\mathfrak{A}} R(a,x)[s_j]$ for $j = 2$, 3, and 4). But,

$$\not\models_{\mathfrak{A}} \forall x\, (R(a,x) \to R(x,a))[s]$$

since $\not\models_{\mathfrak{A}} R(a,x) \to R(x,a)[s_2]$ (because $\models_{\mathfrak{A}} R(a,x)[s_2]$ and $\not\models_{\mathfrak{A}} R(x,a)[s_2]$).

To determine if an existentially quantified wff $\exists x\, \alpha(x)$ is satisfied, we have to determine if $\models_{\mathfrak{A}} \neg\forall x\, \neg\alpha(x)[s]$. For instance, we have

$$\models_{\mathfrak{A}} \exists x\, (R(b,x) \vee R(x,b))[s].$$

First, $\models_{\mathfrak{A}} R(b,x) \vee R(x,b)[s_1]$ (s_3 would also fit the bill). So, $\not\models_{\mathfrak{A}} \neg(R(b,x) \vee R(x,b))[s_1]$, thus $\not\models_{\mathfrak{A}} \forall x \neg((R(b,x) \vee R(x,b))[s]$, and therefore $\models_{\mathfrak{A}} \neg\forall x \neg((R(b,x) \vee R(x,b))[s]$. On the other hand,

$$\not\models_{\mathfrak{A}} \exists x\, (R(b,x) \wedge R(x,b)), [s]$$

since for none of the s_i, $\models_{\mathfrak{A}} R(b,x) \wedge R(x,b)[s_i]$, $\models_{\mathfrak{A}} \forall x \neg(R(b,x) \wedge R(x,b))[s]$. As you can probably guess from these examples, $\models_{\mathfrak{A}} \exists x\, \alpha(x)[s]$ iff $\models_{\mathfrak{A}} \alpha(x)[s']$ for at least one x-variant s' of s.

For a more complicated case, consider

$$\forall x\, (R(a,x) \to \exists y\, R(x,y)).$$

Since $\not\models_{\mathfrak{A}} R(a,x)[s_3]$ and $\not\models_{\mathfrak{A}} R(a,x)[s_4]$, the interesting cases where we have to worry about the consequent of the conditional are only s_1 and s_2. Does $\models_{\mathfrak{A}} \exists y\, R(x,y)[s_1]$ hold? It does if there is at least one y-variant s_1' of s_1 so that $\models_{\mathfrak{A}} R(x,y)[s_1']$. In fact, s_1 is such a y-variant ($s_1(x) = 1$, $s_1(y) = 1$, and $\langle 1,1\rangle \in R^{\mathfrak{A}}$), so the answer is yes. To determine if $\models_{\mathfrak{A}} \exists y\, R(x,y)[s_2]$ we have to look at the y-variants of s_2. Here, s_2 itself does not satisfy $R(x,y)$ ($s_2(x) = 2$, $s_2(y) = 1$, and $\langle 2,1\rangle \notin R^{\mathfrak{A}}$). However, consider $s_2' \sim_y s_2$ with $s_2'(y) = 3$. $\models_{\mathfrak{A}} R(x,y)[s_2']$ since $\langle 2,3\rangle \in R^{\mathfrak{A}}$, and so $\models_{\mathfrak{A}} \exists y\, R(x,y)[s_2]$. In sum, for every x-variant s_i of s, either $\not\models_{\mathfrak{A}} R(a,x)[s_i]$ ($i = 3, 4$) or $\models_{\mathfrak{A}} \exists y\, R(x,y)[s_i]$ ($i = 1, 2$), and so

$$\models_{\mathfrak{A}} \forall x\, (R(a,x) \to \exists y\, R(x,y))[s].$$

On the other hand,

$$\not\models_{\mathfrak{A}} \exists x\, (R(a,x) \wedge \forall y\, R(x,y))[s].$$

The only x-variants s_i of s with $\models_{\mathfrak{A}} R(a,x)[s_i]$ are s_1 and s_2. But for each, there is in turn a y-variant $s_i' \sim_y s_i$ with $s_i'(y) = 4$ so that $\not\models_{\mathfrak{A}} R(x,y)[s_i']$ and so $\not\models_{\mathfrak{A}} \forall y\, R(x,y)[s_i]$ for $i = 1, 2$. In sum, none of the x-variants $s_i \sim_x s$ are such that $\models_{\mathfrak{A}} R(a,x) \wedge \forall y\, R(x,y)[s_i]$.

PROPOSITION 111F $\models_{\mathfrak{A}} \exists x\, \beta(x)[s]$ iff there is an x-variant s' of s so that $\models_{\mathfrak{A}} \beta(x)[s']$.

Proof. Exercise. □

1.12 Variable Assignments

A variable assignment s provides a value for *every* variable—and there are infinitely many of them. This is of course not necessary. We require variable assignments to assign values to all variables simply because it makes things a lot easier. The value of a term t, and whether or not a wff α is satisfied in a structure with respect to s, only depend on the assignments s makes to the variables in t and the free variables of α. This is the content of the next two propositions. To make the idea of "depends on" precise, we show that any two

variable assignments that agree on all the variables in t give the same value, and that α is satisfied relative to one iff it is satisfied relative to the other if two variable assignments agree on all free variables of α.

PROPOSITION 112A If the variables in a term t are among $x_1, \ldots, x_n$, and $s_1(x_i) = s_2(x_i)$ for $i = 1, \ldots, n$, then $\bar{s_1}(t) = \bar{s_2}(t)$.

Proof. By induction on the complexity of t. For the base case, t can be a constant symbol or one one of the variables $x_1, \ldots, x_n$. If $t = c$, then $\bar{s_1}(t) = c^{\mathfrak{A}} = \bar{s_2}(t)$. If $t = x_i$, $s_1(x_i) = s_2(x_i)$ by the hypothesis of the proposition, and so $\bar{s_1}(t) = s_1(x_i) = s_2(x_i) = \bar{s_2}(t)$.

For the inductive step, assume that $t = ft_1 \ldots t_k$ and that the claim holds for $t_1, \ldots, t_k$. Then

$$\begin{aligned}\bar{s_1}(t) &= \bar{s_1}(ft_1 \ldots t_k) = \\ &= f^{\mathfrak{A}}(\bar{s_1}(t_1), \ldots, \bar{s_1}(t_k))\end{aligned}$$

For $j = 1, \ldots, k$, the variables of t_j are among $x_1, \ldots, x_n$. So by induction hypothesis, $\bar{s_1}(t_j) = \bar{s_2}(t_j)$. So,

$$\begin{aligned}\bar{s_1}(t) &= \bar{s_2}(ft_1 \ldots t_k) = \\ &= f^{\mathfrak{A}}(\bar{s_1}(t_1), \ldots, \bar{s_1}(t_k)) = \\ &= f^{\mathfrak{A}}(\bar{s_2}(t_1), \ldots, \bar{s_2}(t_k)) = \\ &= \bar{s_2}(ft_1 \ldots t_k) = \bar{s_2}(t).\end{aligned}$$

□

PROPOSITION 112B If the free variables in α are among $x_1, \ldots, x_n$, and $s_1(x_i) = s_2(x_i)$ for $i = 1, \ldots, n$, then $\models_{\mathfrak{A}} \alpha[s_1]$ iff $\models_{\mathfrak{A}} \alpha[s_2]$.

Proof. We use induction on the complexity of α. For the base case, where α is atomic, α can be: $Rt_1 \ldots t_k$ for a k-place predicate R and terms $t_1, \ldots, t_k$, or $t_1 = t_2$ for terms t_1 and t_2.

1. $\alpha \equiv Rt_1 \ldots t_k$: let $\models_{\mathfrak{A}} \alpha[s_1]$. Then

   $$\langle \bar{s_1}(t_1), \ldots, \bar{s_1}(t_k) \rangle \in R^{\mathfrak{A}}.$$

   For $i = 1, \ldots, k$, $\bar{s_1}(t_i) = \bar{s_2}(t_i)$ by Proposition 112A. So we also have $\langle \bar{s_2}(t_i), \ldots, \bar{s_2}(t_k) \rangle \in R^{\mathfrak{A}}$.

2. $\alpha \equiv t_1 = t_2$: suppose $\models_{\mathfrak{A}} \alpha[s_1]$. Then $\bar{s_1}(t_1) = \bar{s_1}(t_2)$. So,

   $$\begin{aligned}\bar{s_2}(t_1) &= \bar{s_1}(t_1) && \text{(by Proposition 112A)} \\ &= \bar{s_1}(t_2) && \text{(since } \models_{\mathfrak{A}} t_1 = t_2[s_1]) \\ &= \bar{s_2}(t_2) && \text{(by Proposition 112A),}\end{aligned}$$

   so $\models_{\mathfrak{A}} t_1 = t_2[s_2]$.

Now assume $\models_{\mathfrak{A}} \beta[s_1]$ iff $\models_{\mathfrak{A}} \beta[s_2]$ for all wffs β less complex than α. The induction step proceeds by cases determined by the main operator of α. In each case, we only demonstrate the forward direction of the biconditional; the proof of the reverse direction is symmetrical. In all cases except those for the quantifiers, we apply the induction hypothesis to sub-wffs β of α. The free variables of β are among those of α. Thus, if s_1 and s_2 agree on the free variables of α, they also agree on those of β, and the induction hypothesis applies to β.

1. $\alpha \equiv \neg\beta$: if $\models_{\mathfrak{A}} \alpha[s_1]$, then $\not\models_{\mathfrak{A}} \beta[s_1]$, so by the induction hypothesis, $\not\models_{\mathfrak{A}} \beta[s_2]$, hence $\models_{\mathfrak{A}} \alpha[s_2]$.

2. $\alpha \equiv \beta \to \gamma$: if $\models_{\mathfrak{A}} \alpha[s_1]$, then $\not\models_{\mathfrak{A}} \beta[s_1]$ or $\models_{\mathfrak{A}} \gamma[s_1]$. By the induction hypothesis, $\not\models_{\mathfrak{A}} \beta[s_2]$ or $\models_{\mathfrak{A}} \gamma[s_2]$, so $\models_{\mathfrak{A}} \alpha[s_2]$.

3. $\alpha \equiv \forall x\, \beta$: if $\models_{\mathfrak{A}} \alpha[s_1]$, then for every x-variant s_1' of s_1, $\models_{\mathfrak{A}} \beta[s_1']$. Take an arbitrary x-variant s_2' of s_2, let s_1' be the x-variant of s_1 which assigns the same thing to x as does s_2'. The free variables of β are among $x_1, \ldots, x_n$, and x. $s_1'(x_i) = s_2'(x_i)$, since s_1' and s_2' are x-variants of s_1 and s_2, respectively, and by hypothesis $s_1(x_i) = s_2(x_i)$. $s_1'(x) = s_2'(x)$ by the way we have defined s_1'. Then the induction hypothesis applies to β and s_1', s_2, and we have $\models_{\mathfrak{A}} \beta[s_2']$. Since s_2' is an arbitrary x-variant of s_2, every x-variant of s_2 satisfies β, and so $\models_{\mathfrak{A}} \alpha[s_2]$.

By induction, we get that $\models_{\mathfrak{A}} \alpha[s_1]$ iff $\models_{\mathfrak{A}} \alpha[s_2]$ whenever the free variables in α are among $x_1, \ldots, x_n$ and $s_1(x_i) = s_2(x_i)$ for $i = 1, \ldots, n$. □

Sentences have no free variables, so any two variable assignments assign the same things to all the (zero) free variables of any sentence. The proposition just proved then means that whether or not a sentence is satisfied in a structure relative to a variable assignment is completely independent of the assignment. We'll record this fact. It justifies the definition of satisfaction of a sentence in a structure (without mentioning a variable assignment) that follows.

COROLLARY 1.12.3 If α is a sentence and s a variable assignment, then $\models_{\mathfrak{A}} \alpha[s]$ iff $\models_{\mathfrak{A}} \alpha[s']$ for every variable assignment s'.

Proof. Let s' be any variable assignment. Since α is a sentence, it has no free variables, and so every variable assignment s' trivially assigns the same things to all free variables of α as does s. So the condition of Proposition 112B is satisfied, and we have $\models_{\mathfrak{A}} \alpha[s]$ iff $\models_{\mathfrak{A}} \alpha[s']$. □

DEFINITION 112D If α is a sentence, we say that a structure $\mathfrak{A}$ *satisfies* α, $\models_{\mathfrak{A}} \alpha$, iff $\models_{\mathfrak{A}} \alpha[s]$ for all variable assignments s.

If $\models_{\mathfrak{A}} \alpha$, we also simply say that α *is true in* $\mathfrak{A}$.

PROPOSITION 112E Let $\mathfrak{A}$ be a structure, α be a sentence, and s a variable assignment. $\models_{\mathfrak{A}} \alpha$ iff $\models_{\mathfrak{A}} \alpha[s]$.

Proof. Exercise. □

PROPOSITION 112F Suppose $\alpha(x)$ only contains x free, and $\mathfrak{A}$ is a structure. Then: $\models_{\mathfrak{A}} \forall x\, \alpha(x)$ iff $\models_{\mathfrak{A}} \alpha(x)[s]$ for all variable assignments s.

Proof. Exercise. □

1.13 Extensionality

Extensionality, sometimes called relevance, can be expressed informally as follows: the only thing that bears upon the satisfaction of wff α in a structure $\mathfrak{A}$ relative to a variable assignment s, are the assignments made by $\mathfrak{A}$ and s to the elements of the language that actually appear in α.

One immediate consequence of extensionality is that where two structures $\mathfrak{A}$ and $\mathfrak{A}'$ agree on all the elements of the language appearing in a sentence α and have the same domain, $\mathfrak{A}$ and $\mathfrak{A}'$ must also agree on whether or not α itself is true.

PROPOSITION 113A (EXTENSIONALITY) Let α be a wff, and $\mathfrak{A}_1$ and $\mathfrak{A}_2$ be structures with $|\mathfrak{A}_1| = |\mathfrak{A}_2|$, and s a variable assignment on $|\mathfrak{A}_1| = |\mathfrak{A}_2|$. If $c^{\mathfrak{A}_1} = c^{\mathfrak{A}_2}$, $R^{\mathfrak{A}_1} = R^{\mathfrak{A}_2}$, and $f^{\mathfrak{A}_1} = f^{\mathfrak{A}_2}$ for every constant symbol c, relation symbol R, and function symbol f occurring in α, then $\models_{\mathfrak{A}_1} \alpha[s]$ iff $\models_{\mathfrak{A}_2} \alpha[s]$.

Proof. First prove (by induction on t) that for every term, $\bar{s}(t) = \bar{s}(t)$. Then prove the proposition by induction on α, making use of the claim just proved for the induction basis (where α is atomic). □

COROLLARY 1.13.2 (EXTENSIONALITY FOR SENTENCES) Let α be a sentence and $\mathfrak{M}_1$, $\mathfrak{M}_2$ as in Proposition 113A. Then $\models_{\mathfrak{A}_1} \alpha$ iff $\models_{\mathfrak{A}_2} \alpha$.

Proof. Follows from Proposition 113A by corollary 1.12.3. □

Moreover, the value of a term, and whether or not a structure satisfies a wff, only depends on the values of its subterms.

PROPOSITION 113C Let $\mathfrak{A}$ be a structure, t and t' terms, and s a variable assignment. Let $s' \sim_x s$ be the x-variant of s given by $s'(x) = \bar{s}(t')$. Then $\bar{s}(t[t'/x]) = \bar{s'}(t)$.

Proof. By induction on t.

1. If t is a constant, say, $t \equiv c$, then $t[t'/x] = c$, and $\bar{s}(c) = c^{\mathfrak{A}} = \bar{s'}(c)$.
2. If t is a variable other than x, say, $t \equiv y$, then $t[t'/x] = y$, and $\bar{s}(y) = \bar{s'}(y)$ since $s' \sim_x s$.
3. If $t \equiv x$, then $t[t'/x] = t'$. But $\bar{s'}(x) = \bar{s}(t')$ by definition of s'.

4. If $t \equiv ft_1 \dots t_n$ then we have:

$$\begin{aligned}
\bar{s}(t[t'/x]) = & \\
& = \bar{s}(ft_1[t'/x] \dots t_n[t'/x]) \\
& \quad \text{by definition of } t[t'/x] \\
& = f^{\mathfrak{A}}(\bar{s}(t_1[t'/x]), \dots, \bar{s}(t_n[t'/x])) \\
& \quad \text{by definition of } \bar{s}(f \dots) \\
& = f^{\mathfrak{A}}(\bar{s'}(t_1), \dots, \bar{s'}(t_n)) \\
& \quad \text{by induction hypothesis} \\
& = \bar{s'}(t) \text{ by definition of } \bar{s'}(f \dots)
\end{aligned}$$

□

PROPOSITION 113D Let $\mathfrak{A}$ be a structure, α a wff, t a term, and s a variable assignment. Let $s' \sim_x s$ be the x-variant of s given by $s'(x) = \bar{s}(t)$. Then $\models_{\mathfrak{A}} \alpha[t/x][s]$ iff $\models_{\mathfrak{A}} \alpha[s']$.

Proof. Exercise. □

1.14 Semantic Notions

Give the definition of structures for first-order languages, we can define some basic semantic properties of and relationships between sentences. The simplest of these is the notion of *validity* of a sentence. A sentence is valid if it is satisfied in every structure. Valid sentences are those that are satisfied regardless of how the non-logical symbols in it are interpreted. Valid sentences are therefore also called *logical truths*—they are true, i.e., satisfied, in any structure and hence their truth depends only on the logical symbols occurring in them and their syntactic structure, but not on the non-logical symbols or their interpretation.

DEFINITION 114A (VALIDITY) A sentence α is *valid*, $\vDash \alpha$, iff $\models_{\mathfrak{A}} \alpha$ for every structure $\mathfrak{A}$.

DEFINITION 114B (ENTAILMENT) A set of sentences Γ *entails* a sentence α, $\Gamma \vDash \alpha$, iff for every structure $\mathfrak{A}$ with $\models_{\mathfrak{A}} \Gamma$, $\models_{\mathfrak{A}} \alpha$.

DEFINITION 114C (SATISFIABILITY) A set of sentences Γ is *satisfiable* if $\models_{\mathfrak{A}} \Gamma$ for some structure $\mathfrak{A}$. If Γ is not satisfiable it is called *unsatisfiable*.

PROPOSITION 114D A sentence α is valid iff $\Gamma \vDash \alpha$ for every set of sentences Γ.

Proof. For the forward direction, let α be valid, and let Γ be a set of sentences. Let $\mathfrak{A}$ be a structure so that $\models_{\mathfrak{A}} \Gamma$. Since α is valid, $\models_{\mathfrak{A}} \alpha$, hence $\Gamma \vDash \alpha$.

For the contrapositive of the reverse direction, let α be invalid, so there is a structure $\mathfrak{A}$ with $\not\models_{\mathfrak{A}} \alpha$. When $\Gamma = \{\top\}$, since $\top$ is valid, $\models_{\mathfrak{A}} \Gamma$. Hence, there is a structure $\mathfrak{A}$ so that $\models_{\mathfrak{A}} \Gamma$ but $\not\models_{\mathfrak{A}} \alpha$, hence Γ does not entail α. □

PROPOSITION 114E $\Gamma \vDash \alpha$ iff $\Gamma \cup \{\neg\alpha\}$ is unsatisfiable.

Proof. For the forward direction, suppose $\Gamma \vDash \alpha$ and suppose to the contrary that there is a structure $\mathfrak{A}$ so that $\models_{\mathfrak{A}} \Gamma \cup \{\neg\alpha\}$. Since $\models_{\mathfrak{A}} \Gamma$ and $\Gamma \vDash \alpha$, $\models_{\mathfrak{A}} \alpha$. Also, since $\models_{\mathfrak{A}} \Gamma \cup \{\neg\alpha\}$, $\models_{\mathfrak{A}} \neg\alpha$, so we have both $\models_{\mathfrak{A}} \alpha$ and $\not\models_{\mathfrak{A}} \alpha$, a contradiction. Hence, there can be no such structure $\mathfrak{A}$, so $\Gamma \cup \{\alpha\}$ is unsatisfiable.

For the reverse direction, suppose $\Gamma \cup \{\neg\alpha\}$ is unsatisfiable. So for every structure $\mathfrak{A}$, either $\not\models_{\mathfrak{A}} \Gamma$ or $\models_{\mathfrak{A}} \alpha$. Hence, for every structure $\mathfrak{A}$ with $\models_{\mathfrak{A}} \Gamma$, $\models_{\mathfrak{A}} \alpha$, so $\Gamma \vDash \alpha$. □

Proposition 114F If $\Gamma \subseteq \Gamma'$ and $\Gamma \vDash \alpha$, then $\Gamma' \vDash \alpha$.

Proof. Suppose that $\Gamma \subseteq \Gamma'$ and $\Gamma \vDash \alpha$. Let $\mathfrak{A}$ be such that $\models_{\mathfrak{A}} \Gamma'$; then $\models_{\mathfrak{A}} \Gamma$, and since $\Gamma \vDash \alpha$, we get that $\models_{\mathfrak{A}} \alpha$. Hence, whenever $\models_{\mathfrak{A}} \Gamma'$, $\models_{\mathfrak{A}} \alpha$, so $\Gamma' \vDash \alpha$. □

Theorem 114G (Semantic Deduction Theorem) $\Gamma \cup \{\alpha\} \vDash \beta$ iff $\Gamma \vDash \alpha \to \beta$.

Proof. For the forward direction, let $\Gamma \cup \{\alpha\} \vDash \beta$ and let $\mathfrak{A}$ be a structure so that $\models_{\mathfrak{A}} \Gamma$. If $\models_{\mathfrak{A}} \alpha$, then $\models_{\mathfrak{A}} \Gamma \cup \{\alpha\}$, so since $\Gamma \cup \{\alpha\}$ entails β, we get $\models_{\mathfrak{A}} \beta$. Therefore, $\models_{\mathfrak{A}} \alpha \to \beta$, so $\Gamma \vDash \alpha \to \beta$.

For the reverse direction, let $\Gamma \vDash \alpha \to \beta$ and $\mathfrak{A}$ be a structure so that $\models_{\mathfrak{A}} \Gamma \cup \{\alpha\}$. Then $\models_{\mathfrak{A}} \Gamma$, so $\models_{\mathfrak{A}} \alpha \to \beta$, and since $\models_{\mathfrak{A}} \alpha$, $\models_{\mathfrak{A}} \beta$. Hence, whenever $\models_{\mathfrak{A}} \Gamma \cup \{\alpha\}$, $\models_{\mathfrak{A}} \beta$, so $\Gamma \cup \{\alpha\} \vDash \beta$. □

Proposition 114H Let $\mathfrak{A}$ be a structure, and $\alpha(x)$ a wff with one free variable x, and t a closed term. Then: $\forall x\, \alpha(x) \vDash \alpha(t)$

Proof. Suppose $\models_{\mathfrak{A}} \forall x\, \alpha(x)$. Let s be a variable assignment with $s(x) = t^{\mathfrak{A}}$. By Proposition 112F, $\models_{\mathfrak{A}} \alpha(x)[s]$. By Proposition 113D, $\models_{\mathfrak{A}} \alpha(t)[s]$. By Proposition 112E, $\models_{\mathfrak{A}} \alpha(t)$ since $\alpha(t)$ is a sentence. □

Chapter 2

Theories and Their Models

2.1 Introduction

The development of the axiomatic method is a significant achievement in the history of science, and is of special importance in the history of mathematics. An axiomatic development of a field involves the clarification of many questions: What is the field about? What are the most fundamental concepts? How are they related? Can all the concepts of the field be defined in terms of these fundamental concepts? What laws do, and must, these concepts obey?

The axiomatic method and logic were made for each other. Formal logic provides the tools for formulating axiomatic theories, for proving theorems from the axioms of the theory in a precisely specified way, for studying the properties of all systems satisfying the axioms in a systematic way.

DEFINITION 21A A set of sentences Γ is *closed* iff, whenever $\Gamma \vDash \alpha$ then $\alpha \in \Gamma$. The *closure* of a set of sentences Γ is $\{\alpha : \Gamma \vDash \alpha\}$.

We say that Γ is *axiomatized by* a set of sentences Δ if Γ is the closure of Δ

We can think of an axiomatic theory as the set of sentences that is axiomatized by its set of axioms Δ. In other words, when we have a first-order language which contains non-logical symbols for the primitives of the axiomatically developed science we wish to study, together with a set of sentences that express the fundamental laws of the science, we can think of the theory as represented by all the sentences in this language that are entailed by the axioms. This ranges from simple examples with only a single primitive and simple axioms, such as the theory of partial orders, to complex theories such as Newtonian mechanics.

The important logical facts that make this formal approach to the axiomatic method so important are the following. Suppose Γ is an axiom system for a theory, i.e., a set of sentences.

1. We can state precisely when an axiom system captures an intended class of structures. That is, if we are interested in a certain class of struc-

tures, we will successfully capture that class by an axiom system Γ iff the structures are exactly those $\mathfrak{A}$ such that $\models_{\mathfrak{A}} \Gamma$.

2. We may fail in this respect because there are $\mathfrak{A}$ such that $\models_{\mathfrak{A}} \Gamma$, but $\mathfrak{A}$ is not one of the structures we intend. This may lead us to add axioms which are not true in $\mathfrak{A}$.

3. If we are successful at least in the respect that Γ is true in all the intended structures, then a sentence α is true in all intended structures whenever $\Gamma \vDash \alpha$. Thus we can use logical tools (such as proof methods) to show that sentences are true in all intended structures simply by showing that they are entailed by the axioms.

4. Sometimes we don't have intended structures in mind, but instead start from the axioms themselves: we begin with some primitives that we want to satisfy certain laws which we codify in an axiom system. One thing that we would like to verify right away is that the axioms do not contradict each other: if they do, there can be no concepts that obey these laws, and we have tried to set up an incoherent theory. We can verify that this doesn't happen by finding a model of Γ. And if there are models of our theory, we can use logical methods to investigate them, and we can also use logical methods to construct models.

5. The independence of the axioms is likewise an important question. It may happen that one of the axioms is actually a consequence of the others, and so is redundant. We can prove that an axiom α in Γ is redundant by proving $\Gamma \setminus \{\alpha\} \vDash \alpha$. We can also prove that an axiom is not redundant by showing that $(\Gamma \setminus \{\alpha\}) \cup \{\neg\alpha\}$ is satisfiable. For instance, this is how it was shown that the parallel postulate is independent of the other axioms of geometry.

6. Another important question is that of definability of concepts in a theory: The choice of the language determines what the models of a theory consists of. But not every aspect of a theory must be represented separately in its models. For instance, every ordering $\leq$ determines a corresponding strict ordering $<$—given one, we can define the other. So it is not necessary that a model of a theory involving such an order must *also* contain the corresponding strict ordering. When is it the case, in general, that one relation can be defined in terms of others? When is it impossible to define a relation in terms of other (and hence must add it to the primitives of the language)?

2.2 Expressing Properties of Structures

It is often useful and important to express conditions on functions and relations, or more generally, that the functions and relations in a structure satisfy these conditions. For instance, we would like to have ways of distinguishing those

structures for a language which "capture" what we want the predicate symbols to "mean" from those that do not. Of course we're completely free to specify which structures we "intend," e.g., we can specify that the interpretation of the predicate symbol $\leq$ must be an ordering, or that we are only interested in interpretations of $\mathcal{L}$ in which the domain consists of sets and $\in$ is interpreted by the "is an element of" relation. But can we do this with sentences of the language? In other words, which conditions on a structure $\mathfrak{A}$ can we express by a sentence (or perhaps a set of sentences) in the language of $\mathfrak{A}$? There are some conditions that we will not be able to express. For instance, there is no sentence of $\mathcal{L}_A$ which is only true in a structure $\mathfrak{A}$ if $|\mathfrak{A}| = \mathbb{N}$. We cannot express "the domain contains only natural numbers." But there are "structural properties" of structures that we perhaps can express. Which properties of structures can we express by sentences? Or, to put it another way, which collections of structures can we describe as those making a sentence (or set of sentences) true?

Definition 22A (Model of a set) Let Γ be a set of sentences in a language $\mathcal{L}$. We say that a structure $\mathfrak{A}$ *is a model of* Γ if $\models_{\mathfrak{A}} \alpha$ for all $\alpha \in \Gamma$.

Example 2.2.2. The sentence $\forall x\, x \leq x$ is true in $\mathfrak{A}$ iff $\leq^{\mathfrak{A}}$ is a reflexive relation. The sentence $\forall x \forall y\, ((x \leq y \wedge y \leq x) \rightarrow x = y)$ is true in $\mathfrak{A}$ iff $\leq^{\mathfrak{A}}$ is anti-symmetric. The sentence $\forall x \forall y \forall z\, ((x \leq y \wedge y \leq z) \rightarrow x \leq z)$ is true in $\mathfrak{A}$ iff $\leq^{\mathfrak{A}}$ is transitive. Thus, the models of

$$\begin{aligned} \{ \quad & \forall x\, x \leq x, \\ & \forall x \forall y\, ((x \leq y \wedge y \leq x) \rightarrow x = y), \\ & \forall x \forall y \forall z\, ((x \leq y \wedge y \leq z) \rightarrow x \leq z) \quad \} \end{aligned}$$

are exactly those structures in which $\leq^{\mathfrak{A}}$ is reflexive, anti-symmetric, and transitive, i.e., a partial order. Hence, we can take them as axioms for the *first-order theory of partial orders*.

2.3 Examples of First-Order Theories

Example 2.3.1. The theory of strict linear orders in the language $\mathcal{L}_<$ is axiomatized by the set

$$\begin{aligned} & \forall x \neg x < x, \\ & \forall x \forall y\, ((x < y \vee y < x) \vee x = y), \\ & \forall x \forall y \forall z\, ((x < y \wedge y < z) \rightarrow x < z) \end{aligned}$$

It completely captures the intended structures: every strict linear order is a model of this axiom system, and vice versa, if R is a linear order on a set X, then the structure $\mathfrak{A}$ with $|\mathfrak{A}| = X$ and $<^{\mathfrak{A}} = R$ is a model of this theory.

Example 2.3.2. The theory of groups in the language 1 (constant symbol), $\cdot$ (two-place function symbol) is axiomatized by

$$\begin{aligned}
&\forall x\,(x \cdot 1) = x\\
&\forall x\,\forall y\,\forall z\,(x \cdot (y \cdot z)) = ((x \cdot y) \cdot z)\\
&\forall x\,\exists y\,(x \cdot y) = 1
\end{aligned}$$

Example 2.3.3. The theory of Peano arithmetic is axiomatized by the following sentences in the language of arithmetic $\mathcal{L}_A$.

$$\begin{aligned}
&\neg\exists x\,x' = 0\\
&\forall x\,\forall y\,(x' = y' \to x = y)\\
&\forall x\,\forall y\,(x < y \leftrightarrow \exists z\,(x + z' = y))\\
&\forall x\,(x + 0) = x\\
&\forall x\,\forall y\,(x + y') = (x + y)'\\
&\forall x\,(x \times 0) = 0\\
&\forall x\,\forall y\,(x \times y') = ((x \times y) + x)
\end{aligned}$$

plus all sentences of the form

$$(\alpha(0) \wedge \forall x\,(\alpha(x) \to \alpha(x'))) \to \forall x\,\alpha(x)$$

Since there are infinitely many sentences of the latter form, this axiom system is infinite. The latter form is called the *induction schema*. (Actually, the induction schema is a bit more complicated than we let on here.)

The third axiom is an *explicit definition* of $<$.

Example 2.3.4. The theory of pure sets plays an important role in the foundations (and in the philosophy) of mathematics. A set is pure if all its elements are also pure sets. The empty set counts therefore as pure, but a set that has something as an element that is not a set would not be pure. So the pure sets are those that are formed just from the empty set and no "urelements," i.e., objects that are not themselves sets.

The following might be considered as an axiom system for a theory of pure sets:

$$\begin{aligned}
&\exists x\,\neg\exists y\,y \in x\\
&\forall x\,\forall y\,(\forall z(z \in x \leftrightarrow z \in y) \to x = y)\\
&\forall x\,\forall y\,\exists z\,\forall u\,(u \in z \leftrightarrow (u = x \vee u = y))\\
&\forall x\,\exists y\,\forall z\,(z \in y \leftrightarrow \exists u\,(z \in u \wedge u \in x))
\end{aligned}$$

plus all sentences of the form

$$\exists x\,\forall y\,(y \in x \leftrightarrow \alpha(y))$$

The first axiom says that there is a set with no elements (i.e., $\emptyset$ exists); the second says that sets are extensional; the third that for any sets X and Y, the set $\{X, Y\}$ exists; the fourth that for any sets X and Y, the set $X \cup Y$ exists.

The sentences mentioned last are collectively called the *naive comprehension scheme*. It essentially says that for every $\alpha(x)$, the set $\{x : \alpha(x)\}$ exists—so at first glance a true, useful, and perhaps even necessary axiom. It is called "naive" because, as it turns out, it makes this theory unsatisfiable: if you take $\alpha(y)$ to be $\neg y \in y$, you get the sentence

$$\exists x \, \forall y \, (y \in x \leftrightarrow \neg y \in y)$$

and this sentence is not satisfied in any structure.

Example 2.3.5. In the area of *mereology*, the relation of *parthood* is a fundamental relation. Just like theories of sets, there are theories of parthood that axiomatize various conceptions (sometimes conflicting) of this relation.

The language of mereology contains a single two-place predicate symbol P, and Pxy "means" that x is a part of y. When we have this interpretation in mind, a structure for this language is called a *parthood structure*. Of course, not every structure for a single two-place predicate will really deserve this name. To have a chance of capturing "parthood," $P^{\mathfrak{A}}$ must satisfy some conditions, which we can lay down as axioms for a theory of parthood. For instance, parthood is a partial order on objects: every object is a part (albeit an *improper* part) of itself; no two different objects can be parts of each other; a part of a part of an object is itself part of that object. Note that in this sense "is a part of" resembles "is a subset of," but does not resemble "is an element of" which is neither reflexive nor transitive.

$$\begin{aligned}
&\forall x \, Pxx, \\
&\forall x \, \forall y \, ((Pxy \wedge Pyx) \rightarrow x = y), \\
&\forall x \, \forall y \, \forall z \, ((Pxy \wedge Pyz) \rightarrow Pxz),
\end{aligned}$$

Moreover, any two objects have a mereological sum (an object that has these two objects as parts, and is minimal in this respect).

$$\forall x \, \forall y \, \exists z \, \forall u \, (Pzu \leftrightarrow (Pxu \wedge Pyu))$$

These are only some of the basic principles of parthood considered by metaphysicians. Further principles, however, quickly become hard to formulate or write down without first introducting some defined relations. For instance, most metaphysicians interested in mereology also view the following as a valid principle: whenever an object x has a proper part y, it also has a part z that has no parts in common with y, and so that the fusion of y and z is x.

2.4 Expressing Relations in a Structure

One main use wffs can be put to is to express properties and relations in a structure $\mathfrak{A}$ in terms of the primitives of the language $\mathcal{L}$ of $\mathfrak{A}$. By this we

mean the following: the domain of $\mathfrak{A}$ is a set of objects. The constant symbols, function symbols, and predicate symbols are interpreted in $\mathfrak{A}$ by some objects in$|\mathfrak{A}|$, functions on $|\mathfrak{A}|$, and relations on $|\mathfrak{A}|$. For instance, if A_0^2 is in $\mathcal{L}$, then $\mathfrak{A}$ assigns to it a relation $R = A_0^{2\mathfrak{A}}$. Then the formula $A_0^2 v_1 v_2$ *expresses* that very relation, in the following sense: if a variable assignment s maps v_1 to $a \in |\mathfrak{A}|$ and v_2 to $b \in |\mathfrak{A}|$, then

$$Rab \quad \text{iff} \quad \models_{\mathfrak{A}} A_0^2 v_1 v_2[s].$$

Note that we have to involve variable assignments here: we can't just say "Rab iff $\models_{\mathfrak{A}} A_0^2 ab$" because a and b are not symbols of our language: they are elements of $|\mathfrak{A}|$.

Since we don't just have atomic wffs, but can combine them using the logical connectives and the quantifiers, more complex wffs can define other relations which aren't directly built into $\mathfrak{A}$. We're interested in how to do that, and specifically, which relations we can define in a structure.

DEFINITION 24A Let $\alpha(v_1, \ldots, v_n)$ be a wff of $\mathcal{L}$ in which only $v_1, \ldots, v_n$ occur free, and let $\mathfrak{A}$ be a structure for $\mathcal{L}$. $\alpha(v_1, \ldots, v_n)$ *expresses the relation* $R \subseteq |\mathfrak{A}|^n$ iff

$$Ra_1 \ldots a_n \quad \text{iff} \quad \models_{\mathfrak{A}} \alpha v_1 \ldots v_n[s]$$

for any variable assignment s with $s(v_i) = a_i$ $(i = 1, \ldots, n)$.

Example 2.4.2. In the standard model of arithmetic $\mathfrak{B}$, the wff $v_1 < v_2 \vee v_1 = v_2$ expresses the $\leq$ relation on $\mathbb{N}$. The wff $v_2 = v_1'$ expresses the successor relation, i.e., the relation $R \subseteq \mathbb{N}^2$ where Rnm holds if m is the successor of n. The formula $v_1 = v_2'$ expresses the predecessor relation. The wffs $\exists v_3\, (v_3 \neq 0 \wedge v_2 = (v_1 + v_3))$ and $\exists v_3\, (v_1 + v_3') = v_2$ both express the $<$ relation. This means that the predicate symbol $<$ is actually superfluous in the language of arithmetic; it can be defined.

This idea is not just interesting in specific structures, but generally whenever we use a language to describe an intended model or models, i.e., when we consider theories. These theories often only contain a few predicate symbols as basic symbols, but in the domain they are used to describe often many other relations play an important role. If these other relations can be systematically expressed by the relations that interpret the basic predicate symbols of the language, we say we can *define* them in the language.

2.5 The Theory of Sets

Almost all of mathematics can be developed in the theory of sets. Developing mathematics in this theory involves a number of things. First, it requires a set of axioms for the relation $\in$. A number of different axiom systems have been developed, sometimes with conflicting properties of $\in$. The axiom system known as **ZFC**, Zermelo-Fraenkel set theory with the axiom of choice stands out: it is by far the most widely used and studied, because it turns out that its

axioms suffice to prove almost all the things mathematicians expect to be able to prove. But before that can be established, it first is necessary to make clear how we can even *express* all the things mathematicians would like to express. For starters, the language contains no constant symbols or function symbols, so it seems at first glance unclear that we can talk about particular sets (such as $\emptyset$ or $\mathbb{N}$), can talk about operations on sets (such as $X \cup Y$ and $\wp(X)$), let alone other constructions which involve things other than sets, such as relations and functions.

To begin with, "is an element of" is not the only relation we are interested in: "is a subset of" seems almost as important. But we can *define* "is a subset of" in terms of "is an element of." To do this, we have to find a wff $\alpha(x, y)$ in the language of set theory which is satisfied by a pair of sets $\langle X, Y\rangle$ iff $X \subseteq Y$. But X is a subset of Y just in case all elements of X are also elements of Y. So we can define $\subseteq$ by the formula

$$\forall z\,(z \in x \to z \in y)$$

Now, whenever we want to use the relation $\subseteq$ in a formula, we could instead use that formula (with x and y suitably replaced, and the bound variable z renamed if necessary). For instance, extensionality of sets means that if any sets x and y are contained in each other, then x and y must be the same set. This can be expressed by $\forall x\,\forall y\,((x \subseteq y \wedge y \subseteq x) \to x = y)$, or, if we replace $\subseteq$ by the above definition, by

$$\forall x\,\forall y\,((\forall z\,(z \in x \to z \in y) \wedge \forall z\,(z \in y \to z \in x)) \to x = y).$$

This is in fact one of the axioms of **ZFC**, the "axiom of extensionality."

There is no constant symbol for $\emptyset$, but we can express "x is empty" by $\neg\exists y\, y \in x$. Then "$\emptyset$ exists" becomes the sentence $\exists x\,\neg\exists y\, y \in x$. This is another axiom of **ZFC**. (Note that the axiom of extensionality implies that there is only one empty set.) Whenever we want to talk about $\emptyset$ in the language of set theory, we would write this as "there is a set that's empty and ..." As an example, to express the fact that $\emptyset$ is a subset of every set, we could write

$$\exists x\,(\neg\exists y\, y \in x \wedge \forall z\, x \subseteq z)$$

where, of course, $x \subseteq z$ would in turn have to be replaced by its definition.

To talk about operations on sets, such has $X \cup Y$ and $\wp(X)$, we have to use a similar trick. There are no function symbols in the language of set theory, but we can express the functional relations $X \cup Y = Z$ and $\wp(X) = Y$ by

$$\forall u\,((u \in x \vee u \in y) \leftrightarrow u \in z)$$
$$\forall u\,(u \subseteq x \leftrightarrow u \in y)$$

since the elements of $X \cup Y$ are exactly the sets that are either elements of X or elements of Y, and the elements of $\wp(X)$ are exactly the subsets of X. However, this doesn't allow us to use $x \cup y$ or $\wp(x)$ as if they were terms: we can only use

the entire wffs that define the relations $X \cup Y = Z$ and $\wp(X) = Y$. In fact, we do not know that these relations are ever satisfied, i.e., we do not know that unions and power sets always exist. For instance, the sentence $\forall x\, \exists y\, \wp(x) = y$ is another axiom of **ZFC** (the power set axiom).

Now what about talk of ordered pairs or functions? Here we have to explain how we can think of ordered pairs and functions as special kinds of sets. One way to define the ordered pair $\langle x, y\rangle$ is as the set $\{\{x\}, \{x, y\}\}$. But like before, we cannot introduce a function symbol that names this set; we can only define the relation $\langle x, y\rangle = z$, i.e., $\{\{x\}, \{x, y\}\} = z$:

$$\forall u\, (u \in z \leftrightarrow (\forall v\, (v \in u \leftrightarrow v = x) \vee \forall v\, (v \in u \leftrightarrow (v = x \vee v = y))))$$

This says that the elements u of z are exactly those sets which either have x as its only element or have x and y as its only elements (in other words, those sets that are either identical to $\{x\}$ or identical to $\{x, y\}$). Once we have this, we can say further things, e.g., that $X \times Y = Z$:

$$\forall z\, (z \in Z \leftrightarrow \exists x\, \exists y\, (x \in X \wedge y \in Y \wedge \langle x, y\rangle = z))$$

A function $f\colon X \to Y$ can be thought of as the relation $f(x) = y$, i.e., as the set of pairs $\{\langle x, y\rangle : f(x) = y\}$. We can then say that a set f is a function from X to Y if (a) it is a relation $\subseteq X \times Y$, (b) it is total, i.e., for all $x \in X$ there is some $y \in Y$ such that $\langle x, y\rangle \in f$ and (c) it is functional, i.e., whenever $\langle x, y\rangle, \langle x, y'\rangle \in f$, $y = y'$ (because values of functions must be unique). So "f is a function from X to Y" can be written as:

$$\begin{aligned}
&\forall u\, (u \in f \to \exists x\, \exists y\, (x \in X \wedge y \in Y \wedge \langle x, y\rangle = u)) \wedge \\
&\forall x\, (x \in X \to (\exists y\, (y \in Y \wedge \mathrm{maps}(f, x, y)) \wedge \\
&\qquad\qquad (\forall y\, \forall y'\, ((\mathrm{maps}(f, x, y) \wedge \mathrm{maps}(f, x, y')) \to y = y')))
\end{aligned}$$

where $\mathrm{maps}(f, x, y)$ abbreviates $\exists v\, (v \in f \wedge \langle x, y\rangle = v)$ (this wff expresses "$f(x) = y$").

It is now also not hard to express that $f\colon X \to Y$ is injective, for instance:

$$\begin{aligned}
f\colon X \to Y \wedge \forall x\, \forall x'\, ((x \in X \wedge x' \in X \wedge \\
\exists y\, (\mathrm{maps}(f, x, y) \wedge \mathrm{maps}(f, x', y))) \to x = x')
\end{aligned}$$

A function $f\colon X \to Y$ is injective iff, whenever f maps $x, x' \in X$ to a single y, $x = x'$. If we abbreviate this formula as $\mathrm{inj}(f, X, Y)$, we're already in a position to state in the language of set theory something as non-trivial as Cantor's theorem: there is no injective function from $\wp(X)$ to X:

$$\forall X\, \forall Y\, (\wp(X) = Y \to \neg \exists f\, \mathrm{inj}(f, Y, X))$$

One might think that set theory requires another axiom that guarantees the existence of a set for every defining property. If $\alpha(x)$ is a formula of set theory with the variable x free, we can consider the sentence

$$\exists y\, \forall x\, (x \in y \leftrightarrow \alpha(x)).$$

This sentence states that there is a set y whose elements are all and only those x that satisfy $\alpha(x)$. This schema is called the "comprehension principle." It looks very useful; unfortunately it is inconsistent. Take $\alpha(x) \equiv \neg x \in x$, then the comprehension principle states

$$\exists y \, \forall x \, (x \in y \leftrightarrow x \notin x),$$

i.e., it states the existence of a set of all sets that are not elements of themselves. No such set can exist—this is Russell's Paradox. **ZFC**, in fact, contains a restricted—and consistent—version of this principle, the separation principle:

$$\forall z \, \exists y \, \forall x \, (x \in y \leftrightarrow (x \in z \wedge \alpha(x)).$$

2.6 Expressing the Size of Structures

There are some properties of structures we can express even without using the non-logical symbols of a language. For instance, there are sentences which are true in a structure iff the domain of the structure has at least, at most, or exactly a certain number n of elements.

PROPOSITION 26A The sentence

$$\begin{aligned}\alpha_{\geq n} \equiv \exists x_1 \, \exists x_2 \, \dots \exists x_n \quad (x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_1 \neq x_4 \wedge \dots \wedge x_1 \neq x_n \wedge \\ x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge \dots \wedge x_2 \neq x_n \wedge \\ \vdots \\ x_{n-1} \neq x_n)\end{aligned}$$

is true in a structure $\mathfrak{A}$ iff $|\mathfrak{A}|$ contains at least n elements. Consequently, $\models_{\mathfrak{A}} \neg\alpha_{\geq n+1}$ iff $|\mathfrak{A}|$ contains at most n elements.

PROPOSITION 26B The sentence

$$\begin{aligned}\alpha_{=n} \equiv \exists x_1 \, \exists x_2 \, \dots \exists x_n \quad (x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_1 \neq x_4 \wedge \dots \wedge x_1 \neq x_n \wedge \\ x_2 \neq x_3 \wedge x_2 \neq x_4 \wedge \dots \wedge x_2 \neq x_n \wedge \\ \vdots \\ x_{n-1} \neq x_n \wedge \\ \forall y \, (y = x_1 \vee \dots y = x_n) \dots))\end{aligned}$$

is true in a structure $\mathfrak{A}$ iff $|\mathfrak{A}|$ contains exactly n elements.

PROPOSITION 26C A structure is infinite iff it is a model of

$$\{\alpha_{\geq 1}, \alpha_{\geq 2}, \alpha_{\geq 3}, \dots\}$$

There is no single purely logical sentence which is true in $\mathfrak{A}$ iff $|\mathfrak{A}|$ is infinite. However, one can give sentences with non-logical predicate symbols which only have infinite models (although not every infinite structure is a model of them). The property of being a finite structure, and the property of being a non-enumerable structure cannot even be expressed with an infinite set of sentences. These facts follow from the compactness and Löwenheim-Skolem theorems.

Chapter 3

Derivation Systems

3.1 Introduction

Logics commonly have both a semantics and a derivation system. The semantics concerns concepts such as truth, satisfiability, validity, and entailment. The purpose of derivation systems is to provide a purely syntactic method of establishing entailment and validity. They are purely syntactic in the sense that a derivation in such a system is a finite syntactic object, usually a sequence (or other finite arrangement) of sentences or wffs. Good derivation systems have the property that any given sequence or arrangement of sentences or wffs can be verified mechanically to be "correct."

The simplest (and historically first) derivation systems for first-order logic were *axiomatic*. A sequence of wffs counts as a derivation in such a system if each individual wff in it is either among a fixed set of "axioms" or follows from wffs coming before it in the sequence by one of a fixed number of "inference rules"—and it can be mechanically verified if a wff is an axiom and whether it follows correctly from other wffs by one of the inference rules. Axiomatic proof systems are easy to describe—and also easy to handle meta-theoretically—but derivations in them are hard to read and understand, and are also hard to produce.

Other derivation systems have been developed with the aim of making it easier to construct derivations or easier to understand derivations once they are complete. Examples are natural deduction, truth trees, also known as tableaux proofs, and the sequent calculus. Some derivation systems are designed especially with mechanization in mind, e.g., the resolution method is easy to implement in software (but its derivations are essentially impossible to understand). Most of these other proof systems represent derivations as trees of wffs rather than sequences. This makes it easier to see which parts of a derivation depend on which other parts.

So for a given logic, such as first-order logic, the different derivation systems will give different explications of what it is for a sentence to be a *theorem* and what it means for a sentence to be derivable from some others. However that is

done (via axiomatic derivations, natural deductions, sequent derivations, truth trees, resolution refutations), we want these relations to match the semantic notions of validity and entailment. Let's write $\vdash \alpha$ for "α is a theorem" and "$\Gamma \vdash \alpha$" for "α is derivable from Γ." However $\vdash$ is defined, we want it to match up with $\models$, that is:

1. $\vdash \alpha$ if and only if $\models \alpha$
2. $\Gamma \vdash \alpha$ if and only if $\Gamma \models \alpha$

The "only if" direction of the above is called *soundness*. A derivation system is sound if derivability guarantees entailment (or validity). Every decent derivation system has to be sound; unsound derivation systems are not useful at all. After all, the entire purpose of a derivation is to provide a syntactic guarantee of validity or entailment. We'll prove soundness for the derivation systems we present.

The converse "if" direction is also important: it is called *completeness*. A complete derivation system is strong enough to show that α is a theorem whenever α is valid, and that there $\Gamma \vdash \alpha$ whenever $\Gamma \models \alpha$. Completeness is harder to establish, and some logics have no complete derivation systems. First-order logic does. Kurt Gödel was the first one to prove completeness for a derivation system of first-order logic in his 1929 dissertation.

Another concept that is connected to derivation systems is that of *consistency*. A set of sentences is called inconsistent if anything whatsoever can be derived from it, and consistent otherwise. Inconsistency is the syntactic counterpart to unsatisfiablity: like unsatisfiable sets, inconsistent sets of sentences do not make good theories, they are defective in a fundamental way. Consistent sets of sentences may not be true or useful, but at least they pass that minimal threshold of logical usefulness. For different derivation systems the specific definition of consistency of sets of sentences might differ, but like $\vdash$, we want consistency to coincide with its semantic counterpart, satisfiability. We want it to always be the case that Γ is consistent if and only if it is satisfiable. Here, the "if" direction amounts to completeness (consistency guarantees satisfiability), and the "only if" direction amounts to soundness (satisfiability guarantees consistency). In fact, for classical first-order logic, the two versions of soundness and completeness are equivalent.

3.2 The Sequent Calculus

While many derivation systems operate with arrangements of sentences, the sequent calculus operates with *sequents*. A sequent is an expression of the form

$$\alpha_1, \ldots, \alpha_m \Rightarrow \beta_1, \ldots, \beta_m,$$

that is a pair of sequences of sentences, separated by the sequent symbol $\Rightarrow$. Either sequence may be empty. A derivation in the sequent calculus is a tree of sequents, where the topmost sequents are of a special form (they are called

"initial sequents" or "axioms") and every other sequent follows from the sequents immediately above it by one of the rules of inference. The rules of inference either manipulate the sentences in the sequents (adding, removing, or rearranging them on either the left or the right), or they introduce a complex wff in the conclusion of the rule. For instance, the $\wedge$L rule allows the inference from $\alpha, \Gamma \Rightarrow \Delta$ to $A \wedge \beta, \Gamma \Rightarrow \Delta$, and the $\rightarrow$R allows the inference from $\alpha, \Gamma \Rightarrow \Delta, \beta$ to $\Gamma \Rightarrow \Delta, \alpha \rightarrow \beta$, for any Γ, Δ, α, and β. (In particular, Γ and Δ may be empty.)

The $\vdash$ relation based on the sequent calculus is defined as follows: $\Gamma \vdash \alpha$ iff there is some sequence Γ_0 such that every α in Γ_0 is in Γ and there is a derivation with the sequent $\Gamma_0 \Rightarrow \alpha$ at its root. α is a theorem in the sequent calculus if the sequent $\Rightarrow \alpha$ has a derivation. For instance, here is a derivation that shows that $\vdash (\alpha \wedge \beta) \rightarrow \alpha$:

$$\dfrac{\dfrac{\alpha \Rightarrow \alpha}{\alpha \wedge \beta \Rightarrow \alpha}\wedge\text{L}}{\Rightarrow (\alpha \wedge \beta) \rightarrow \alpha}\rightarrow\text{R}$$

A set Γ is inconsistent in the sequent calculus if there is a derivation of $\Gamma_0 \Rightarrow$ (where every $\alpha \in \Gamma_0$ is in Γ and the right side of the sequent is empty). Using the rule WR, any sentence can be derived from an inconsistent set.

The sequent calculus was invented in the 1930 by Gerhard Gentzen. Because of its systematic and symmetric design, it is a very useful formalism for developing a theory of derivations. It is relatively easy to find derivations in the sequent calculus, but these derivations are often hard to read and their connection to proofs are sometimes not easy to see. It has proved to be a very elegant approach to derivation systems, however, and many logics have sequent calculus systems.

3.3 Natural Deduction

Natural deduction is a derivation system intended to mirror actual reasoning (especially the kind of regimented reasoning employed by mathematicians). Actual reasoning proceeds by a number of "natural" patterns. For instance, proof by cases allows us to establish a conclusion on the basis of a disjunctive premise, by establishing that the conclusion follows from either of the disjuncts. Indirect proof allows us to establish a conclusion by showing that its negation leads to a contradiction. Conditional proof establishes a conditional claim "if ... then ..." by showing that the consequent follows from the antecedent. Natural deduction is a formalization of some of these natural inferences. Each of the logical connectives and quantifiers comes with two rules, an introduction and an elimination rule, and they each correspond to one such natural inference pattern. For instance, $\rightarrow$Intro corresponds to conditional proof, and $\vee$Elim to proof by cases. A particularly simple rule is $\wedge$Elim which allows the inference from $\alpha \wedge \beta$ to α (or β).

One feature that distinguishes natural deduction from other derivation systems is its use of assumptions. A derivation in natural deduction is a tree of wffs. A single wff stands at the root of the tree of wffs, and the "leaves" of the tree are wffs from which the conclusion is derived. In natural deduction, some leaf wffs play a role inside the derivation but are "used up" by the time the derivation reaches the conclusion. This corresponds to the practice, in actual reasoning, of introducing hypotheses which only remain in effect for a short while. For instance, in a proof by cases, we assume the truth of each of the disjuncts; in conditional proof, we assume the truth of the antecedent; in indirect proof, we assume the truth of the negation of the conclusion. This way of introducing hypothetical assumptions and then doing away with them in the service of establishing an intermediate step is a hallmark of natural deduction. The formulas at the leaves of a natural deduction derivation are called assumptions, and some of the rules of inference may "discharge" them. For instance, if we have a derivation of β from some assumptions which include α, then the $\rightarrow$Intro rule allows us to infer $\alpha \rightarrow \beta$ and discharge any assumption of the form α. (To keep track of which assumptions are discharged at which inferences, we label the inference and the assumptions it discharges with a number.) The assumptions that remain undischarged at the end of the derivation are together sufficient for the truth of the conclusion, and so a derivation establishes that its undischarged assumptions entail its conclusion.

The relation $\Gamma \vdash \alpha$ based on natural deduction holds iff there is a derivation in which α is the last sentence in the tree, and every leaf which is undischarged is in Γ. α is a theorem in natural deduction iff there is a derivation in which α is the last sentence and all assumptions are discharged. For instance, here is a derivation that shows that $\vdash (\alpha \wedge \beta) \rightarrow \alpha$:

$$\cfrac{\cfrac{[\alpha \wedge \beta]^1}{\alpha}\,\wedge\text{Elim}}{(\alpha \wedge \beta) \rightarrow \alpha}\,{}^{1}\rightarrow\text{Intro}$$

The label 1 indicates that the assumption $\alpha \wedge \beta$ is discharged at he $\rightarrow$Intro inference.

A set Γ is inconsistent iff $\Gamma \vdash \bot$ in natural deduction. The rule $\bot_I$ makes it so that from an inconsistent set, any sentence can be derived.

Natural deduction systems were developed by Gerhard Gentzen and Stanisław Jaśkowski in the 1930s, and later developed by Dag Prawitz and Frederic Fitch. Because its inferences mirror natural methods of proof, it is favored by philosophers. The versions developed by Fitch are often used in introductory logic textbooks. In the philosophy of logic, the rules of natural deduction have sometimes been taken to give the meanings of the logical operators ("proof-theoretic semantics").

Chapter 4

The Sequent Calculus

4.1 Rules and Derivations

For the following, let $\Gamma, \Delta, \Pi, \Lambda$ represent finite sequences of sentences.

Definition 41A (Sequent) A *sequent* is an expression of the form

$$\Gamma \Rightarrow \Delta$$

where Γ and Δ are finite (possibly empty) sequences of sentences of the language $\mathcal{L}$. Γ is called the *antecedent*, while Δ is the *succedent*.

The intuitive idea behind a sequent is: if all of the sentences in the antecedent hold, then at least one of the sentences in the succedent holds. That is, if $\Gamma = \langle \alpha_1, \ldots, \alpha_m \rangle$ and $\Delta = \langle \beta_1, \ldots, \beta_n \rangle$, then $\Gamma \Rightarrow \Delta$ holds iff

$$(\alpha_1 \wedge \cdots \wedge \alpha_m) \rightarrow (\beta_1 \vee \cdots \vee \beta_n)$$

holds. There are two special cases: where Γ is empty and when Δ is empty. When Γ is empty, i.e., $m = 0$, $\Rightarrow \Delta$ holds iff $\beta_1 \vee \cdots \vee \beta_n$ holds. When Δ is empty, i.e., $n = 0$, $\Gamma \Rightarrow$ holds iff $\neg(\alpha_1 \wedge \cdots \wedge \alpha_m)$ does. We say a sequent is valid iff the corresponding sentence is valid.

If Γ is a sequence of sentences, we write Γ, α for the result of appending α to the right end of Γ (and α, Γ for the result of appending α to the left end of Γ). If Δ is a sequence of sentences also, then Γ, Δ is the concatenation of the two sequences.

Definition 41B (Initial Sequent) An *initial sequent* is a sequent of the form $\alpha \Rightarrow \alpha$ for any sentence α in the language.

Derivations in the sequent calculus are certain trees of sequents, where the topmost sequents are initial sequents, and if a sequent stands below one or tw other sequents, it must follow correctly by a rule of inference. The rules for **LK** are divided into two main types: *logical* rules and *structural* rules. The logical rules are named for the main operator of the sentence containing α and/or β in the lower sequent. Each one comes in two versions, one for inferring a sequent with the sentence containg the logical operator on the left, and one with the sentence on the right.

4.2 Propositional Rules

Rules for $\neg$

$$\frac{\Gamma \Rightarrow \Delta, \alpha}{\neg\alpha, \Gamma \Rightarrow \Delta}\,\neg\mathrm{L} \qquad \frac{\alpha, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \neg\alpha}\,\neg\mathrm{R}$$

Rules for $\wedge$

$$\frac{\alpha, \Gamma \Rightarrow \Delta}{\alpha \wedge \beta, \Gamma \Rightarrow \Delta}\,\wedge\mathrm{L} \qquad \frac{\beta, \Gamma \Rightarrow \Delta}{\alpha \wedge \beta, \Gamma \Rightarrow \Delta}\,\wedge\mathrm{L} \qquad \frac{\Gamma \Rightarrow \Delta, \alpha \qquad \Gamma \Rightarrow \Delta, \beta}{\Gamma \Rightarrow \Delta, \alpha \wedge \beta}\,\wedge\mathrm{R}$$

Rules for $\vee$

$$\frac{\alpha, \Gamma \Rightarrow \Delta \qquad \beta, \Gamma \Rightarrow \Delta}{\alpha \vee \beta, \Gamma \Rightarrow \Delta}\,\vee\mathrm{L} \qquad \frac{\Gamma \Rightarrow \Delta, \alpha}{\Gamma \Rightarrow \Delta, \alpha \vee \beta}\,\vee\mathrm{R} \qquad \frac{\Gamma \Rightarrow \Delta, \beta}{\Gamma \Rightarrow \Delta, \alpha \vee \beta}\,\vee\mathrm{R}$$

Rules for $\rightarrow$

$$\frac{\Gamma \Rightarrow \Delta, \alpha \qquad \beta, \Pi \Rightarrow \Lambda}{\alpha \rightarrow \beta, \Gamma, \Pi \Rightarrow \Delta, \Lambda}\,\rightarrow\mathrm{L} \qquad \frac{\alpha, \Gamma \Rightarrow \Delta, \beta}{\Gamma \Rightarrow \Delta, \alpha \rightarrow \beta}\,\rightarrow\mathrm{R}$$

4.3 Quantifier Rules

Rules for $\forall$

$$\frac{\alpha(t), \Gamma \Rightarrow \Delta}{\forall x\, \alpha(x), \Gamma \Rightarrow \Delta}\,\forall\mathrm{L} \qquad \frac{\Gamma \Rightarrow \Delta, \alpha(a)}{\Gamma \Rightarrow \Delta, \forall x\, \alpha(x)}\,\forall\mathrm{R}$$

In $\forall$L, t is a closed term (i.e., one without variables). In $\forall$R, a is a constant symbol which must not occur anywhere in the lower sequent of the $\forall$R rule. We call a the *eigenvariable* of the $\forall$R inference.

Rules for $\exists$

$$\frac{\alpha(a), \Gamma \Rightarrow \Delta}{\exists x\, \alpha(x), \Gamma \Rightarrow \Delta}\,\exists\mathrm{L} \qquad \frac{\Gamma \Rightarrow \Delta, \alpha(t)}{\Gamma \Rightarrow \Delta, \exists x\, \alpha(x)}\,\exists\mathrm{R}$$

Again, t is a closed term, and a is a constant symbol which does not occur in the lower sequent of the $\exists$L rule. We call a the *eigenvariable* of the $\exists$L inference.

The condition that an eigenvariable not occur in the lower sequent of the $\forall$R or $\exists$L inference is called the *eigenvariable condition*.

We use the term "eigenvariable" even though a in the above rules is a constant symbol. This has historical reasons.

In ∃R and ∀L there are no restrictions on the term t. On the other hand, in the ∃L and ∀R rules, the eigenvariable condition requires that the constant symbol a does not occur anywhere outside of $\alpha(a)$ in the upper sequent. It is necessary to ensure that the system is sound, i.e., only derives sequents that are valid. Without this condition, the following would be allowed:

$$\dfrac{\dfrac{\alpha(a) \Rightarrow \alpha(a)}{\exists x\, \alpha(x) \Rightarrow \alpha(a)}\,{*\exists}\mathrm{L}}{\exists x\, \alpha(x) \Rightarrow \forall x\, \alpha(x)}\,\forall\mathrm{R} \qquad \dfrac{\dfrac{\alpha(a) \Rightarrow \alpha(a)}{\alpha(a) \Rightarrow \forall x\, \alpha(x)}\,{*\forall}\mathrm{R}}{\exists x\, \alpha(x) \Rightarrow \forall x\, \alpha(x)}\,\exists\mathrm{L}$$

However, $\exists x\, \alpha(x) \Rightarrow \forall x\, \alpha(x)$ is not valid.

4.4 Structural Rules

We also need a few rules that allow us to rearrange sentences in the left and right side of a sequent. Since the logical rules require that the sentences in the premise which the rule acts upon stand either to the far left or to the far right, we need an "exchange" rule that allows us to move sentences to the right position. It's also important sometimes to be able to combine two identical sentences into one, and to add a sentence on either side.

Weakening

$$\dfrac{\Gamma \Rightarrow \Delta}{\alpha, \Gamma \Rightarrow \Delta}\,\mathrm{WL} \qquad \dfrac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \alpha}\,\mathrm{WR}$$

Contraction

$$\dfrac{\alpha, \alpha, \Gamma \Rightarrow \Delta}{\alpha, \Gamma \Rightarrow \Delta}\,\mathrm{CL} \qquad \dfrac{\Gamma \Rightarrow \Delta, \alpha, \alpha}{\Gamma \Rightarrow \Delta, \alpha}\,\mathrm{CR}$$

Exchange

$$\dfrac{\Gamma, \alpha, \beta, \Pi \Rightarrow \Delta}{\Gamma, \beta, \alpha, \Pi \Rightarrow \Delta}\,\mathrm{XL} \qquad \dfrac{\Gamma \Rightarrow \Delta, \alpha, \beta, \Lambda}{\Gamma \Rightarrow \Delta, \beta, \alpha, \Lambda}\,\mathrm{XR}$$

A series of weakening, contraction, and exchange inferences will often be indicated by double inference lines.

The following rule, called "cut," is not strictly speaking necessary, but makes it a lot easier to reuse and combine derivations.

$$\dfrac{\Gamma \Rightarrow \Delta, \alpha \qquad \alpha, \Pi \Rightarrow \Lambda}{\Gamma, \Pi \Rightarrow \Delta, \Lambda}\,\mathrm{Cut}$$

4.5 Derivations

We've said what an initial sequent looks like, and we've given the rules of inference. Derivations in the sequent calculus are inductively generated from

these: each derivation either is an initial sequent on its own, or consists of one or two derivations followed by an inference.

DEFINITION 45A (**LK** DERIVATION) An **LK**-*derivation* of a sequent S is a tree of sequents satisfying the following conditions:

1. The topmost sequents of the tree are initial sequents.
2. The bottommost sequent of the tree is S.
3. Every sequent in the tree except S is a premise of a correct application of an inference rule whose conclusion stands directly below that sequent in the tree.

We then say that S is the *end-sequent* of the derivation and that S is *derivable in* **LK** (or **LK**-derivable).

Example 4.5.2. Every initial sequent, e.g., $\gamma \Rightarrow \gamma$ is a derivation. We can obtain a new derivation from this by applying, say, the WL rule,

$$\dfrac{\Gamma \Rightarrow \Delta}{\alpha, \Gamma \Rightarrow \Delta}\,\text{WL}$$

The rule, however, is meant to be general: we can replace the α in the rule with any sentence, e.g., also with δ. If the premise matches our initial sequent $\gamma \Rightarrow \gamma$, that means that both Γ and Δ are just γ, and the conclusion would then be $\delta, \gamma \Rightarrow \gamma$. So, the following is a derivation:

$$\dfrac{\gamma \Rightarrow \gamma}{\delta, \gamma \Rightarrow \gamma}\,\text{WL}$$

We can now apply another rule, say XL, which allows us to switch two sentences on the left. So, the following is also a correct derivation:

$$\dfrac{\dfrac{\gamma \Rightarrow \gamma}{\delta, \gamma \Rightarrow \gamma}\,\text{WL}}{\gamma, \delta \Rightarrow \gamma}\,\text{XL}$$

In this application of the rule, which was given as

$$\dfrac{\Gamma, \alpha, \beta, \Pi \Rightarrow \Delta}{\Gamma, \beta, \alpha, \Pi \Rightarrow \Delta,}\,\text{XL}$$

both Γ and Π were empty, Δ is γ, and the roles of α and β are played by δ and γ, respectively. In much the same way, we also see that

$$\dfrac{\delta \Rightarrow \delta}{\gamma, \delta \Rightarrow \delta}\,\text{WL}$$

is a derivation. Now we can take these two derivations, and combine them using $\wedge$R. That rule was

$$\dfrac{\Gamma \Rightarrow \Delta, \alpha \qquad \Gamma \Rightarrow \Delta, \beta}{\Gamma \Rightarrow \Delta, \alpha \wedge \beta}\,\wedge\mathrm{R}$$

In our case, the premises must match the last sequents of the derivations ending in the premises. That means that Γ is γ, δ, Δ is empty, α is γ and β is δ. So the conclusion, if the inference should be correct, is $\gamma, \delta \Rightarrow \gamma \wedge \delta$. Of course, we can also reverse the premises, then α would be δ and β would be γ. So both of the following are correct derivations.

$$\dfrac{\dfrac{\dfrac{\dfrac{\gamma \Rightarrow \gamma}{\delta, \gamma \Rightarrow \gamma}\,\mathrm{WL}}{\gamma, \delta \Rightarrow \gamma}\,\mathrm{XL} \qquad \dfrac{\delta \Rightarrow \delta}{\gamma, \delta \Rightarrow \delta}\,\mathrm{WL}}{\gamma, \delta \Rightarrow \gamma \wedge \delta}\,\wedge\mathrm{R} \qquad \dfrac{\dfrac{\dfrac{\delta \Rightarrow \delta}{\gamma, \delta \Rightarrow \delta}\,\mathrm{WL} \qquad \dfrac{\dfrac{\gamma \Rightarrow \gamma}{\delta, \gamma \Rightarrow \gamma}\,\mathrm{WL}}{\gamma, \delta \Rightarrow \gamma}\,\mathrm{XL}}{\gamma, \delta \Rightarrow \delta \wedge \gamma}\,\wedge\mathrm{R}}$$

4.6 Examples of Derivations

Example 4.6.1. Give an **LK**-derivation for the sequent $\alpha \wedge \beta \Rightarrow \alpha$.

We begin by writing the desired end-sequent at the bottom of the derivation.

$$\dfrac{}{\alpha \wedge \beta \Rightarrow \alpha}$$

Next, we need to figure out what kind of inference could have a lower sequent of this form. This could be a structural rule, but it is a good idea to start by looking for a logical rule. The only logical connective occurring in the lower sequent is $\wedge$, so we're looking for an $\wedge$ rule, and since the $\wedge$ symbol occurs in the antecedent, we're looking at the $\wedge$L rule.

$$\dfrac{}{\alpha \wedge \beta \Rightarrow \alpha}\,\wedge\mathrm{L}$$

There are two options for what could have been the upper sequent of the $\wedge$L inference: we could have an upper sequent of $\alpha \Rightarrow \alpha$, or of $\beta \Rightarrow \alpha$. Clearly, $\alpha \Rightarrow \alpha$ is an initial sequent (which is a good thing), while $\beta \Rightarrow \alpha$ is not derivable in general. We fill in the upper sequent:

$$\dfrac{\alpha \Rightarrow \alpha}{\alpha \wedge \beta \Rightarrow \alpha}\,\wedge\mathrm{L}$$

We now have a correct **LK**-derivation of the sequent $\alpha \wedge \beta \Rightarrow \alpha$.

Example 4.6.2. Give an **LK**-derivation for the sequent $\neg\alpha \vee \beta \Rightarrow \alpha \to \beta$.

Begin by writing the desired end-sequent at the bottom of the derivation.

$$\dfrac{}{\neg\alpha \vee \beta \Rightarrow \alpha \to \beta}$$

To find a logical rule that could give us this end-sequent, we look at the logical connectives in the end-sequent: $\neg$, $\vee$, and $\to$. We only care at the moment about $\vee$ and $\to$ because they are main operators of sentences in the end-sequent, while $\neg$ is inside the scope of another connective, so we will take care of it later. Our options for logical rules for the final inference are therefore the $\vee$L rule and the $\to$R rule. We could pick either rule, really, but let's pick the $\to$R rule

(if for no reason other than it allows us to put off splitting into two branches). According to the form of →R inferences which can yield the lower sequent, this must look like:

$$\dfrac{\dfrac{}{\alpha, \neg\alpha \vee \beta \Rightarrow \beta}}{\neg\alpha \vee \beta \Rightarrow \alpha \to \beta} \to\text{R}$$

If we move $\neg\alpha \vee \beta$ to the outside of the antecedent, we can apply the ∨L rule. According to the schema, this must split into two upper sequents as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg\alpha, \alpha \Rightarrow \beta} \quad \dfrac{}{\beta, \alpha \Rightarrow \beta}}{\neg\alpha \vee \beta, \alpha \Rightarrow \beta} \vee\text{L}}{\alpha, \neg\alpha \vee \beta \Rightarrow \beta} \text{XR}}{\neg\alpha \vee \beta \Rightarrow \alpha \to \beta} \to\text{R}$$

Remember that we are trying to wind our way up to initial sequents; we seem to be pretty close! The right branch is just one weakening and one exchange away from an initial sequent and then it is done:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\neg\alpha, \alpha \Rightarrow \beta} \quad \dfrac{\dfrac{\dfrac{}{\beta \Rightarrow \beta}}{\alpha, \beta \Rightarrow \beta} \text{WL}}{\beta, \alpha \Rightarrow \beta} \text{XL}}{\neg\alpha \vee \beta, \alpha \Rightarrow \beta} \vee\text{L}}{\alpha, \neg\alpha \vee \beta \Rightarrow \beta} \text{XR}}{\neg\alpha \vee \beta \Rightarrow \alpha \to \beta} \to\text{R}$$

Now looking at the left branch, the only logical connective in any sentence is the ¬ symbol in the antecedent sentences, so we're looking at an instance of the ¬L rule.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\alpha \Rightarrow \beta, \alpha}}{\neg\alpha, \alpha \Rightarrow \beta} \neg\text{L} \quad \dfrac{\dfrac{\dfrac{}{\beta \Rightarrow \beta}}{\alpha, \beta \Rightarrow \beta} \text{WL}}{\beta, \alpha \Rightarrow \beta} \text{XL}}{\neg\alpha \vee \beta, \alpha \Rightarrow \beta} \vee\text{L}}{\alpha, \neg\alpha \vee \beta \Rightarrow \beta} \text{XR}}{\neg\alpha \vee \beta \Rightarrow \alpha \to \beta} \to\text{R}$$

Similarly to how we finished off the right branch, we are just one weakening and one exchange away from finishing off this left branch as well.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\alpha \Rightarrow \alpha}}{\alpha \Rightarrow \alpha, \beta} \text{WR}}{\alpha \Rightarrow \beta, \alpha} \text{XR}}{\neg\alpha, \alpha \Rightarrow \beta} \neg\text{L} \quad \dfrac{\dfrac{\dfrac{}{\beta \Rightarrow \beta}}{\alpha, \beta \Rightarrow \beta} \text{WL}}{\beta, \alpha \Rightarrow \beta} \text{XL}}{\neg\alpha \vee \beta, \alpha \Rightarrow \beta} \vee\text{L}}{\alpha, \neg\alpha \vee \beta \Rightarrow \beta} \text{XR}}{\neg\alpha \vee \beta \Rightarrow \alpha \to \beta} \to\text{R}$$

Example 4.6.3. Give an **LK**-derivation of the sequent $\neg\alpha \vee \neg\beta \Rightarrow \neg(\alpha \wedge \beta)$

Using the techniques from above, we start by writing the desired end-sequent at the bottom.

$$\dfrac{}{\neg\alpha \vee \neg\beta \;\Rightarrow\; \neg(\alpha \wedge \beta)}$$

The available main connectives of sentences in the end-sequent are the $\vee$ symbol and the $\neg$ symbol. It would work to apply either the $\vee$L or the $\neg$R rule here, but we start with the $\neg$R rule because it avoids splitting up into two branches for a moment:

$$\dfrac{\dfrac{}{\alpha \wedge \beta, \neg\alpha \vee \neg\beta \;\Rightarrow}}{\neg\alpha \vee \neg\beta \;\Rightarrow\; \neg(\alpha \wedge \beta)}\,\neg\text{R}$$

Now we have a choice of whether to look at the $\wedge$L or the $\vee$L rule. Let's see what happens when we apply the $\wedge$L rule: we have a choice to start with either the sequent $\alpha, \neg\alpha \vee \beta \Rightarrow$ or the sequent $\beta, \neg\alpha \vee \beta \Rightarrow$. Since the proof is symmetric with regards to α and β, let's go with the former:

$$\dfrac{\dfrac{\dfrac{}{\alpha, \neg\alpha \vee \neg\beta \;\Rightarrow}}{\alpha \wedge \beta, \neg\alpha \vee \neg\beta \;\Rightarrow}\,\wedge\text{L}}{\neg\alpha \vee \neg\beta \;\Rightarrow\; \neg(\alpha \wedge \beta)}\,\neg\text{R}$$

Continuing to fill in the derivation, we see that we run into a problem:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\alpha \;\Rightarrow\; \alpha}{\neg\alpha, \alpha \;\Rightarrow}\,\neg\text{L} \qquad \dfrac{\dfrac{}{\alpha \;\Rightarrow\; \beta}\,?}{\neg\beta, \alpha \;\Rightarrow}\,\neg\text{L}}{\neg\alpha \vee \neg\beta, \alpha \;\Rightarrow}\,\vee\text{L}}{\alpha, \neg\alpha \vee \neg\beta \;\Rightarrow}\,\text{XL}}{\alpha \wedge \beta, \neg\alpha \vee \neg\beta \;\Rightarrow}\,\wedge\text{L}}{\neg\alpha \vee \neg\beta \;\Rightarrow\; \neg(\alpha \wedge \beta)}\,\neg\text{R}$$

The top of the right branch cannot be reduced any further, and it cannot be brought by way of structural inferences to an initial sequent, so this is not the right path to take. So clearly, it was a mistake to apply the $\wedge$L rule above. Going back to what we had before and carrying out the $\vee$L rule instead, we get

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg\alpha, \alpha \wedge \beta \;\Rightarrow} \qquad \dfrac{}{\neg\beta, \alpha \wedge \beta \;\Rightarrow}}{\neg\alpha \vee \neg\beta, \alpha \wedge \beta \;\Rightarrow}\,\vee\text{L}}{\alpha \wedge \beta, \neg\alpha \vee \neg\beta \;\Rightarrow}\,\text{XL}}{\neg\alpha \vee \neg\beta \;\Rightarrow\; \neg(\alpha \wedge \beta)}\,\neg\text{R}$$

Completing each branch as we've done before, we get

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\alpha \Rightarrow \alpha}}{\alpha \wedge \beta \Rightarrow \alpha}\wedge\mathrm{L}}{\neg\alpha, \alpha \wedge \beta \Rightarrow}\neg\mathrm{L} \quad \dfrac{\dfrac{\dfrac{}{\beta \Rightarrow \beta}}{\alpha \wedge \beta \Rightarrow \beta}\wedge\mathrm{L}}{\neg\beta, \alpha \wedge \beta \Rightarrow}\neg\mathrm{L}}{\neg\alpha \vee \neg\beta, \alpha \wedge \beta \Rightarrow}\vee\mathrm{L}}{\alpha \wedge \beta, \neg\alpha \vee \neg\beta \Rightarrow}\,\mathrm{XL}}{\neg\alpha \vee \neg\beta \Rightarrow \neg(\alpha \wedge \beta)}\neg\mathrm{R}$$

(We could have carried out the $\wedge$ rules lower than the $\neg$ rules in these steps and still obtained a correct derivation).

Example 4.6.4. So far we haven't used the contraction rule, but it is sometimes required. Here's an example where that happens. Suppose we want to prove $\Rightarrow A \vee \neg\alpha$. Applying $\vee$R backwards would give us one of these two derivations:

$$\dfrac{\dfrac{}{\Rightarrow \alpha}}{\Rightarrow \alpha \vee \neg\alpha}\vee\mathrm{R} \qquad \dfrac{\dfrac{\dfrac{}{\alpha \Rightarrow}}{\Rightarrow \neg\alpha}\neg\mathrm{R}}{\Rightarrow \alpha \vee \neg\alpha}\vee\mathrm{R}$$

Neither of these of course ends in an initial sequent. The trick is to realize that the contraction rule allows us to combine two copies of a sentence into one—and when we're searching for a proof, i.e., going from bottom to top, we can keep a copy of $\alpha \vee \neg\alpha$ in the premise, e.g.,

$$\dfrac{\dfrac{\dfrac{}{\Rightarrow \alpha \vee \neg\alpha, \alpha}}{\Rightarrow \alpha \vee \neg\alpha, \alpha \vee \neg\alpha}\vee\mathrm{R}}{\Rightarrow \alpha \vee \neg\alpha}\,\mathrm{CR}$$

Now we can apply $\vee$R a second time, and also get $\neg\alpha$, which leads to a complete derivation.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\alpha \Rightarrow \alpha}}{\Rightarrow \alpha, \neg\alpha}\neg\mathrm{R}}{\Rightarrow \alpha, \alpha \vee \neg\alpha}\vee\mathrm{R}}{\Rightarrow \alpha \vee \neg\alpha, \alpha}\,\mathrm{XR}}{\Rightarrow \alpha \vee \neg\alpha, \alpha \vee \neg\alpha}\vee\mathrm{R}}{\Rightarrow \alpha \vee \neg\alpha}\,\mathrm{CR}$$

4.7 Derivations with Quantifiers

Example 4.7.1. Give an **LK**-derivation of the sequent $\exists x\, \neg\alpha(x) \Rightarrow \neg\forall x\, \alpha(x)$.

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 lower down in the finished proof). Also, it is a good idea to try and look ahead and try to guess what the initial sequent might look like. In our case, it will have to be something like $\alpha(a) \Rightarrow \alpha(a)$. That means that when we are "reversing" the quantifier rules, we will have to pick the same term—what we will call a—for

both the $\forall$ and the $\exists$ rule. If we picked different terms for each rule, we would end up with something like $\alpha(a) \Rightarrow \alpha(b)$, which, of course, is not derivable.

Starting as usual, we write

$$\frac{}{\exists x\, \neg\alpha(x) \;\Rightarrow\; \neg\forall x\, \alpha(x)}$$

We could either carry out the $\exists$L rule or the $\neg$R rule. Since the $\exists$L rule is subject to the eigenvariable condition, it's a good idea to take care of it sooner rather than later, so we'll do that one first.

$$\dfrac{\dfrac{}{\neg\alpha(a) \;\Rightarrow\; \neg\forall x\, \alpha(x)}}{\exists x\, \neg\alpha(x) \;\Rightarrow\; \neg\forall x\, \alpha(x)}\,\exists\text{L}$$

Applying the $\neg$L and $\neg$R rules backwards, we get

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\forall x\, \alpha(x) \;\Rightarrow\; \alpha(a)}}{\neg\alpha(a), \forall x\, \alpha(x) \;\Rightarrow}\,\neg\text{L}}{\forall x\, \alpha(x), \neg\alpha(a) \;\Rightarrow}\,\text{XL}}{\neg\alpha(a) \;\Rightarrow\; \neg\forall x\alpha(x)}\,\neg\text{R}}{\exists x\neg\alpha(x) \;\Rightarrow\; \neg\forall x\alpha(x)}\,\exists\text{L}$$

At this point, our only option is to carry out the $\forall$L rule. Since this rule is not subject to the eigenvariable restriction, we're in the clear. Remember, we want to try and obtain an initial sequent (of the form $\alpha(a) \Rightarrow \alpha(a)$), so we should choose a as our argument for α when we apply the rule.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\alpha(a) \;\Rightarrow\; \alpha(a)}{\forall x\, \alpha(x) \;\Rightarrow\; \alpha(a)}\,\forall\text{L}}{\neg\alpha(a), \forall x\, \alpha(x) \;\Rightarrow}\,\neg\text{L}}{\forall x\, \alpha(x), \neg\alpha(a) \;\Rightarrow}\,\text{XL}}{\neg\alpha(a) \;\Rightarrow\; \neg\forall x\, \alpha(x)}\,\neg\text{R}}{\exists x\, \neg\alpha(x) \;\Rightarrow\; \neg\forall x\, \alpha(x)}\,\exists\text{L}$$

It is important, especially when dealing with quantifiers, to double check at this point that the eigenvariable condition has not been violated. Since the only rule we applied that is subject to the eigenvariable condition was $\exists$L, and the eigenvariable a does not occur in its lower sequent (the end-sequent), this is a correct derivation.

4.8 Proof-Theoretic Notions

Just as we've defined a number of important semantic notions (validity, entailment, satisfiabilty), we now define corresponding *proof-theoretic notions*. These are not defined by appeal to satisfaction of sentences in structures, but by appeal to the derivability or non-derivability of certain sequents. It was an

important discovery, due to Gödel, that these notions coincide. That they do is the content of the *completeness theorem*.

DEFINITION 48A (THEOREMS) A sentence α is a *theorem* if there is a derivation in **LK** of the sequent $\Rightarrow \alpha$. We write $\vdash \alpha$ if α is a theorem and $\nvdash \alpha$ if it is not.

DEFINITION 48B (DERIVABILITY) A sentence α is *derivable from* a set of sentences Γ, $\Gamma \vdash \alpha$, iff there is a finite subset $\Gamma_0 \subseteq \Gamma$ and a sequence Γ_0' of the sentences in Γ_0 such that **LK** derives $\Gamma_0' \Rightarrow \alpha$. If α is not derivable from Γ we write $\Gamma \nvdash \alpha$.

Because of the contraction, weakening, and exchange rules, the order and number of sentences in Γ_0' does not matter: if a sequent $\Gamma_0' \Rightarrow \alpha$ is derivable, then so is $\Gamma_0'' \Rightarrow \alpha$ for any Γ_0'' that contains the same sentences as Γ_0'. For instance, if $\Gamma_0 = \{\beta, \gamma\}$ then both $\Gamma_0' = \langle \beta, \beta, \gamma \rangle$ and $\Gamma_0'' = \langle \gamma, \gamma, \beta \rangle$ are sequences containing just the sentences in Γ_0. If a sequent containing one is derivable, so is the other, e.g.:

$$\dfrac{\dfrac{\dfrac{\vdots}{\beta, \beta, \gamma \Rightarrow \alpha}\,\text{CL}}{\beta, \gamma \Rightarrow \alpha}\,\text{XL}}{\gamma, \beta \Rightarrow \alpha}\,\text{WL}$$

From now on we'll say that if Γ_0 is a finite set of sentences then $\Gamma_0 \Rightarrow \alpha$ is any sequent where the antecedent is a sequence of sentences in Γ_0 and tacitly include contractions, exchanges, and weakenings if necessary.

DEFINITION 48C (CONSISTENCY) A set of sentences Γ is *inconsistent* iff there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that **LK** derives $\Gamma_0 \Rightarrow$. If Γ is not inconsistent, i.e., if for every finite $\Gamma_0 \subseteq \Gamma$, **LK** does not derive $\Gamma_0 \Rightarrow$, we say it is *consistent*.

PROPOSITION 48D (REFLEXIVITY) If $\alpha \in \Gamma$, then $\Gamma \vdash \alpha$.

Proof. The initial sequent $\alpha \Rightarrow \alpha$ is derivable, and $\{\alpha\} \subseteq \Gamma$. □

PROPOSITION 48E (MONOTONY) If $\Gamma \subseteq \Delta$ and $\Gamma \vdash \alpha$, then $\Delta \vdash \alpha$.

Proof. Any finite $\Gamma_0 \subseteq \Gamma$ is also a finite subset of Δ, so a derivation of $\Gamma_0 \Rightarrow \alpha$ also shows $\Delta \vdash \alpha$. □

PROPOSITION 48F (TRANSITIVITY) If $\Gamma \vdash \alpha$ for every $\alpha \in \Delta$ and $\Delta \vdash \beta$, then $\Gamma \vdash \beta$.

Proof. If $\Delta \vdash \beta$, then for some finite subset $\Delta_0 \subseteq \Delta$, there is a derivation of $\Delta_0 \Rightarrow \beta$. We show that $\Gamma \vdash \beta$ by induction on the number n of sentences in Δ_0.

If $n = 0$, then Δ_0 is empty, and $\Rightarrow \beta$ is provable. Since $\emptyset \subseteq \Gamma$, $\Gamma \vdash \beta$.

Otherwise, pick $\alpha \in \Delta_0$ and let $\Delta_1 = \Delta_0 \setminus \{\alpha\}$. There is a derivation π_0 of $\alpha, \Delta_1 \Rightarrow \beta$. We obtain the derivation π_1:

$$\dfrac{\begin{array}{c}\vdots\\ \pi\\ \vdots\\ \alpha, \Delta_1 \Rightarrow \beta\end{array}}{\Delta_1 \Rightarrow \alpha \to \beta} \to\text{R}$$

Since the number of sentences in Δ_1 is $n-1$, the inductive hypothesis applies: there is a derivation π_2 of $\Gamma_0 \Rightarrow \alpha \to \beta$ for some $\Gamma_0 \subseteq \Gamma$. Since $\Gamma \vdash \alpha$ there is also a derivation π_3 of $\Gamma_1 \Rightarrow \alpha$. Now consider:

$$\dfrac{\begin{array}{c}\vdots\\ \pi_2\\ \vdots\\ \Gamma_0 \Rightarrow \alpha \to \beta\end{array} \qquad \dfrac{\begin{array}{c}\vdots\\ \pi_3\\ \vdots\\ \Gamma_1 \Rightarrow \alpha\end{array} \qquad \beta \Rightarrow \beta}{\alpha \to \beta, \Gamma_1 \Rightarrow \beta} \to\text{L}}{\Gamma_0, \Gamma_1 \Rightarrow \beta} \text{Cut}$$

This shows $\Gamma \vdash \beta$. □

PROPOSITION 48G Γ is inconsistent iff $\Gamma \vdash \alpha$ for every sentence α.

Proof. Exercise. □

PROPOSITION 48H (COMPACTNESS)
1. If $\Gamma \vdash \alpha$ then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vdash \alpha$.
2. If every finite subset of Γ is consistent, then Γ is consistent.

Proof.
1. If $\Gamma \vdash \alpha$, then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that the sequent $\Gamma_0 \Rightarrow \alpha$ has a derivation. Consequently, $\Gamma_0 \vdash \alpha$.
2. If Γ is inconsistent, there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that **LK** derives $\Gamma_0 \Rightarrow$. But then Γ_0 is a finite subset of Γ that is inconsistent.

□

4.9 Derivability and Consistency

We will now establish a number of properties of the derivability relation. They are independently interesting, but each will play a role in the proof of the completeness theorem.

PROPOSITION 49A If $\Gamma \vdash \alpha$ and $\Gamma \cup \{\alpha\}$ is inconsistent, then Γ is inconsistent.

Proof. There are finite Γ_0 and $\Gamma_1 \subseteq \Gamma$ such that **LK** derives $\Gamma_0 \Rightarrow \alpha$ and $\alpha, \Gamma_1 \Rightarrow$. Let the **LK**-derivation of $\Gamma_0 \Rightarrow \alpha$ be π_0 and the **LK**-derivation of $\Gamma_1, \alpha \Rightarrow$ be π_1. We can then derive

$$\dfrac{\begin{array}{cc}\vdots\,\pi_0 & \vdots\,\pi_1\\ \Gamma_0 \Rightarrow \alpha & \alpha, \Gamma_1 \Rightarrow\end{array}}{\Gamma_0, \Gamma_1 \Rightarrow}\ \text{Cut}$$

Since $\Gamma_0 \subseteq \Gamma$ and $\Gamma_1 \subseteq \Gamma$, $\Gamma_0 \cup \Gamma_1 \subseteq \Gamma$, hence Γ is inconsistent. □

PROPOSITION 49B $\Gamma \vdash \alpha$ iff $\Gamma \cup \{\neg\alpha\}$ is inconsistent.

Proof. First suppose $\Gamma \vdash \alpha$, i.e., there is a derivation π_0 of $\Gamma \Rightarrow \alpha$. By adding a ¬L rule, we obtain a derivation of $\neg\alpha, \Gamma \Rightarrow$, i.e., $\Gamma \cup \{\neg\alpha\}$ is inconsistent.

If $\Gamma \cup \{\neg A\}$ is inconsistent, there is a derivation π_1 of $\neg\alpha, \Gamma \Rightarrow$. The following is a derivation of $\Gamma \Rightarrow \alpha$:

$$\dfrac{\dfrac{\alpha \Rightarrow \alpha}{\Rightarrow \alpha, \neg\alpha}\,\neg\text{R} \qquad \begin{array}{c}\vdots\,\pi_1\\ \neg\alpha, \Gamma \Rightarrow\end{array}}{\Gamma \Rightarrow \alpha}\ \text{Cut}$$

□

PROPOSITION 49C If $\Gamma \vdash \alpha$ and $\neg\alpha \in \Gamma$, then Γ is inconsistent.

Proof. Suppose $\Gamma \vdash \alpha$ and $\neg\alpha \in \Gamma$. Then there is a derivation π of a sequent $\Gamma_0 \Rightarrow \alpha$. The sequent $\neg\alpha, \Gamma_0 \Rightarrow$ is also derivable:

$$\dfrac{\begin{array}{cc}\vdots\,\pi & \dfrac{\dfrac{\alpha \Rightarrow \alpha}{\neg\alpha, \alpha \Rightarrow}\,\neg\text{L}}{\alpha, \neg\alpha \Rightarrow}\,\text{XL}\\ \Gamma_0 \Rightarrow \alpha & \end{array}}{\Gamma, \neg\alpha \Rightarrow}\ \text{Cut}$$

Since $\neg\alpha \in \Gamma$ and $\Gamma_0 \subseteq \Gamma$, this shows that Γ is inconsistent. □

PROPOSITION 49D If $\Gamma \cup \{\alpha\}$ and $\Gamma \cup \{\neg\alpha\}$ are both inconsistent, then Γ is inconsistent.

Proof. There are finite sets $\Gamma_0 \subseteq \Gamma$ and $\Gamma_1 \subseteq \Gamma$ and **LK**-derivations π_0 and π_1 of $\alpha, \Gamma_0 \Rightarrow$ and $\neg\alpha, \Gamma_1 \Rightarrow$, respectively. We can then derive

$$\dfrac{\dfrac{\begin{array}{c}\vdots\,\pi_0\\ \alpha, \Gamma_0 \Rightarrow\end{array}}{\Gamma_0 \Rightarrow \neg\alpha}\,\neg\text{R} \qquad \begin{array}{c}\vdots\,\pi_1\\ \neg\alpha, \Gamma_1 \Rightarrow\end{array}}{\Gamma_0, \Gamma_1 \Rightarrow}\ \text{Cut}$$

Since $\Gamma_0 \subseteq \Gamma$ and $\Gamma_1 \subseteq \Gamma$, $\Gamma_0 \cup \Gamma_1 \subseteq \Gamma$. Hence Γ is inconsistent. □

4.10 Derivability and the Propositional Connectives

PROPOSITION 410A 1. Both $\alpha \wedge \beta \vdash \alpha$ and $\alpha \wedge \beta \vdash \beta$.

2. $\alpha, \beta \vdash \alpha \wedge \beta$.

Proof. 1. Both sequents $\alpha \wedge \beta \Rightarrow \alpha$ and $\alpha \wedge \beta \Rightarrow \beta$ are derivable:

$$\dfrac{\alpha \Rightarrow \alpha}{\alpha \wedge \beta \Rightarrow \alpha}\,\wedge\mathrm{L} \qquad \dfrac{\beta \Rightarrow \beta}{\alpha \wedge \beta \Rightarrow \beta}\,\wedge\mathrm{L}$$

2. Here is a derivation of the sequent $\alpha, \beta \Rightarrow \alpha \wedge \beta$:

$$\dfrac{\alpha \Rightarrow \alpha \qquad \beta \Rightarrow \beta}{\alpha, \beta \Rightarrow \alpha \wedge \beta}\,\wedge\mathrm{R}$$

□

PROPOSITION 410B 1. $\alpha \vee \beta, \neg\alpha, \neg\beta$ is inconsistent.

2. Both $\alpha \vdash \alpha \vee \beta$ and $\beta \vdash \alpha \vee \beta$.

Proof. 1. We give a derivation of the sequent $\alpha \vee \beta, \neg\alpha, \neg\beta \Rightarrow$:

$$\dfrac{\dfrac{\dfrac{\alpha \Rightarrow \alpha}{\neg\alpha, \alpha \Rightarrow}\,\neg\mathrm{L}}{\alpha, \neg\alpha, \neg\beta \Rightarrow} \qquad \dfrac{\dfrac{\beta \Rightarrow \beta}{\neg\beta, \beta \Rightarrow}\,\neg\mathrm{L}}{\beta, \neg\alpha, \neg\beta \Rightarrow}}{\alpha \vee \beta, \neg\alpha, \neg\beta \Rightarrow}\,\vee\mathrm{L}$$

(Recall that double inference lines indicate several weakening, contraction, and exchange inferences.)

2. Both sequents $\alpha \Rightarrow \alpha \vee \beta$ and $\beta \Rightarrow \alpha \vee \beta$ have derivations:

$$\dfrac{\alpha \Rightarrow \alpha}{\alpha \Rightarrow \alpha \vee \beta}\,\vee\mathrm{R} \qquad \dfrac{\beta \Rightarrow \beta}{\beta \Rightarrow \alpha \vee \beta}\,\vee\mathrm{R}$$

□

PROPOSITION 410C 1. $\alpha, \alpha \to \beta \vdash \beta$.

2. Both $\neg\alpha \vdash \alpha \to \beta$ and $\beta \vdash \alpha \to \beta$.

Proof. 1. The sequent $\alpha \to \beta, \alpha \Rightarrow \beta$ is derivable:

$$\dfrac{\alpha \Rightarrow \alpha \qquad \beta \Rightarrow \beta}{\alpha \to \beta, \alpha \Rightarrow \beta}\,\to\mathrm{L}$$

2. Both sequents $\neg\alpha \Rightarrow \alpha \to \beta$ and $\beta \Rightarrow \alpha \to \beta$ are derivable:

$$\dfrac{\dfrac{\dfrac{\dfrac{\alpha \Rightarrow \alpha}{\neg\alpha, \alpha \Rightarrow}\,\neg\mathrm{L}}{\alpha, \neg\alpha \Rightarrow}\,\mathrm{XL}}{\alpha, \neg\alpha \Rightarrow \beta}\,\mathrm{WR}}{\neg\alpha \Rightarrow \alpha \to \beta}\,{\to}\mathrm{R} \qquad \dfrac{\dfrac{\dfrac{\beta \Rightarrow \beta}{\alpha, \beta \Rightarrow \beta}\,\mathrm{WL}}{\beta \Rightarrow \alpha \to \beta}\,{\to}\mathrm{R}}{}$$

□

4.11 Derivability and the Quantifiers

THEOREM 411A If c is a constant not occurring in Γ or $\alpha(x)$ and $\Gamma \vdash \alpha(c)$, then $\Gamma \vdash \forall x\, \alpha(x)$.

Proof. Let π_0 be an **LK**-derivation of $\Gamma_0 \Rightarrow \alpha(c)$ for some finite $\Gamma_0 \subseteq \Gamma$. By adding a $\forall$R inference, we obtain a proof of $\Gamma_0 \Rightarrow \forall x\, \alpha(x)$, since c does not occur in Γ or $\alpha(x)$ and thus the eigenvariable condition is satisfied. □

THEOREM 411B $\forall x\, \alpha(x) \vdash \alpha(t)$.

Proof. The sequent $\forall x\, \alpha(x) \Rightarrow \alpha(t)$ is derivable:

$$\dfrac{\alpha(t) \Rightarrow \alpha(t)}{\forall x\, \alpha(x) \Rightarrow \alpha(t)}\,\forall\mathrm{L}$$

□

4.12 Soundness

A derivation system, such as the sequent calculus, is *sound* if it cannot derive things that do not actually hold. Soundness is thus a kind of guaranteed safety property for derivation systems. Depending on which proof theoretic property is in question, we would like to know for instance, that

1. every derivable sentence is valid;
2. if a sentence is derivable from some others, it is also a consequence of them;
3. if a set of sentences is inconsistent, it is unsatisfiable.

These are important properties of a derivation system. If any of them do not hold, the derivation system is deficient—it would derive too much. Consequently, establishing the soundness of a derivation system is of the utmost importance.

Because all these proof-theoretic properties are defined via derivability in the sequent calculus of certain sequents, proving (1)–(3) above requires proving something about the semantic properties of derivable sequents. We will first define what it means for a sequent to be *valid*, and then show that every derivable sequent is valid. (1)–(3) then follow as corollaries from this result.

DEFINITION 412A A structure $\mathfrak{A}$ *satisfies* a sequent $\Gamma \Rightarrow \Delta$ iff either $\not\models_{\mathfrak{A}} \alpha$ for some $\alpha \in \Gamma$ or $\models_{\mathfrak{A}} \alpha$ for some $\alpha \in \Delta$.

A sequent is *valid* iff every structure $\mathfrak{A}$ satisfies it.

THEOREM 412B (SOUNDNESS) If **LK** derives $\Theta \Rightarrow \Xi$, then $\Theta \Rightarrow \Xi$ is valid.

Proof. Let π be a derivation of $\Theta \Rightarrow \Xi$. We proceed by induction on the number of inferences n in π.

If the number of inferences is 0, then π consists only of an initial sequent. Every initial sequent $\alpha \Rightarrow \alpha$ is obviously valid, since for every $\mathfrak{A}$, either $\not\models_{\mathfrak{A}} \alpha$ or $\models_{\mathfrak{A}} \alpha$.

If the number of inferences is greater than 0, we distinguish cases according to the type of the lowermost inference. By induction hypothesis, we can assume that the premises of that inference are valid, since the number of inferences in the proof of any premise is smaller than n.

First, we consider the possible inferences with only one premise.

1. The last inference is a weakening. Then $\Theta \Rightarrow \Xi$ is either $A, \Gamma \Rightarrow \Delta$ (if the last inference is WL) or $\Gamma \Rightarrow \Delta, \alpha$ (if it's WR), and the derivation ends in one of

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma \Rightarrow \Delta\end{array}}{\alpha, \Gamma \Rightarrow \Delta}\,\mathrm{WL} \qquad \dfrac{\begin{array}{c}\vdots\\ \Gamma \Rightarrow \Delta\end{array}}{\Gamma \Rightarrow \Delta, \alpha}\,\mathrm{WR}$$

   By induction hypothesis, $\Gamma \Rightarrow \Delta$ is valid, i.e., for every structure $\mathfrak{A}$, either there is some $\gamma \in \Gamma$ such that $\not\models_{\mathfrak{A}} \gamma$ or there is some $\gamma \in \Delta$ such that $\models_{\mathfrak{A}} \gamma$.

   If $\not\models_{\mathfrak{A}} \gamma$ for some $\gamma \in \Gamma$, then $\gamma \in \Theta$ as well since $\Theta = \alpha, \Gamma$, and so $\not\models_{\mathfrak{A}} \gamma$ for some $\gamma \in \Theta$. Similarly, if $\models_{\mathfrak{A}} \gamma$ for some $\gamma \in \Delta$, as $\gamma \in \Xi$, $\models_{\mathfrak{A}} \gamma$ for some $\gamma \in \Xi$. Consequently, $\Theta \Rightarrow \Xi$ is valid.

2. The last inference is ¬L: Then the premise of the last inference is $\Gamma \Rightarrow \Delta, \alpha$ and the conclusion is $\neg\alpha, \Gamma \Rightarrow \Delta$, i.e., the derivation ends in

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma \Rightarrow \Delta, \alpha\end{array}}{\neg\alpha, \Gamma \Rightarrow \Delta}\,\neg\mathrm{L}$$

   and $\Theta = \neg\alpha, \Gamma$ while $\Xi = \Delta$.

   The induction hypothesis tells us that $\Gamma \Rightarrow \Delta, \alpha$ is valid, i.e., for every $\mathfrak{A}$, either (a) for some $\gamma \in \Gamma$, $\not\models_{\mathfrak{A}} \gamma$, or (b) for some $\gamma \in \Delta$, $\models_{\mathfrak{A}} \gamma$, or (c) $\models_{\mathfrak{A}} \alpha$. We want to show that $\Theta \Rightarrow \Xi$ is also valid. Let $\mathfrak{A}$ be a structure. If (a) holds, then there is $\gamma \in \Gamma$ so that $\not\models_{\mathfrak{A}} \alpha$, but $\alpha \in \Theta$ as well. If

(b) holds, there is $\gamma \in \Delta$ such that $\models_{\mathfrak{A}} \gamma$, but $\gamma \in \Xi$ as well. Finally, if $\models_{\mathfrak{A}} \alpha$, then $\not\models_{\mathfrak{A}} \neg\alpha$. Since $\neg\alpha \in \Theta$, there is $\gamma \in \Theta$ such that $\not\models_{\mathfrak{A}} \gamma$. Consequently, $\Theta \Rightarrow \Xi$ is valid.

3. The last inference is ¬R: Exercise.

4. The last inference is ∧L: There are two variants: $\alpha \wedge \beta$ may be inferred on the left from α or from β on the left side of the premise. In the first case, the π ends in

$$\dfrac{\begin{array}{c}\vdots\\ \alpha, \Gamma \Rightarrow \Delta\end{array}}{\alpha \wedge \beta, \Gamma \Rightarrow \Delta}\ \wedge\mathrm{L}$$

and $\Theta = \alpha \wedge \beta, \Gamma$ while $\Xi = \Delta$. Consider a structure $\mathfrak{A}$. Since by induction hypothesis, $\alpha, \Gamma \Rightarrow \Delta$ is valid, (a) $\not\models_{\mathfrak{A}} \alpha$, (b) $\not\models_{\mathfrak{A}} \gamma$ for some $\gamma \in \Gamma$, or (c) $\models_{\mathfrak{A}} \gamma$ for some $\gamma \in \Delta$. In case (a), $\not\models_{\mathfrak{A}} \alpha \wedge \beta$, so there is $\gamma \in \Theta$ (namely, $\alpha \wedge \beta$) such that $\not\models_{\mathfrak{A}} \gamma$. In case (b), there is $\gamma \in \Gamma$ such that $\not\models_{\mathfrak{A}} \phi$, and $\gamma \in \Theta$ as well. In case (c), there is $\gamma \in \Delta$ such that $\models_{\mathfrak{A}} \gamma$, and $\gamma \in \Xi$ as well since $\Xi = \Delta$. So in each case, $\mathfrak{A}$ satisfies $\alpha \wedge \beta, \Gamma \Rightarrow \Delta$. Since $\mathfrak{A}$ was arbitrary, $\Gamma \Rightarrow \Delta$ is valid. The case where $\alpha \wedge \beta$ is inferred from β is handled the same, changing α to β.

5. The last inference is ∨R: There are two variants: $\alpha \vee \beta$ may be inferred on the right from α or from β on the right side of the premise. In the first case, π ends in

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma \Rightarrow \Delta, \alpha\end{array}}{\Gamma \Rightarrow \Delta, \alpha \vee \beta}\ \vee\mathrm{R}$$

Now $\Theta = \Gamma$ and $\Xi = \Delta, \alpha \vee \beta$. Consider a structure $\mathfrak{A}$. Since $\Gamma \Rightarrow \Delta, \alpha$ is valid, (a) $\models_{\mathfrak{A}} \alpha$, (b) $\not\models_{\mathfrak{A}} \gamma$ for some $\gamma \in \Gamma$, or (c) $\models_{\mathfrak{A}} \gamma$ for some $\gamma \in \Delta$. In case (a), $\models_{\mathfrak{A}} \alpha \vee \beta$. In case (b), there is $\gamma \in \Gamma$ such that $\not\models_{\mathfrak{A}} \gamma$. In case (c), there is $\gamma \in \Delta$ such that $\models_{\mathfrak{A}} \gamma$. So in each case, $\mathfrak{A}$ satisfies $\Gamma \Rightarrow \Delta, \alpha \vee \beta$, i.e., $\Theta \Rightarrow \Xi$. Since $\mathfrak{A}$ was arbitrary, $\Theta \Rightarrow \Xi$ is valid. The case where $\alpha \vee \beta$ is inferred from β is handled the same, changing α to β.

6. The last inference is →R: Then π ends in

$$\dfrac{\begin{array}{c}\vdots\\ \alpha, \Gamma \Rightarrow \Delta, \alpha\end{array}}{\Gamma \Rightarrow \Delta, \alpha \to \beta}\ \to\mathrm{R}$$

Again, the induction hypothesis says that the premise is valid; we want to show that the conclusion is valid as well. Let $\mathfrak{A}$ be arbitrary. Since $\alpha, \Gamma \Rightarrow \Delta, \beta$ is valid, at least one of the following cases obtains: (a) $\not\models_{\mathfrak{A}} \alpha$, (b) $\models_{\mathfrak{A}} \beta$, (c) $\not\models_{\mathfrak{A}} \gamma$ for some $\gamma \in \Gamma$, or (c) $\models_{\mathfrak{A}} \gamma$ for some $\gamma \in \Delta$. In cases (a) and (b), $\models_{\mathfrak{A}} \alpha \to \beta$ and so there is a $\gamma \in \Delta, \alpha \to \beta$ such that $\models_{\mathfrak{A}} \gamma$. In case (c), for some $\gamma \in \Gamma$, $\not\models_{\mathfrak{A}} \gamma$. In case (d), for some $\gamma \in \Delta$, $\models_{\mathfrak{A}} \gamma$. In each case, $\mathfrak{A}$ satisfies $\Gamma \Rightarrow \Delta, \alpha \to \beta$. Since $\mathfrak{A}$ was arbitrary, $\Gamma \Rightarrow \Delta, \alpha \to \beta$ is valid.

7. The last inference is $\forall$L: Then there is a wff $\alpha(x)$ and a closed term t such that π ends in

$$\dfrac{\begin{array}{c}\vdots\\ \alpha(t), \Gamma \Rightarrow \Delta\end{array}}{\forall x\, \alpha(x), \Gamma \Rightarrow \Delta}\,\forall\text{L}$$

We want to show that the conclusion $\forall x\, \alpha(x), \Gamma \Rightarrow \Delta$ is valid. Consider a structure $\mathfrak{A}$. Since the premise $\alpha(t), \Gamma \Rightarrow \Delta$ is valid, (a) $\not\models_{\mathfrak{A}} \alpha(t)$, (b) $\not\models_{\mathfrak{A}} \gamma$ for some $\gamma \in \Gamma$, or (c) $\models_{\mathfrak{A}} \gamma$ for some $\gamma \in \Delta$. In case (a), by Proposition 114H, if $\models_{\mathfrak{A}} \forall x\, \alpha(x)$, then $\models_{\mathfrak{A}} \alpha(t)$. Since $\not\models_{\mathfrak{A}} \alpha(t)$, $\not\models_{\mathfrak{A}} \forall x\, \alpha(x)$. In case (b) and (c), $\mathfrak{A}$ also satisfies $\forall x\, \alpha(x), \Gamma \Rightarrow \Delta$. Since $\mathfrak{A}$ was arbitrary, $\forall x\, \alpha(x), \Gamma \Rightarrow \Delta$ is valid.

8. The last inference is $\exists$R: Exercise.

9. The last inference is $\forall$R: Then there is a wff $\alpha(x)$ and a constant symbol a such that π ends in

$$\dfrac{\begin{array}{c}\vdots\\ \Gamma \Rightarrow \Delta, \alpha(a)\end{array}}{\Gamma \Rightarrow \Delta, \forall x\, \alpha(x)}\,\forall\text{R}$$

where the eigenvariable condition is satisfied, i.e., a does not occur in $\alpha(x)$, Γ, or Δ. By induction hypothesis, the premise of the last inference is valid. We have to show that the conclusion is valid as well, i.e., that for any structure $\mathfrak{A}$, (a) $\models_{\mathfrak{A}} \forall x\, \alpha(x)$, (b) $\not\models_{\mathfrak{A}} \gamma$ for some $\gamma \in \Gamma$, or (c) $\models_{\mathfrak{A}} \gamma$ for some $\gamma \in \Delta$.

Suppose $\mathfrak{A}$ is an arbitrary structure. If (b) or (c) holds, we are done, so suppose neither holds: for all $\gamma \in \Gamma$, $\models_{\mathfrak{A}} \gamma$, and for all $\gamma \in \Delta$, $\not\models_{\mathfrak{A}} \gamma$. We have to show that (a) holds, i.e., $\models_{\mathfrak{A}} \forall x\, \alpha(x)$. By Proposition 112F, if suffices to show that $\models_{\mathfrak{A}} \alpha(x)[s]$ for all variable assignments s. So let s be an arbitrary variable assignment. Consider the structure $\mathfrak{M}'$ which is just like $\mathfrak{A}$ except $a^{\mathfrak{A}'} = s(x)$. By corollary 1.13.2, for any $\gamma \in \Gamma$,

$\models_{\mathfrak{A}'} \gamma$ since a does not occur in Γ, and for any $\gamma \in \Delta$, $\not\models_{\mathfrak{A}'} \gamma$. But the premise is valid, so $\models_{\mathfrak{A}'} \alpha(a)$. By Proposition 112E, $\models_{\mathfrak{A}'} \alpha(a)[s]$, since $\alpha(a)$ is a sentence. Now $s \sim_x s$ with $s(x) = \bar{s}(a)$, since we've defined $\mathfrak{M}'$ in just this way. So Proposition 113D applies, and we get $\models_{\mathfrak{A}'} \alpha(x)[s]$. Since a does not occur in $\alpha(x)$, by Proposition 113A, $\models_{\mathfrak{A}} \alpha(x)[s]$. Since s was arbitrary, we've completed the proof that $\models_{\mathfrak{A}} \alpha(x)[s]$ for all variable assignments.

10. The last inference is ∃L: Exercise.

Now let's consider the possible inferences with two premises.

1. The last inference is a cut: then π ends in

$$\dfrac{\begin{array}{cc}\vdots & \vdots \\ \Gamma \Rightarrow \Delta, \alpha & \alpha, \Pi \Rightarrow \Lambda\end{array}}{\Gamma, \Pi \Rightarrow \Delta, \Lambda}\ \text{Cut}$$

   Let $\mathfrak{A}$ be a structure. By induction hypothesis, the premises are valid, so $\mathfrak{A}$ satisfies both premises. We distinguish two cases: (a) $\not\models_{\mathfrak{A}} \alpha$ and (b) $\models_{\mathfrak{A}} \alpha$. In case (a), in order for $\mathfrak{A}$ to satisfy the left premise, it must satisfy $\Gamma \Rightarrow \Delta$. But then it also satisfies the conclusion. In case (b), in order for $\mathfrak{A}$ to satisfy the right premise, it must satisfy $\Pi \setminus \Lambda$. Again, $\mathfrak{A}$ satisfies the conclusion.

2. The last inference is ∧R. Then π ends in

$$\dfrac{\begin{array}{cc}\vdots & \vdots \\ \Gamma \Rightarrow \Delta, \alpha & \Gamma \Rightarrow \Delta, \beta\end{array}}{\Gamma \Rightarrow \Delta, \alpha \wedge \beta}\ \wedge\text{R}$$

   Consider a structure $\mathfrak{A}$. If $\mathfrak{A}$ satisfies $\Gamma \Rightarrow \Delta$, we are done. So suppose it doesn't. Since $\Gamma \Rightarrow \Delta, \alpha$ is valid by induction hypothesis, $\models_{\mathfrak{A}} \alpha$. Similarly, since $\Gamma \Rightarrow \Delta, \beta$ is valid, $\models_{\mathfrak{A}} \beta$. But then $\models_{\mathfrak{A}} \alpha \wedge \beta$.

3. The last inference is ∨L: Exercise.

4. The last inference is →L. Then π ends in

$$\dfrac{\begin{array}{cc}\vdots & \vdots \\ \Gamma \Rightarrow \Delta, \alpha & \beta, \Pi \Rightarrow \Lambda\end{array}}{\alpha \to \beta, \Gamma, \Pi \Rightarrow \Delta, \Lambda}\ \to\text{L}$$

Again, consider a structure $\mathfrak{A}$ and suppose $\mathfrak{A}$ doesn't satisfy $\Gamma, \Pi \Rightarrow \Lambda, \Pi$. We have to show that $\not\models_{\mathfrak{A}} \alpha \to \beta$. If $\mathfrak{A}$ doesn't satisfy $\Gamma, \Pi \Rightarrow \Lambda, \Pi$, it satisfies neither $\Gamma \Rightarrow \Delta$ nor $\Pi \Rightarrow \Lambda$. Since, $\Gamma \Rightarrow \Delta, \alpha$ is valid, we have $\models_{\mathfrak{A}} \alpha$. Since $\beta, \Pi \Rightarrow \Lambda$ is valid, we have $\not\models_{\mathfrak{A}} \beta$. But then $\not\models_{\mathfrak{A}} \alpha \to \beta$, which is what we wanted to show.

□

COROLLARY 4.12.3 If $\vdash \alpha$ then α is valid.

COROLLARY 4.12.4 If $\Gamma \vdash \alpha$ then $\Gamma \vDash \alpha$.

Proof. If $\Gamma \vdash \alpha$ then for some finite subset $\Gamma_0 \subseteq \Gamma$, there is a derivation of $\Gamma_0 \Rightarrow \alpha$. By Theorem 412B, every structure $\mathfrak{A}$ either makes some $\beta \in \Gamma_0$ false or makes α true. Hence, if $\models_{\mathfrak{A}} \Gamma$ then also $\models_{\mathfrak{A}} \alpha$. □

COROLLARY 4.12.5 If Γ is satisfiable, then it is consistent.

Proof. We prove the contrapositive. Suppose that Γ is not consistent. Then there is a finite $\Gamma_0 \subseteq \Gamma$ and a derivation of $\Gamma_0 \Rightarrow \quad$. By Theorem 412B, $\Gamma_0 \Rightarrow$ is valid. In other words, for every structure $\mathfrak{A}$, there is $\gamma \in \Gamma_0$ so that $\not\models_{\mathfrak{A}} \gamma$, and since $\Gamma_0 \subseteq \Gamma$, that γ is also in Γ. Thus, no $\mathfrak{A}$ satisfies Γ, and Γ is not satisfiable. □

4.13 Derivations with Equality symbol

Derivations with equality symbol require additional initial sequents and inference rules.

DEFINITION 413A (INITIAL SEQUENTS FOR =) If t is a closed term, then $\Rightarrow t = t$ is an initial sequent.

The rules for = are (t_1 and t_2 are closed terms):

$$\frac{t_1 = t_2, \Gamma \Rightarrow \Delta, \alpha(t_1)}{t_1 = t_2, \Gamma \Rightarrow \Delta, \alpha(t_2)} = \qquad \frac{t_1 = t_2, \Gamma \Rightarrow \Delta, \alpha(t_2)}{t_1 = t_2, \Gamma \Rightarrow \Delta, \alpha(t_1)} =$$

Example 4.13.2. If s and t are closed terms, then $s = t, \alpha(s) \vdash \alpha(t)$:

$$\dfrac{\dfrac{\alpha(s) \Rightarrow \alpha(s)}{s = t, \alpha(s) \Rightarrow \alpha(s)}\,\mathrm{WL}}{s = t, \alpha(s) \Rightarrow \alpha(t)}\,=$$

This may be familiar as the principle of substitutability of identicals, or Leibniz' Law.

LK proves that = is symmetric and transitive:

$$\dfrac{\dfrac{\Rightarrow t_1 = t_1}{t_1 = t_2 \Rightarrow t_1 = t_1}\,\mathrm{WL}}{t_1 = t_2 \Rightarrow t_2 = t_1}\,= \qquad \dfrac{\dfrac{t_1 = t_2 \Rightarrow t_1 = t_2}{t_2 = t_3, t_1 = t_2 \Rightarrow t_1 = t_2}\,\mathrm{WL}}{\dfrac{t_2 = t_3, t_1 = t_2 \Rightarrow t_1 = t_3}{t_1 = t_2, t_2 = t_3 \Rightarrow t_1 = t_3}\,\mathrm{XL}}\,=$$

In the proof on the left, the wff $x = t_1$ is our $\alpha(x)$. On the right, we take $\alpha(x)$ to be $t_1 = x$.

4.14 Soundness with Equality symbol

PROPOSITION 414A **LK** with initial sequents and rules for identity is sound.

Proof. Initial sequents of the form $\Rightarrow t = t$ are valid, since for every structure $\mathfrak{A}$, $\models_{\mathfrak{A}} t = t$. (Note that we assume the term t to be closed, i.e., it contains no variables, so variable assignments are irrelevant).

Suppose the last inference in a derivation is $=$. Then the premise is $t_1 = t_2, \Gamma \Rightarrow \Delta, \alpha(t_1)$ and the conclusion is $t_1 = t_2, \Gamma \Rightarrow \Delta, \alpha(t_2)$. Consider a structure $\mathfrak{A}$. We need to show that the conclusion is valid, i.e., if $\models_{\mathfrak{A}} t_1 = t_2$ and $\models_{\mathfrak{A}} \Gamma$, then either $\models_{\mathfrak{A}} \gamma$ for some $\gamma \in \Delta$ or $\models_{\mathfrak{A}} \alpha(t_2)$.

By induction hypothesis, the premise is valid. This means that if $\models_{\mathfrak{A}} t_1 = t_2$ and $\models_{\mathfrak{A}} \Gamma$ either (a) for some $\gamma \in \Delta$, $\models_{\mathfrak{A}} \gamma$ or (b) $\models_{\mathfrak{A}} \alpha(t_1)$. In case (a) we are done. Consider case (b). Let s be a variable assignment with $s(x) = {t_1}^{\mathfrak{A}}$. By Proposition 112E, $\models_{\mathfrak{A}} \alpha(t_1)[s]$. Since $s \sim_x s$, by Proposition 113D, $\models_{\mathfrak{A}} \alpha(x)[s]$. since $\models_{\mathfrak{A}} t_1 = t_2$, we have ${t_1}^{\mathfrak{A}} = {t_2}^{\mathfrak{A}}$, and hence $s(x) = {t_2}^{\mathfrak{A}}$. By applying Proposition 113D again, we also have $\models_{\mathfrak{A}} \alpha(t_2)[s]$. By Proposition 112E, $\models_{\mathfrak{A}} \alpha(t_2)$. □

Chapter 5

Natural Deduction

5.1 Rules and Derivations

Natural deduction systems are meant to closely parallel the informal reasoning used in mathematical proof (hence it is somewhat "natural"). Natural deduction proofs begin with assumptions. Inference rules are then applied. Assumptions are "discharged" by the ¬Intro, →Intro, ∨Elim and ∃Elim inference rules, and the label of the discharged assumption is placed beside the inference for clarity.

Definition 51A (Initial Wff) An *initial wff* or *assumption* is any wff in the topmost position of any branch.

Derivations in natural deduction are certain trees of sentences, where the topmost sentences are assumptions, and if a sentence stands below one, two, or three other sequents, it must follow correctly by a rule of inference. The sentences at the top of the inference are called the *premises* and the sentence below the *conclusion* of the inference. The rules come in pairs, an introduction and an elimination rule for each logical operator. They introduce a logical operator in the conclusion or remove a logical operator from a premise of the rule. Some of the rules allow an assumption of a certain type to be *discharged*. To indicate which assumption is discharged by which inference, we also assign labels to both the assumption and the inference. This is indicated by writing the assumption as "$[\alpha]^n$".

It is customary to consider rules for all logical operators, even for those (if any) that we consider as defined.

5.2 Propositional Rules

Rules for $\wedge$

$$\frac{\alpha \qquad \beta}{\alpha \wedge \beta}\ \wedge\text{Intro}$$

$$\frac{\alpha \wedge \beta}{\alpha}\ \wedge\text{Elim} \qquad \frac{\alpha \wedge \beta}{\beta}\ \wedge\text{Elim}$$

Rules for $\vee$

$$\frac{\alpha}{\alpha \vee \beta}\ \vee\text{Intro} \qquad \frac{\beta}{\alpha \vee \beta}\ \vee\text{Intro}$$

$$n\ \frac{\alpha \vee \beta \qquad \begin{array}{c}[\alpha]^n \\ \vdots \\ \gamma\end{array} \qquad \begin{array}{c}[\beta]^n \\ \vdots \\ \gamma\end{array}}{\gamma}\ \vee\text{Elim}$$

Rules for $\rightarrow$

$$n\ \frac{\begin{array}{c}[\alpha]^n \\ \vdots \\ \beta\end{array}}{\alpha \rightarrow \beta}\ \rightarrow\text{Intro}$$

$$\frac{\alpha \rightarrow \beta \qquad \alpha}{\beta}\ \rightarrow\text{Elim}$$

Rules for $\neg$

$$n\ \frac{\begin{array}{c}[\alpha]^n \\ \vdots \\ \bot\end{array}}{\neg\alpha}\ \neg\text{Intro}$$

$$\frac{\neg\alpha \qquad \alpha}{\bot}\ \neg\text{Elim}$$

Rules for $\bot$

$$\frac{\bot}{\alpha}\ \bot_I$$

$$n\ \frac{\begin{array}{c}[\neg\alpha]^n \\ \vdots \\ \bot\end{array}}{\alpha}\ \bot_C$$

Note that $\neg$Intro and $\bot_C$ are very similar: The difference is that $\neg$Intro derives a negated sentence $\neg\alpha$ but $\bot_C$ a positive sentence α.

5.3 Quantifier Rules

Rules for ∀

$$\frac{\alpha(a)}{\forall x\, \alpha x}\,\forall\text{Intro} \qquad\qquad \frac{\forall x\, \alpha x}{\alpha(t)}\,\forall\text{Elim}$$

In the rules for ∀, t is a ground term (a term that does not contain any variables), and a is a constant symbol which does not occur in the conclusion $\forall x\, \alpha(x)$, or in any assumption which is undischarged in the derivation ending with the premise $\alpha(a)$. We call a the *eigenvariable* of the ∀Intro inference.

Rules for ∃

$$\frac{\alpha t}{\exists x\, \alpha x}\,\exists\text{Intro} \qquad\qquad {}_{n}\frac{\exists x\, \alpha x \qquad \begin{array}{c}[\alpha a]^n\\ \vdots\\ \gamma\end{array}}{\gamma}\,\exists\text{Elim}$$

Again, t is a ground term, and a is a constant which does not occur in the premise $\exists x\, \alpha(x)$, in the conclusion γ, or any assumption which is undischarged in the derivations ending with the two premises (other than the assumptions $\alpha(a)$). We call a the *eigenvariable* of the ∃Elim inference.

The condition that an eigenvariable neither occur in the premises nor in any assumption that is undischarged in the derivations leading to the premises for the ∀Intro or ∃Elim inference is called the *eigenvariable condition*.

We use the term "eigenvariable" even though a in the above rules is a constant. This has historical reasons.

In ∃Intro and ∀Elim there are no restrictions, and the term t can be anything, so we do not have to worry about any conditions. On the other hand, in the ∃Elim and ∀Intro rules, the eigenvariable condition requires that the constant symbol a does not occur anywhere in the conclusion or in an undischarged assumption. The condition is necessary to ensure that the system is sound, i.e., only derives sentences from undischarged assumptions from which the follow. Without this condition, the following would be allowed:

$$\frac{\exists x\, \alpha(x) \qquad \dfrac{[\alpha(a)]^1}{\forall x\, \alpha(x)}\,{*}\forall\text{Intro}}{\forall x\, \alpha(x)}\,\exists\text{Elim}$$

However, $\exists x\, \alpha(x) \nvDash \forall x\, \alpha(x)$.

5.4 Derivations

We've said what an assumption is, and we've given the rules of inference. Derivations in the sequent calculus are inductively generated from these: each

derivation either is an assumption on its own, or consists of one, two, or three derivations followed by a correct inference.

DEFINITION 54A (DERIVATION) A *derivation* of a sentence α from assumptions Γ is a tree of sentences satisfying the following conditions:

1. The topmost sentences of the tree are either in Γ or are discharged by an inference in the tree.

2. The bottommost sentence of the tree is α.

3. Every sentence in the tree except α is a premise of a correct application of am inference rule whose conclusion stands directly below that sentence in the tree.

We then say that α is the *conclusion* of the derivation and that α is *derivable* from Γ.

Example 5.4.2. Every assumption on its own is a derivation. So, e.g., γ by itself is a derivation, and so is δ by itself. We can obtain a new derivation from these by applying, say, the $\wedge$Intro rule,

$$\dfrac{\alpha \qquad \beta}{\alpha \wedge \beta}\,\wedge\text{Intro}$$

These rules are meant to be general: we can replace the α and β in it with any sentences, e.g., by γ and δ. Then the conclusion would be $\gamma \wedge \delta$, and so

$$\dfrac{\gamma \qquad \delta}{\gamma \wedge \delta}\,\wedge\text{Intro}$$

is a correct derivation. Of course, we can also switch the assumptions, so that δ plays the role of α and γ that of β. Thus,

$$\dfrac{\delta \qquad \gamma}{\delta \wedge \gamma}\,\wedge\text{Intro}$$

is also a correct derivation.

We can now apply another rule, say, $\rightarrow$Intro, which allows us to conclude a conditional and allows us to discharge any assumption that is identical to the conclusion of that conditional. So both of the following would be correct derivations:

$$1\,\dfrac{\dfrac{[\gamma]^1 \qquad \delta}{\gamma \wedge \delta}\,\wedge\text{Intro}}{\gamma \rightarrow (\gamma \wedge \delta)}\,\rightarrow\text{Intro} \qquad 1\,\dfrac{\dfrac{\gamma \qquad [\delta]^1}{\gamma \wedge \delta}\,\wedge\text{Intro}}{\delta \rightarrow (\gamma \wedge \delta)}\,\rightarrow\text{Intro}$$

5.5 Examples of Derivations

Example 5.5.1. Let's give a derivation of the sentence $(\alpha \wedge \beta) \to \alpha$.

We begin by writing the desired conclusion at the bottom of the derivation.

$$\frac{}{(\alpha \wedge \beta) \to \alpha}$$

Next, we need to figure out what kind of inference could result in a sentence of this form. The main operator of the conclusion is $\to$, so we'll try to arrive at the conclusion using the $\to$Intro rule. It is best to write down the assumptions involved and label the inference rules as you progress, so it is easy to see whether all assumptions have been discharged at the end of the proof.

$$\begin{array}{c} [\alpha \wedge \beta]^1 \\ \vdots \\ \alpha \\ \hline (\alpha \wedge \beta) \to \alpha \end{array} {}^{1}\,\to\text{Intro}$$

We now need to fill in the steps from the assumption $\alpha \wedge \beta$ to α. Since we only have one connective to deal with, $\wedge$, we must use the $\wedge$ elim rule. This gives us the following proof:

$$\dfrac{\dfrac{[\alpha \wedge \beta]^1}{\alpha}\,\wedge\text{Elim}}{(\alpha \wedge \beta) \to \alpha}\,{}^{1}\,\to\text{Intro}$$

We now have a correct derivation of $(\alpha \wedge \beta) \to \alpha$.

Example 5.5.2. Now let's give a derivation of $(\neg\alpha \vee \beta) \to (\alpha \to \beta)$.

We begin by writing the desired conclusion at the bottom of the derivation.

$$\frac{}{(\neg\alpha \vee \beta) \to (\alpha \to \beta)}$$

To find a logical rule that could give us this conclusion, we look at the logical connectives in the conclusion: $\neg$, $\vee$, and $\to$. We only care at the moment about the first occurence of $\to$ because it is the main operator of the sentence in the end-sequent, while $\neg$, $\vee$ and the second occurence of $\to$ are inside the scope of another connective, so we will take care of those later. We therefore start with the $\to$Intro rule. A correct application must look as follows:

$$\begin{array}{c} [\neg\alpha \vee \beta]^1 \\ \vdots \\ \alpha \to \beta \\ \hline (\neg\alpha \vee \beta) \to (\alpha \to \beta) \end{array} {}^{1}\,\to\text{Intro}$$

This leaves us with two possibilities to continue. Either we can keep working from the bottom up and look for another application of the $\to$Intro rule, or we

can work from the top down and apply a $\vee$Elim rule. Let us apply the latter. We will use the assumption $\neg\alpha \vee \beta$ as the leftmost premise of $\vee$Elim. For a valid application of $\vee$Elim, the other two premises must be identical to the conclusion $\alpha \to \beta$, but each may be derived in turn from another assumption, namely the two disjuncts of $\neg\alpha \vee \beta$. So our derivation will look like this:

$$\dfrac{\dfrac{[\neg\alpha \vee \beta]^1 \qquad \begin{array}{c}[\neg\alpha]^2\\ \vdots\\ \alpha \to \beta\end{array} \qquad \begin{array}{c}[\beta]^2\\ \vdots\\ \alpha \to \beta\end{array}}{\alpha \to \beta}\ \vee\text{Elim}^{\,2}}{(\neg\alpha \vee \beta) \to (\alpha \to \beta)}\ \to\text{Intro}^{\,1}$$

In each of the two branches on the right, we want to derive $\alpha \to \beta$, which is best done using $\to$Intro.

$$\dfrac{\dfrac{[\neg\alpha \vee \beta]^1 \qquad \dfrac{\begin{array}{c}[\neg\alpha]^2, [\alpha]^3\\ \vdots\\ \beta\end{array}}{\alpha \to \beta}\ \to\text{Intro}^{\,3} \qquad \dfrac{\begin{array}{c}[\beta]^2, [\alpha]^4\\ \vdots\\ \beta\end{array}}{\alpha \to \beta}\ \to\text{Intro}^{\,4}}{\alpha \to \beta}\ \vee\text{Elim}^{\,2}}{(\neg\alpha \vee \beta) \to (\alpha \to \beta)}\ \to\text{Intro}^{\,1}$$

For the two missing parts of the derivation, we need derivations of β from $\neg\alpha$ and α in the middle, and from α and β on the left. Let's take the former first. $\neg\alpha$ and α are the two premises of $\neg$Elim:

$$\dfrac{\begin{array}{c}\dfrac{[\neg\alpha]^2 \qquad [\alpha]^3}{\bot}\ \neg\text{Elim}\\ \vdots\\ \beta\end{array}}{}$$

By using $\bot_I$, we can obtain β as a conclusion and complete the branch.

$$\dfrac{\dfrac{[\neg\alpha \vee \beta]^1 \qquad \dfrac{\dfrac{\dfrac{[\neg\alpha]^2 \qquad [\alpha]^3}{\bot}\ \bot\text{Intro}}{\beta}\ \bot_I}{\alpha \to \beta}\ \to\text{Intro}^{\,3} \qquad \dfrac{\begin{array}{c}[\beta]^2, [\alpha]^4\\ \vdots\\ \beta\end{array}}{\alpha \to \beta}\ \to\text{Intro}^{\,4}}{\alpha \to \beta}\ \vee\text{Elim}^{\,2}}{(\neg\alpha \vee \beta) \to (\alpha \to \beta)}\ \to\text{Intro}^{\,1}$$

Let's now look at the rightmost branch. Here it's important to realize that the definition of derivation *allows assumptions to be discharged* but *does*

not require them to be. In other words, if we can derive β from one of the assumptions α and β without using the other, that's ok. And to derive β from β is trivial: β by itself is such a derivation, and no inferences are needed. So we can simply delete the assumtion α.

$$\dfrac{\dfrac{\dfrac{[\neg\alpha\vee\beta]^1 \qquad \dfrac{\dfrac{\dfrac{[\neg\alpha]^2 \qquad [\alpha]^3}{\bot}\neg\text{Elim}}{\beta}\bot_I}{\alpha\to\beta}{}^{3}\to\text{Intro} \qquad \dfrac{[\beta]^2}{\alpha\to\beta}\to\text{Intro}}{\alpha\to\beta}{}^{2}\vee\text{Elim}}{(\neg\alpha\vee\beta)\to(\alpha\to\beta)}{}^{1}\to\text{Intro}$$

Note that in the finished derivation, the rightmost $\to$Intro inference does not actually discharge any assumptions.

Example 5.5.3. So far we have not needed the $\bot_C$ rule. It is special in that it allows us to discharge an assumption that isn't a sub-wff of the conclusion of the rule. It is closely related to the $\bot_I$ rule. In fact, the $\bot_I$ rule is a special case of the $\bot_C$ rule—there is a logic called "intuitionistic logic" in which only $\bot_I$ is allowed. The $\bot_C$ rule is a last resort when nothing else works. For instance, suppose we want to derive $\alpha\vee\neg\alpha$. Our usual strategy would be to attempt to derive $\alpha\vee\neg\alpha$ using $\vee$Intro. But this would require us to derive either α or $\neg\alpha$ from no assumptions, and this can't be done. $\bot_C$ to the rescue!

$$\dfrac{\dfrac{[\neg(\alpha\vee\neg\alpha)]^1}{\vdots}}{\alpha\vee\neg\alpha}{}^{1}\bot_C$$

Now we're looking for a derivation of $\bot$ from $\neg(\alpha\vee\neg\alpha)$. Since $\bot$ is the conclusion of $\neg$Elim we might try that:

$$\dfrac{\dfrac{\dfrac{[\neg(\alpha\vee\neg\alpha)]^1}{\vdots}}{\neg\alpha} \qquad \dfrac{[\neg(\alpha\vee\neg\alpha)]^1}{\vdots}}{\alpha}\neg\text{Elim}}{\alpha\vee\neg\alpha}{}^{1}\bot_C$$

Our strategy for finding a derivation of $\neg\alpha$ calls for an application of $\neg$Intro:

$$\dfrac{\dfrac{\dfrac{\dfrac{[\neg(\alpha\vee\neg\alpha)]^1,[\alpha]^2}{\vdots}}{\bot}}{\neg\alpha}{}^{2}\neg\text{Intro} \qquad \dfrac{[\neg(\alpha\vee\neg\alpha)]^1}{\vdots}}{\alpha}\neg\text{Elim}}{\alpha\vee\neg\alpha}{}^{1}\bot_C$$

Here, we can get $\bot$ easily by applying $\neg$Elim to the assumption $\neg(\alpha \vee \neg\alpha)$ and $\alpha \vee \neg\alpha$ which follows from our new assumption α by $\vee$Intro:

$$\dfrac{\dfrac{[\neg(\alpha \vee \neg\alpha)]^1 \qquad \dfrac{[\alpha]^2}{\alpha \vee \neg\alpha}\vee\text{Intro}}{\bot}\neg\text{Elim}}{\dfrac{\bot}{\neg\alpha}\neg\text{Intro}^{2}} \qquad \begin{array}{c}[\neg(\alpha \vee \neg\alpha)]^1\\ \vdots \\ \alpha\end{array}\neg\text{Elim}$$
$$\dfrac{\bot}{\alpha \vee \neg\alpha}\bot_C{}^{1}$$

On the right side we use the same strategy, except we get α by $\bot_C$:

$$\dfrac{\dfrac{[\neg(\alpha \vee \neg\alpha)]^1 \qquad \dfrac{[\alpha]^2}{\alpha \vee \neg\alpha}\vee\text{Intro}}{\bot}\neg\text{Elim}}{\dfrac{\bot}{\neg\alpha}\neg\text{Intro}^{2}} \qquad \dfrac{\dfrac{[\neg(\alpha \vee \neg\alpha)]^1 \qquad \dfrac{[\neg\alpha]^3}{\alpha \vee \neg\alpha}\vee\text{Intro}}{\bot}\neg\text{Elim}}{\dfrac{\bot}{\alpha}\bot_C{}^{3}}\neg\text{Elim}$$
$$\dfrac{\bot}{\alpha \vee \neg\alpha}\bot_C{}^{1}$$

5.6 Derivations with Quantifiers

Example 5.6.1. 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 lower down in the finished proof).

Let's see how we'd give a derivation of the wff $\exists x\, \neg\alpha(x) \to \neg\forall x\, \alpha(x)$. Starting as usual, we write

$$\dfrac{}{\exists x\, \neg\alpha(x) \to \neg\forall x\, \alpha(x)}$$

We start by writing down what it would take to justify that last step using the $\to$Intro rule.

$$\dfrac{\begin{array}{c}[\exists x\, \neg\alpha(x)]^1\\ \vdots \\ \neg\forall x\, \alpha(x)\end{array}}{\exists x\, \neg\alpha(x) \to \neg\forall x\, \alpha(x)}\to\text{Intro}$$

Since there is no obvious rule to apply to $\neg\forall x\, \alpha(x)$, we will proceed by setting up the derivation so we can use the $\exists$Elim rule. Here we must pay attention to the eigenvariable condition, and choose a constant that does not appear in $\exists x\, \alpha(x)$ or any assumptions that it depends on. (Since no constant symbols

appear, however, any choice will do fine.)

$$\dfrac{\dfrac{[\exists x\,\neg\alpha(x)]^1 \qquad \begin{array}{c}[\neg\alpha(a)]^2\\ \vdots\\ \neg\forall x\,\alpha(x)\end{array}}{\neg\forall x\,\alpha(x)}\,\exists\text{Elim}^{\,2}}{\exists x\,\neg\alpha(x) \to \neg\forall x\,\alpha(x)}\,{\to}\text{Intro}$$

In order to derive $\neg\forall x\,\alpha(x)$, we will attempt to use the $\neg$Intro rule: this requires that we derive a contradiction, possibly using $\forall x\,\alpha(x)$ as an additional assumption. Of coursem, this contradiction may involve the assumption $\neg\alpha(a)$ which will be discharged by the $\to$Intro inference. We can set it up as follows:

$$\dfrac{\dfrac{[\exists x\,\neg\alpha(x)]^1 \qquad \dfrac{\begin{array}{c}[\neg\alpha(a)]^2, [\forall x\,\alpha(x)]^3\\ \vdots\\ \bot\end{array}}{\neg\forall x\,\alpha(x)}\,\neg\text{Intro}^{\,3}}{\neg\forall x\,\alpha(x)}\,\exists\text{Elim}^{\,2}}{\exists x\,\neg\alpha(x) \to \neg\forall x\,\alpha(x)}\,{\to}\text{Intro}$$

It looks like we are close to getting a contradiction. The easiest rule to apply is the $\forall$Elim, which has no eigenvariable conditions. Since we can use any term we want to replace the universally quantified x, it makes the most sense to continue using a so we can reach a contradiction.

$$\dfrac{\dfrac{[\exists x\,\neg\alpha(x)]^1 \qquad \dfrac{\dfrac{[\neg\alpha(a)]^2 \qquad \dfrac{[\forall x\,\alpha(x)]^3}{\alpha(a)}\,\forall\text{Elim}}{\bot}\,\neg\text{Elim}}{\neg\forall x\,\alpha(x)}\,\neg\text{Intro}^{\,3}}{\neg\forall x\,\alpha(x)}\,\exists\text{Elim}^{\,2}}{\exists x\,\neg\alpha(x) \to \neg\forall x\,\alpha(x)}\,{\to}\text{Intro}$$

It is important, especially when dealing with quantifiers, to double check at this point that the eigenvariable condition has not been violated. Since the only rule we applied that is subject to the eigenvariable condition was $\exists$Elim, and the eigenvariable a does not occur in any assumptions it depends on, this is a correct derivation.

Example 5.6.2. Sometimes we may derive a wff from other wffs. In these cases, we may have undischarged assumptions. It is important to keep track of our assumptions as well as the end goal.

Let's see how we'd give a derivation of the wff $\exists x\,\gamma(x, b)$ from the assumptions $\exists x\,(\alpha(x) \wedge \beta(x))$ and $\forall x\,(\beta(x) \to \gamma(x, b))$. Starting as usual, we write the conclusion at the bottom.

$$\dfrac{}{\exists x\,\gamma(x, b)}$$

We have two premises to work with. To use the first, i.e., try to find a derivation of $\exists x\, \gamma(x, b)$ from $\exists x\, (\alpha(x) \wedge \beta(x))$ we would use the $\exists$Elim rule. Since it has an eigenvariable condition, we will apply that rule first. We get the following:

$$\dfrac{\exists x\, (\alpha(x) \wedge \beta(x)) \qquad \begin{array}{c}[\alpha(a) \wedge \beta(a)]^1 \\ \vdots \\ \exists x\, \gamma(x, b)\end{array}}{\exists x\, \gamma(x, b)}\, \exists\text{Elim}\ {}^{1}$$

The two assumptions we are working with share β. It may be useful at this point to apply $\wedge$Elim to separate out $\beta(a)$.

$$\dfrac{\exists x\, (\alpha(x) \wedge \beta(x)) \qquad \begin{array}{c}\dfrac{[\alpha(a) \wedge \beta(a)]^1}{\beta(a)}\, \wedge\text{Elim} \\ \vdots \\ \exists x\, \gamma(x, b)\end{array}}{\exists x\, \gamma(x, b)}\, \exists\text{Elim}\ {}^{1}$$

The second assumption we have to work with is $\forall x\, (\beta(x) \to \gamma(x, b))$. Since there is no eigenvariable condition we can instantiate x with the constant symbol a using $\forall$Elim to get $\beta(a) \to \gamma(a, b)$. We now have both $\beta(a) \to \gamma(a, b)$ and $\beta(a)$. Our next move should be a straightforward application of the $\to$Elim rule.

$$\dfrac{\exists x\, (\alpha(x) \wedge \beta(x)) \qquad \begin{array}{c}\dfrac{\dfrac{\forall x\, (\beta(x) \to \gamma(x, b))}{\beta(a) \to \gamma(a, b)}\, \forall\text{Elim} \qquad \dfrac{[\alpha(a) \wedge \beta(a)]^1}{\beta(a)}\, \wedge\text{Elim}}{\gamma(a, b)}\, \to\text{Elim} \\ \vdots \\ \exists x\, \gamma(x, b)\end{array}}{\exists x\, \gamma(x, b)}\, \exists\text{Elim}\ {}^{1}$$

We are so close! One application of $\exists$Intro and we have reached our goal.

$$\dfrac{\exists x\, (\alpha(x) \wedge \beta(x)) \qquad \begin{array}{c}\dfrac{\dfrac{\forall x\, (\beta(x) \to \gamma(x, b))}{\beta(a) \to \gamma(a, b)}\, \forall\text{Elim} \qquad \dfrac{[\alpha(a) \wedge \beta(a)]^1}{\beta(a)}\, \wedge\text{Elim}}{\gamma(a, b)}\, \to\text{Elim} \\ \dfrac{\gamma(a, b)}{\exists x\, \gamma(x, b)}\, \exists\text{Intro}\end{array}}{\exists x\, \gamma(x, b)}\, \exists\text{Elim}\ {}^{1}$$

Since we ensured at each step that the eigenvariable conditions were not violated, we can be confident that this is a correct derivation.

Example 5.6.3. Give a derivation of the wff $\neg\forall x\,\alpha(x)$ from the assumptions $\forall x\,\alpha(x) \to \exists y\,\beta(y)$ and $\neg\exists y\,\beta(y)$. Starting as usual, we write the target wff at the bottom.

$$\dfrac{}{\neg\forall x\,\alpha(x)}$$

The last line of the derivation is a negation, so let's try using ¬Intro. This will require that we figure out how to derive a contradiction.

$$\dfrac{\begin{array}{c}[\forall x\,\alpha(x)]^1\\ \vdots\\ \bot\end{array}}{\neg\forall x\,\alpha(x)}\,{}^1\,\neg\text{Intro}$$

So far so good. We can use ∀Elim but it's not obvious if that will help us get to our goal. Instead, let's use one of our assumptions. $\forall x\,\alpha(x) \to \exists y\,\beta(y)$ together with $\forall x\,\alpha(x)$ will allow us to use the →Elim rule.

$$\dfrac{\dfrac{\forall x\,\alpha(x) \to \exists y\,\beta(y) \qquad [\forall x\,\alpha(x)]^1}{\exists y\,\beta(y)}\,\to\text{Elim}}{\begin{array}{c}\vdots\\ \bot\end{array}}\,{}^1\,\neg\text{Intro}$$

We now have one final assumption to work with, and it looks like this will help us reach a contradiction by using ¬Elim.

$$\dfrac{\neg\exists y\,\beta(y) \qquad \dfrac{\forall x\,\alpha(x) \to \exists y\,\beta(y) \qquad [\forall x\,\alpha(x)]^1}{\exists y\,\beta(y)}\,\to\text{Elim}}{\bot}\,\neg\text{Elim}$$
$$\dfrac{\bot}{\neg\forall x\,\alpha(x)}\,{}^1\,\neg\text{Intro}$$

5.7 Proof-Theoretic Notions

Just as we've defined a number of important semantic notions (validity, entailment, satisfiabilty), we now define corresponding *proof-theoretic notions*. These are not defined by appeal to satisfaction of sentences in structures, but by appeal to the derivability or non-derivability of certain sentences from others. It was an important discovery, due to Gödel, that these notions coincide. That they do is the content of the *completeness theorem*.

Definition 57A (Theorems) A sentence α is a *theorem* if there is a derivation of α in natural deduction in which all assumptions are discharged. We write $\vdash \alpha$ if α is a theorem and $\nvdash \alpha$ if it is not.

Definition 57B (Derivability) A sentence α is *derivable from* a set of sentences Γ, $\Gamma \vdash \alpha$, if there is a derivation with conclusion α and in which

every assumption is either discharged or is in Γ. If α is not derivable from Γ we write $\Gamma \nvdash \alpha$.

DEFINITION 57C (CONSISTENCY) A set of sentences Γ is *inconsistent* iff $\Gamma \vdash \bot$. If Γ is not inconsistent, i.e., if $\Gamma \nvdash \bot$, we say it is *consistent*.

PROPOSITION 57D (REFLEXIVITY) If $\alpha \in \Gamma$, then $\Gamma \vdash \alpha$.

Proof. The assumption α by itself is a derivation of α where every undischarged assumption (i.e., α) is in Γ. □

PROPOSITION 57E (MONOTONY) If $\Gamma \subseteq \Delta$ and $\Gamma \vdash \alpha$, then $\Delta \vdash \alpha$.

Proof. Any derivation of α from Γ is also a derivation of α from Δ. □

PROPOSITION 57F (TRANSITIVITY) If $\Gamma \vdash \alpha$ for every $\alpha \in \Delta$ and $\Delta \vdash \beta$, then $\Gamma \vdash \beta$.

Proof. If $\Delta \vdash \beta$, then there is a derivation δ_0 of β with all undischarged assumptions in Δ. We show that $\Gamma \vdash \beta$ by induction on the number n of undischarged assumptions in δ_0.

If $n = 0$, then δ_0 has no undischarged assumptions, and so also counts as a derivation of β from Γ.

Otherwise, pick an undischarged assumption α in δ_0 and let Δ_1 be the remaining undischarged assumptions. We obtain the derivation δ_1:

$$\dfrac{\begin{array}{c}\Delta_1, [\alpha]^1 \\ \vdots\,\delta_0 \\ \beta\end{array}}{\alpha \to \beta} \to\text{Intro}$$

Since the number of undischarged assumptions in δ_1 is $n - 1$, the inductive hypothesis applies: there is a derivation δ_2 of $\alpha \to \beta$ from Γ. Since $\Gamma \vdash \alpha$ there is also a derivation δ_3 of α from Γ. Now consider:

$$\dfrac{\begin{array}{cc}\Gamma & \Gamma \\ \vdots\,\delta_2 & \vdots\,\delta_3 \\ \alpha \to \beta & \alpha\end{array}}{\beta} \to\text{Elim}$$

This shows $\Gamma \vdash \beta$. □

PROPOSITION 57G Γ is inconsistent iff $\Gamma \vdash \alpha$ for every sentence α.

Proof. Exercise. □

PROPOSITION 57H (COMPACTNESS) 1. If $\Gamma \vdash \alpha$ then there is a finite subset $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vdash \alpha$.

2. If every finite subset of Γ is consistent, then Γ is consistent.

Proof. 1. If $\Gamma \vdash \alpha$, then there is a derivation δ of α from Γ. Let Γ_0 be the set of undischarged assumptions of δ. Since any derivation is finite, Γ_0 can only contain finitely many sentences. So, δ is a derivation of α from a finite $\Gamma_0 \subseteq \Gamma$.

2. This is the contrapositive of (1) for the special case $\alpha \equiv \bot$.

□

5.8 Derivability and Consistency

We will now establish a number of properties of the derivability relation. They are independently interesting, but each will play a role in the proof of the completeness theorem.

PROPOSITION 58A If $\Gamma \vdash \alpha$ and $\Gamma \cup \{\alpha\}$ is inconsistent, then Γ is inconsistent.

Proof. Let the derivation of α from Γ be δ_1 and the derivation of $\bot$ from $\Gamma \cup \{\alpha\}$ be δ_2. We can then derive:

$$\dfrac{\dfrac{\Gamma, [\alpha]^1 \quad \vdots\ \delta_2}{\dfrac{\bot}{\neg\alpha}\ 1\ \neg\text{Intro}} \qquad \begin{matrix}\Gamma \\ \vdots\ \delta_1 \\ \alpha\end{matrix}}{\bot}\ \neg\text{Elim}$$

In the new derivation, the assumption α is discharged, so it is a derivation from Γ. □

PROPOSITION 58B $\Gamma \vdash \alpha$ iff $\Gamma \cup \{\neg\alpha\}$ is inconsistent.

Proof. First suppose $\Gamma \vdash \alpha$, i.e., there is a derivation δ_0 of α from undischarged assumptions Γ. We obtain a derivation of $\bot$ from $\Gamma \cup \{\neg\alpha\}$ as follows:

$$\dfrac{\neg\alpha \qquad \begin{matrix}\Gamma \\ \vdots\ \delta_0 \\ \alpha\end{matrix}}{\bot}\ \neg\text{Elim}$$

Now assume $\Gamma \cup \{\neg\alpha\}$ is inconsistent, and let δ_1 be the corresponding derivation of $\bot$ from undischarged assumptions in $\Gamma \cup \{\neg\alpha\}$. We obtain a derivation of α from Γ alone by using $\bot_C$:

$$\dfrac{\dfrac{\Gamma, [\neg\alpha]^1 \\ \vdots\, \delta_1 \\ \bot}{\alpha}\, \bot_C}{}\ \square$$

PROPOSITION 58C If $\Gamma \vdash \alpha$ and $\neg\alpha \in \Gamma$, then Γ is inconsistent.

Proof. Suppose $\Gamma \vdash \alpha$ and $\neg\alpha \in \Gamma$. Then there is a derivation δ of α from Γ. Consider this simple application of the ¬Elim rule:

$$\dfrac{\neg\alpha \qquad \begin{array}{c}\Gamma \\ \vdots\, \delta \\ \alpha\end{array}}{\bot}\, \neg\text{Elim}$$

Since $\neg\alpha \in \Gamma$, all undischarged assumptions are in Γ, this shows that $\Gamma \vdash \bot$. $\square$

PROPOSITION 58D If $\Gamma \cup \{\alpha\}$ and $\Gamma \cup \{\neg\alpha\}$ are both inconsistent, then Γ is inconsistent.

Proof. There are derivations δ_1 and δ_2 of $\bot$ from $\Gamma \cup \{\alpha\}$ and $\bot$ from $\Gamma \cup \{\neg\alpha\}$, respectively. We can then derive

$$\dfrac{\dfrac{\dfrac{\Gamma, [\neg\alpha]^2 \\ \vdots\, \delta_2 \\ \bot}{\neg\neg\alpha}\, {}^2\neg\text{Intro} \qquad \dfrac{\Gamma, [\alpha]^1 \\ \vdots\, \delta_1 \\ \bot}{\neg\alpha}\, {}^1\neg\text{Intro}}{\bot}\, \neg\text{Elim}}{}$$

Since the assumptions α and $\neg\alpha$ are discharged, this is a derivation of $\bot$ from Γ alone. Hence Γ is inconsistent. $\square$

5.9 Derivability and the Propositional Connectives

PROPOSITION 59A 1. Both $\alpha \wedge \beta \vdash \alpha$ and $\alpha \wedge \beta \vdash \beta$

2. $\alpha, \beta \vdash \alpha \wedge \beta$.

Proof. 1. We can derive both

$$\dfrac{\alpha \wedge \beta}{\alpha}\, \wedge\text{Elim} \qquad \dfrac{\alpha \wedge \beta}{\beta}\, \wedge\text{Elim}$$

2. We can derive:

$$\frac{\alpha \qquad \beta}{\alpha \wedge \beta}\,\wedge\text{Intro}$$

□

PROPOSITION 59B 1. $\alpha \vee \beta, \neg\alpha, \neg\beta$ is inconsistent.

2. Both $\alpha \vdash \alpha \vee \beta$ and $\beta \vdash \alpha \vee \beta$.

Proof. 1. Consider the following derivation:

$$\dfrac{\alpha \vee \beta \qquad \dfrac{\neg\alpha \qquad [\alpha]^1}{\bot}\,\neg\text{Elim} \qquad \dfrac{\neg\beta \qquad [\beta]^1}{\bot}\,\neg\text{Elim}}{\bot}\,{}^{1}\vee\text{Elim}$$

This is a derivation of $\bot$ from undischarged assumptions $\alpha \vee \beta$, $\neg\alpha$, and $\neg\beta$.

2. We can derive both

$$\frac{\alpha}{\alpha \vee \beta}\,\vee\text{Intro} \qquad\qquad \frac{\beta}{\alpha \vee \beta}\,\vee\text{Intro}$$

□

PROPOSITION 59C 1. $\alpha, \alpha \to \beta \vdash \beta$.

2. Both $\neg\alpha \vdash \alpha \to \beta$ and $\beta \vdash \alpha \to \beta$.

Proof. 1. We can derive:

$$\frac{\alpha \to \beta \qquad \beta}{\beta}\,\to\text{Elim}$$

2. This is shown by the following two derivations:

$$\dfrac{\dfrac{\dfrac{\neg\alpha \qquad [\alpha]^1}{\bot}\,\neg\text{Elim}}{\beta}\,\bot_I}{\alpha \to \beta}\,{}^{1}\to\text{Intro} \qquad\qquad \frac{\beta}{\alpha \to \beta}\,\to\text{Intro}$$

Note that $\to$Intro may, but does not have to, discharge the assumption α.

□

5.10 Derivability and the Quantifiers

THEOREM 510A If c is a constant not occurring in Γ or $\alpha(x)$ and $\Gamma \vdash \alpha(c)$, then $\Gamma \vdash \forall x\, \alpha(x)$.

Proof. Let δ be a derivation of $\alpha(c)$ from Γ. By adding a $\forall$Intro inference, we obtain a proof of $\forall x\, \alpha(x)$. Since c does not occur in Γ or $\alpha(x)$, the eigenvariable condition is satisfied. □

PROPOSITION 510B $\forall x\, \alpha(x) \vdash \alpha(t)$.

Proof. The following is a derivation of $\alpha(t)$ from $\forall x\, \alpha(x)$:

$$\dfrac{\forall x\, \alpha(x)}{\alpha(t)}\ \forall\text{Elim}$$

□

5.11 Soundness

A derivation system, such as natural deduction, is *sound* if it cannot derive things that do not actually follow. Soundness is thus a kind of guaranteed safety property for derivation systems. Depending on which proof theoretic property is in question, we would like to know for instance, that

1. every derivable sentence is valid;
2. if a sentence is derivable from some others, it is also a consequence of them;
3. if a set of sentences is inconsistent, it is unsatisfiable.

These are important properties of a derivation system. If any of them do not hold, the derivation system is deficient—it would derive too much. Consequently, establishing the soundness of a derivation system is of the utmost importance.

THEOREM 511A (SOUNDNESS) If α is derivable from the undischarged assumptions Γ, then $\Gamma \vDash \alpha$.

Proof. Let δ be a derivation of α. We proceed by induction on the number of inferences in δ.

For the induction basis we show the claim if the number of inferences is 0. In this case, δ consists only of an initial wff. Every initial wff α is an undischarged assumption, and as such, any structure $\mathfrak{A}$ that satisfies all of the undischarged assumptions of the proof also satisfies α.

Now for the inductive step. Suppose that δ contains n inferences. The premise(s) of the lowermost inference are derived using sub-derivations, each of which contains fewer than n inferences. We assume the induction hypothesis: The premises of the last inference follow from the undischarged assumptions of

the sub-derivations ending in those premises. We have to show that α follows from the undischarged assumptions of the entire proof.

We distinguish cases according to the type of the lowermost inference. First, we consider the possible inferences with only one premise.

1. Suppose that the last inference is ¬Intro: The derivation has the form

$$\dfrac{\begin{array}{c}\Gamma, [\alpha]^n \\ \vdots \\ \delta_1 \\ \vdots \\ \bot\end{array}}{\neg\alpha}\, {}^n\neg\text{Intro}$$

   By inductive hypothesis, $\bot$ follows from the undischarged assumptions $\Gamma \cup \{\alpha\}$ of δ_1. Consider a structure $\mathfrak{A}$. We need to show that, if $\models_{\mathfrak{A}} \Gamma$, then $\models_{\mathfrak{A}} \neg\alpha$. Suppose for reductio that $\models_{\mathfrak{A}} \Gamma$, but $\not\models_{\mathfrak{A}} \neg\alpha$, i.e., $\models_{\mathfrak{A}} \alpha$. This would mean that $\models_{\mathfrak{A}} \Gamma \cup \{\alpha\}$. This is contrary to our inductive hypothesis. So, $\models_{\mathfrak{A}} \neg\alpha$.

2. The last inference is ∧Elim: There are two variants: α or β may be inferred from the premise $\alpha \wedge \beta$. Consider the first case. The derivation δ looks like this:

$$\dfrac{\begin{array}{c}\Gamma \\ \vdots \\ \delta_1 \\ \vdots \\ \alpha \wedge \beta\end{array}}{\alpha}\,\wedge\text{Elim}$$

   By inductive hypothesis, $\alpha \wedge \beta$ follows from the undischarged assumptions Γ of δ_1. Consider a structure $\mathfrak{A}$. We need to show that, if $\models_{\mathfrak{A}} \Gamma$, then $\models_{\mathfrak{A}} \alpha$. Suppose $\models_{\mathfrak{A}} \Gamma$. By our inductive hypothesis ($\Gamma \vDash \alpha \vee \beta$), we know that $\models_{\mathfrak{A}} \alpha \wedge \beta$. By definition, $\models_{\mathfrak{A}} \alpha \wedge \beta$ iff $\models_{\mathfrak{A}} \alpha$ and $\models_{\mathfrak{A}} \beta$. (The case where β is inferred from $\alpha \wedge \beta$ is handled similarly.)

3. The last inference is ∨Intro: There are two variants: $\alpha \vee \beta$ may be inferred from the premise α or the premise β. Consider the first case. The derivation has the form

$$\dfrac{\begin{array}{c}\Gamma \\ \vdots \\ \delta_1 \\ \vdots \\ \alpha\end{array}}{\alpha \vee \beta}\,\vee\text{Intro}$$

By inductive hypothesis, α follows from the undischarged assumptions Γ of δ_1. Consider a structure $\mathfrak{A}$. We need to show that, if $\models_{\mathfrak{A}} \Gamma$, then $\models_{\mathfrak{A}} \alpha \vee \beta$. Suppose $\models_{\mathfrak{A}} \Gamma$; then $\models_{\mathfrak{A}} \alpha$ since $\Gamma \vDash \alpha$ (the inductive hypothesis). So it must also be the case that $\models_{\mathfrak{A}} \alpha \vee \beta$. (The case where $\alpha \vee \beta$ is inferred from β is handled similarly.)

4. The last inference is $\to$Intro: $\alpha \to \beta$ is inferred from a subproof with assumption α and conclusion β, i.e.,

$$\dfrac{\begin{array}{c}\Gamma, [\alpha]^n \\ \vdots\,\delta_1 \\ \vdots \\ \beta\end{array}}{\alpha \to \beta}\, n \to\text{Intro}$$

   By inductive hypothesis, β follows from the undischarged assumptions of δ_1, i.e., $\Gamma \cup \{\alpha\} \vDash \beta$. Consider a structure $\mathfrak{A}$. The undischarged assumptions of δ are just Γ, since α is discharged at the last inference. So we need to show that $\Gamma \vDash \alpha \to \beta$. For reductio, suppose that for some structure $\mathfrak{A}$, $\models_{\mathfrak{A}} \Gamma$ but $\not\models_{\mathfrak{A}} \alpha \to \beta$. So, $\models_{\mathfrak{A}} \alpha$ and $\not\models_{\mathfrak{A}} \beta$. But by hypothesis, β is a consequence of $\Gamma \cup \{\alpha\}$, i.e., $\models_{\mathfrak{A}} \beta$, which is a contradiction. So, $\Gamma \vDash \alpha \to \beta$.

5. The last inference is $\bot_I$: Here, δ ends in

$$\dfrac{\begin{array}{c}\Gamma \\ \vdots\,\delta_1 \\ \vdots \\ \bot\end{array}}{\alpha}\, \bot_I$$

   By induction hypothesis, $\Gamma \vDash \bot$. We have to show that $\Gamma \vDash \alpha$. Suppose not; then for some $\mathfrak{A}$ we have $\models_{\mathfrak{A}} \Gamma$ and $\not\models_{\mathfrak{A}} \alpha$. But we always have $\not\models_{\mathfrak{A}} \bot$, so this would mean that $\Gamma \not\vDash \bot$, contrary to the induction hypothesis.

6. The last inference is $\bot_C$: Exercise.

7. The last inference is $\forall$Intro: Then δ has the form

$$\dfrac{\begin{array}{c}\Gamma \\ \vdots\,\delta_1 \\ \vdots \\ \alpha(a)\end{array}}{\forall x\, \alpha(x)}\, \forall\text{Intro}$$

The premise $\alpha(a)$ is a consequence of the undischarged assumptions Γ by induction hypothesis. Consider some structure, $\mathfrak{A}$, such that $\models_{\mathfrak{A}} \Gamma$. We need to show that $\models_{\mathfrak{A}} \forall x\, \alpha(x)$. Since $\forall x\, \alpha(x)$ is a sentence, this means we have to show that for every variable assignment s, $\models_{\mathfrak{A}} \alpha(x)[s]$ (Proposition 112F). Since Γ consists entirely of sentences, $\models_{\mathfrak{A}} \beta[s]$ for all $\beta \in \Gamma$ by Definition 111D. Let $\mathfrak{M}'$ be like $\mathfrak{A}$ except that $a^{\mathfrak{A}'} = s(x)$. Since a does not occur in Γ, $\models_{\mathfrak{A}'} \Gamma$ by corollary 1.13.2. Since $\Gamma \vDash A(a)$, $\models_{\mathfrak{A}'} A(a)$. Since $\alpha(a)$ is a sentence, $\models_{\mathfrak{A}} \alpha(a)[s]$ by Proposition 112E. $\models_{\mathfrak{A}'} \alpha(x)[s]$ iff $\models_{\mathfrak{A}'} \alpha(a)$ by Proposition 113D (recall that $\alpha(a)$ is just $\alpha(x)[a/x]$). So, $\models_{\mathfrak{A}'} \alpha(x)[s]$. Since a does not occur in $\alpha(x)$, by Proposition 113A, $\models_{\mathfrak{A}} \alpha(x)[s]$. But s was an arbitrary variable assignment, so $\models_{\mathfrak{A}} \forall x\, \alpha(x)$.

8. The last inference is ∃Intro: Exercise.

9. The last inference is ∀Elim: Exercise.

Now let's consider the possible inferences with several premises: ∨Elim, ∧Intro, →Elim, and ∃Elim.

1. The last inference is ∧Intro. $\alpha \wedge \beta$ is inferred from the premises α and β and δ has the form

$$\dfrac{\begin{array}{cc}\Gamma_1 & \Gamma_2 \\ \vdots & \vdots \\ \delta_1 & \delta_2 \\ \vdots & \vdots \\ \alpha & \beta\end{array}}{\alpha \wedge \beta}\ \wedge\text{Intro}$$

   By induction hypothesis, α follows from the undischarged assumptions Γ_1 of δ_1 and β follows from the undischarged assumptions Γ_2 of δ_2. The undischarged assumptions of δ are $\Gamma_1 \cup \gamma_2$, so we have to show that $\Gamma_1 \cup \Gamma_2 \vDash \alpha \wedge \beta$. Consider a structure $\mathfrak{A}$ with $\models_{\mathfrak{A}} \Gamma_1 \cup \Gamma_2$. Since $\models_{\mathfrak{A}} \Gamma_1$, it must be the case that $\models_{\mathfrak{A}} \alpha$ as $\Gamma_1 \vDash \alpha$, and since $\models_{\mathfrak{A}} \Gamma_2$, $\models_{\mathfrak{A}} \beta$ since $\Gamma_2 \vDash \beta$. Together, $\models_{\mathfrak{A}} \alpha \wedge \beta$.

2. The last inference is ∨Elim: Exercise.

3. The last inference is →Elim. β is inferred from the premises $\alpha \to \beta$ and α. The derivation δ looks like this:

$$\dfrac{\begin{array}{cc}\Gamma_1 & \Gamma_2 \\ \vdots & \vdots \\ \delta_1 & \delta_2 \\ \vdots & \vdots \\ \alpha \to \beta & \alpha\end{array}}{\beta}\ \to\text{Elim}$$

By induction hypothesis, $\alpha \to \beta$ follows from the undischarged assumptions Γ_1 of δ_1 and α follows from the undischarged assumptions Γ_2 of δ_2. Consider a structure $\mathfrak{A}$. We need to show that, if $\models_{\mathfrak{A}} \Gamma_1 \cup \Gamma_2$, then $\models_{\mathfrak{A}} \beta$. Suppose $\models_{\mathfrak{A}} \Gamma_1 \cup \Gamma_2$. Since $\Gamma_1 \vDash \alpha \to \beta$, $\models_{\mathfrak{A}} \alpha \to \beta$. Since $\Gamma_2 \vDash \alpha$, we have $\models_{\mathfrak{A}} \alpha$. This means that $\models_{\mathfrak{A}} \beta$ (For if $\not\models_{\mathfrak{A}} \beta$, since $\models_{\mathfrak{A}} \alpha$, we'd have $\not\models_{\mathfrak{A}} \alpha \to \beta$, contradicting $\models_{\mathfrak{A}} \alpha \to \beta$).

4. The last inference is ¬Elim: Exercise.

5. The last inference is ∃Elim: Exercise.

□

COROLLARY 5.11.2 If $\vdash \alpha$, then α is valid.

COROLLARY 5.11.3 If Γ is satisfiable, then it is consistent.

Proof. We prove the contrapositive. Suppose that Γ is not consistent. Then $\Gamma \vdash \bot$, i.e., there is a derivation of $\bot$ from undischarged assumptions in Γ. By Theorem 511A, any structure $\mathfrak{A}$ that satisfies Γ must satisfy $\bot$. Since $\not\models_{\mathfrak{A}} \bot$ for every structure $\mathfrak{A}$, no $\mathfrak{A}$ can satisfy Γ, i.e., Γ is not satisfiable. □

5.12 Derivations with Equality symbol

Derivations with equality symbol require additional inference rules.

$$\frac{}{t = t}\ =\text{Intro} \qquad \frac{t_1 = t_2 \qquad \alpha(t_1)}{\alpha(t_2)}\ =\text{Elim} \qquad \frac{t_1 = t_2 \qquad \alpha(t_2)}{\alpha(t_1)}\ =\text{Elim}$$

In the above rules, t, t_1, and t_2 are closed terms. The =Intro rule allows us to derive any identity statement of the form $t = t$ outright, from no assumptions.

Example 5.12.1. If s and t are closed terms, then $\alpha(s), s = t \vdash \alpha(t)$:

$$\frac{s = t \qquad \alpha(s)}{\alpha(t)}\ =\text{Elim}$$

This may be familiar as the "principle of substitutability of identicals," or Leibniz' Law.

Example 5.12.2. We derive the sentence

$$\forall x\, \forall y\, ((\alpha(x) \wedge \alpha(y)) \to x = y)$$

from the sentence

$$\exists x\, \forall y\, (\alpha(y) \to y = x)$$

We develop the derivation backwards:

$$\dfrac{\dfrac{\dfrac{\exists x\,\forall y\,(\alpha(y)\to y=x)\quad [\alpha(a)\wedge\alpha(b)]^1}{\vdots}}{\dfrac{\dfrac{\dfrac{a=b}{((\alpha(a)\wedge\alpha(b))\to a=b)}\,{}^1\to\text{Intro}}{\forall y\,((\alpha(a)\wedge\alpha(y))\to a=y)}\,\forall\text{Intro}}{\forall x\,\forall y\,((\alpha(x)\wedge\alpha(y))\to x=y)}\,\forall\text{Intro}}$$

We'll now have to use the main assumption: since it is an existential wff, we use ∃Elim to derive the intermediary conclusion $a = b$.

$$\dfrac{\dfrac{\dfrac{\exists x\,\forall y\,(\alpha(y)\to y=x)\qquad \dfrac{[\forall y\,(\alpha(y)\to y=c)]^2\quad [\alpha(a)\wedge\alpha(b)]^1}{\vdots}}{a=b}\,{}^2\exists\text{Elim}}{\dfrac{\dfrac{\dfrac{a=b}{((\alpha(a)\wedge\alpha(b))\to a=b)}\,{}^1\to\text{Intro}}{\forall y\,((\alpha(a)\wedge\alpha(y))\to a=y)}\,\forall\text{Intro}}{\forall x\,\forall y\,((\alpha(x)\wedge\alpha(y))\to x=y)}\,\forall\text{Intro}}$$

The sub-derivation on the top right is completed by using its assumptions to show that $a = c$ and $b = c$. This requies two separate derivations. The derivation for $a = c$ is as follows:

$$\dfrac{\dfrac{[\forall y\,(\alpha(y)\to y=c)]^2}{\alpha(a)\to a=c}\,\forall\text{Elim}\qquad \dfrac{[\alpha(a)\wedge\alpha(b)]^1}{\alpha(a)}\,\wedge\text{Elim}}{a=c}\,\to\text{Elim}$$

From $a = c$ and $b = c$ we derive $a = b$ by =Elim.

5.13 Soundness with Equality symbol

PROPOSITION 513A Natural deduction with rules for = is sound.

Proof. Any wff of the form $t = t$ is valid, since for every structure $\mathfrak{A}$, $\models_{\mathfrak{A}} t = t$. (Note that we assume the term t to be ground, i.e., it contains no variables, so variable assignments are irrelevant).

Suppose the last inference in a derivation is =Elim, i.e., the derivation has the following form:

$$\dfrac{\begin{array}{cc}\Gamma_1 & \Gamma_2\\ \vdots & \vdots\\ \delta_1 & \delta_2\\ \vdots & \vdots\\ t_1=t_2 & \alpha(t_1)\end{array}}{\alpha(t_2)}\,=\text{Elim}$$

The premises $t_1 = t_2$ and $\alpha(t_1)$ are derived from undischarged assumptions Γ_1 and Γ_2, respectively. We want to show that $\alpha(t_2)$ follows from $\Gamma_1 \cup \Gamma_2$. Consider a structure $\mathfrak{A}$ with $\models_{\mathfrak{A}} \Gamma_1 \cup \Gamma_2$. By induction hypothesis, $\models_{\mathfrak{A}} \alpha(t_1)$ and $\models_{\mathfrak{A}} t_1 = t_2$. Therefore, ${t_1}^{\mathfrak{A}} = {t_2}^{\mathfrak{A}}$. Let s be any variable assignment, and s' be the x-variant given by $s'(x) = {t_1}^{\mathfrak{A}} = {t_2}^{\mathfrak{A}}$. By Proposition 113D, $\models_{\mathfrak{A}} \alpha(t_1)[s]$ iff $\models_{\mathfrak{A}} \alpha(x)[s']$ iff $\models_{\mathfrak{A}} \alpha(t_2)[s]$. Since $\models_{\mathfrak{A}} \alpha(t_1)$, we have $\models_{\mathfrak{A}} \alpha(t_2)$. □

Chapter 6

The Completeness Theorem

6.1 Introduction

The completeness theorem is one of the most fundamental results about logic. It comes in two formulations, the equivalence of which we'll prove. In its first formulation it says something fundamental about the relationship between semantic consequence and our proof system: if a sentence α follows from some sentences Γ, then there is also a derivation that establishes $\Gamma \vdash \alpha$. Thus, the proof system is as strong as it can possibly be without proving things that don't actually follow. In its second formulation, it can be stated as a model existence result: every consistent set of sentences is satisfiable.

These aren't the only reasons the completeness theorem—or rather, its proof—is important. It has a number of important consequences, some of which we'll discuss separately. For instance, since any derivation that shows $\Gamma \vdash \alpha$ is finite and so can only use finitely many of the sentences in Γ, it follows by the completeness theorem that if α is a consequence of Γ, it is already a consequence of a finite subset of Γ. This is called *compactness*. Equivalently, if every finite subset of Γ is consistent, then Γ itself must be consistent. It also follows from *the proof of* the completeness theorem that any satisfiable set of sentences has a finite or denumerable model. This result is called the Löwenheim-Skolem theorem.

6.2 Outline of the Proof

The proof of the completeness theorem is a bit complex, and upon first reading it, it is easy to get lost. So let us outline the proof. The first step is a shift of perspective, that allows us to see a route to a proof. When completeness is thought of as "whenever $\Gamma \vDash \alpha$ then $\Gamma \vdash \alpha$," it may be hard to even come up with an idea: for to show that $\Gamma \vdash \alpha$ we have to find a derivation, and it does not look like the hypothesis that $\Gamma \vDash \alpha$ helps us for this in any way. For some proof systems it is possible to directly construct a derivation, but we will take a slightly different tack. The shift in perspective required is this: completeness

can also be formulated as: "if Γ is consistent, it has a model." Perhaps we can use the information in Γ together with the hypothesis that it is consistent to construct a model. After all, we know what kind of model we are looking for: one that is as Γ describes it!

If Γ contains only atomic sentences, it is easy to construct a model for it: for atomic sentences are all of the form $Pa_1 \dots a_n$ where the a_i are constant symbols. So all we have to do is come up with a domain $|\mathfrak{A}|$ and an interpretation for P so that $\models_{\mathfrak{A}} Pa_1 \dots a_n$. But nothing's easier than that: put $|\mathfrak{A}| = \mathbb{N}$, $c_i^{\mathfrak{A}} = i$, and for every $Pa_1 \dots a_n \in \Gamma$, put the tuple $\langle k_1, \dots, k_n \rangle$ into $P^{\mathfrak{A}}$, where k_i is the index of the constant symbol a_i (i.e., $a_i \equiv c_{k_i}$).

Now suppose Γ contains some sentence $\neg\beta$, with β atomic. We might worry that the construction of $\mathfrak{A}$ interferes with the possibility of making $\neg\beta$ true. But here's where the consistency of Γ comes in: if $\neg\beta \in \Gamma$, then $\beta \notin \Gamma$, or else Γ would be inconsistent. And if $\beta \notin \Gamma$, then according to our construction of $\mathfrak{A}$, $\not\models_{\mathfrak{A}} \beta$, so $\models_{\mathfrak{A}} \neg\beta$. So far so good.

Now what if Γ contains complex, non-atomic formulas? Say, it contains $\alpha \wedge \beta$. Then we should proceed as if both α and β were in Γ. And if $\alpha \vee \beta \in \Gamma$, then we will have to make at least one of them true, i.e., proceed as if one of them was in Γ.

This suggests the following idea: we add additional sentences to Γ so as to (a) keep the resulting set consistent and (b) make sure that for every possible atomic sentence α, either α is in the resulting set, or $\neg\alpha$, and (c) such that, whenever $\alpha \wedge \beta$ is in the set, so are both α and β, if $\alpha \vee \beta$ is in the set, at least one of α or β is also, etc. We keep doing this (potentially forever). Call the set of all sentences so added Γ^*. Then our construction above would provide us with a structure for which we could prove, by induction, that all sentences in Γ^* are true in $\mathfrak{A}$, and hence also all sentence in Γ since $\Gamma \subseteq \Gamma^*$. It turns out that guaranteeing (a) is enough. A set of sentences for which (a) holds is called *complete*. So our task will be to extend the consistent set Γ to a consistent and complete set Γ^*.

There is one wrinkle in this plan: if $\exists x\, \alpha(x) \in \Gamma$ we would hope to be able to pick some constant symbol c and add $\alpha(c)$ in this process. But how do we know we can always do that? Perhaps we only have a few constant symbols in our language, and for each one of them we have $\neg\beta(c) \in \Gamma$. We can't also add $\beta(c)$, since this would make the set inconsistent, and we wouldn't know whether $\mathfrak{A}$ has to make $\beta(c)$ or $\neg\beta(c)$ true. Moreover, it might happen that Γ contains only sentences in a language that has no constant symbols at all (e.g., the language of set theory).

The solution to this problem is to simply add infinitely many constants at the beginning, plus sentences that connect them with the quantifiers in the right way. (Of course, we have to verify that this cannot introduce an inconsistency.)

Our original construction works well if we only have constant symbols in the atomic sentences. But the language might also contain function symbols. In that case, it might be tricky to find the right functions on $\mathbb{N}$ to assign to these function symbols to make everything work. So here's another trick: instead of using i to interpret c_i, just take the set of constant symbols itself as the

domain. Then $\mathfrak{A}$ can assign every constant symbol to itself: $c_i^{\mathfrak{A}} = c_i$. But why not go all the way: let $|\mathfrak{A}|$ be all *terms* of the language! If we do this, there is an obvious assignment of functions (that take terms as arguments and have terms as values) to function symbols: we assign to the function symbol f_i^n the function which, given n terms $t_1, \ldots, t_n$ as input, produces the term $f_i^n(t_1, \ldots, t_n)$ as value.

The last piece of the puzzle is what to do with $=$. The predicate symbol $=$ has a fixed interpretation: $\models_{\mathfrak{A}} t = t'$ iff $t^{\mathfrak{A}} = t'^{\mathfrak{A}}$. Now if we set things up so that the value of a term t is t itself, then this structure will make *no* sentence of the form $t = t'$ true unless t and t' are one and the same term. And of course this is a problem, since basically every interesting theory in a language with function symbols will have as theorems sentences $t = t'$ where t and t' are not the same term (e.g., in theories of arithmetic: $(0 + 0) = 0$). To solve this problem, we change the domain of $\mathfrak{A}$: instead of using terms as the objects in $|\mathfrak{A}|$, we use sets of terms, and each set is so that it contains all those terms which the sentences in Γ require to be equal. So, e.g., if Γ is a theory of arithmetic, one of these sets will contain: 0, $(0 + 0)$, (0×0), etc. This will be the set we assign to 0, and it will turn out that this set is also the value of all the terms in it, e.g., also of $(0 + 0)$. Therefore, the sentence $(0 + 0) = 0$ will be true in this revised structure.

So here's what we'll do. First we investigate the properties of complete consistent sets, in particular we prove that a complete consistent set contains $\alpha \wedge \beta$ iff it contains both α and β, $\alpha \vee \beta$ iff it contains at least one of them, etc. (Proposition 63B). Then we define and investigate "saturated" sets of sentences. A saturated set is one which contains conditionals that link each quantified sentence to instances of it (Definition 64C). We show that any consistent set Γ can always be extended to a saturated set Γ' (Lemma 64D). If a set is consistent, saturated, and complete it also has the property that it contains $\forall x\, \alpha(x)$ iff it contains $\alpha(t)$ for all closed terms t (Proposition 64E). We'll then take the saturated consistent set Γ' and show that it can be extended to a saturated, consistent, and complete set Γ^* (Lemma 65A). This set Γ^* is what we'll use to define our term model $\mathfrak{M}(\mathfrak{d}^*)$. The term model has the set of closed terms as its domain, and the interpretation of its predicate symbols is given by the atomic sentences in Γ^* (Definition 66A). We'll use the properties of consistent, complete, saturated sets to show that indeed $\models_{\mathfrak{A}(\Gamma^*)} \alpha$ iff $\alpha \in \Gamma^*$ (Lemma 66C), and thus in particular, $\models_{\mathfrak{A}(\Gamma^*)} \Gamma$. Finally, we'll consider how to define a term model if Γ contains $=$ as well (Definition 67D) and show that it satisfies Γ^* (Lemma 67F).

6.3 Complete Consistent Sets of Sentences

Definition 63A (Complete set) A set Γ of sentences is *complete* iff for any sentence α, either $\alpha \in \Gamma$ or $\neg\alpha \in \Gamma$.

Complete sets of sentences leave no questions unanswered. For any sentence A, Γ "says" if α is true or false. The importance of complete sets extends

beyond the proof of the completeness theorem. A theory which is complete and axiomatizable, for instance, is always decidable.

Complete consistent sets are important in the completeness proof since we can guarantee that every consistent set of sentences Γ is contained in a complete consistent set Γ^*. A complete consistent set contains, for each sentence α, either α or its negation $\neg\alpha$, but not both. This is true in particular for atomic sentences, so from a complete consistent set in a language suitably expanded by constant symbols, we can construct a structure where the interpretation of predicate symbols is defined according to which atomic sentences are in Γ^*. This structure can then be shown to make all sentences in Γ^* (and hence also all those in Γ) true. The proof of this latter fact requires that $\neg\alpha \in \Gamma^*$ iff $\alpha \notin \Gamma^*$, $(\alpha \vee \beta) \in \Gamma^*$ iff $\alpha \in \Gamma^*$ or $\beta \in \Gamma^*$, etc.

In what follows, we will often tacitly use the properties of reflexivity, monotonicity, and transitivity of $\vdash$ (see sections 4.8 and 5.7).

PROPOSITION 63B Suppose Γ is complete and consistent. Then:

1. If $\Gamma \vdash \alpha$, then $\alpha \in \Gamma$.
2. $\alpha \to \beta \in \Gamma$ iff either $\alpha \notin \Gamma$ or $\beta \in \Gamma$.

Proof. Let us suppose for all of the following that Γ is complete and consistent.

1. If $\Gamma \vdash \alpha$, then $\alpha \in \Gamma$.

   Suppose that $\Gamma \vdash \alpha$. Suppose to the contrary that $\alpha \notin \Gamma$. Since Γ is complete, $\neg\alpha \in \Gamma$. By Propositions 49C and 58C, Γ is inconsistent. This contradicts the assumption that Γ is consistent. Hence, it cannot be the case that $\alpha \notin \Gamma$, so $\alpha \in \Gamma$.

2. For the forward direction, suppose $\alpha \to \beta \in \Gamma$, and suppose to the contrary that $\alpha \in \Gamma$ and $\beta \notin \Gamma$. On these assumptions, $\alpha \to \beta \in \Gamma$ and $\alpha \in \Gamma$. By Propositions 410C and 59C, item (1), $\Gamma \vdash \beta$. But then by (1), $\beta \in \Gamma$, contradicting the assumption that $\beta \notin \Gamma$.

   For the reverse direction, first consider the case where $\alpha \notin \Gamma$. Since Γ is complete, $\neg\alpha \in \Gamma$. By Propositions 410C and 59C, item (2), $\Gamma \vdash \alpha \to \beta$. Again by (1), we get that $\alpha \to \beta \in \Gamma$, as required.

   Now consider the case where $\beta \in \Gamma$. By Propositions 410C and 59C, item (2) again, $\Gamma \vdash \alpha \to \beta$. By (1), $\alpha \to \beta \in \Gamma$.

□

6.4 Henkin Expansion

Part of the challenge in proving the completeness theorem is that the model we construct from a complete consistent set Γ must make all the quantified wffs in Γ true. In order to guarantee this, we use a trick due to Leon Henkin. In essence, the trick consists in expanding the language by infinitely many

constant symbols and adding, for each wff with one free variable $\alpha(x)$ a formula of the form $\neg\forall x\,\alpha \to \neg\alpha(c)$, where c is one of the new constant symbols. When we construct the structure satisfying Γ, this will guarantee that each false universal sentence has a counterexample among the new constants.

PROPOSITION 64A If Γ is consistent in $\mathcal{L}$ and $\mathcal{L}'$ is obtained from $\mathcal{L}$ by adding a denumerable set of new constant symbols d_0, d_1, ..., then Γ is consistent in $\mathcal{L}'$.

DEFINITION 64B (SATURATED SET) A set Γ of wffs of a language $\mathcal{L}$ is *saturated* iff for each wff $\alpha(x) \in \mathrm{Frm}(\mathcal{L})$ with one free variable x there is a constant symbol $c \in \mathcal{L}$ such that $\neg\forall x\,\alpha(x) \to \neg\alpha(c) \in \Gamma$.

The following definition will be used in the proof of the next theorem.

DEFINITION 64C Let $\mathcal{L}'$ be as in Proposition 64A. Fix an enumeration $\alpha_0(x_0)$, $\alpha_1(x_1)$, ...of all wffs $\alpha_i(x_i)$ of $\mathcal{L}'$ in which one variable (x_i) occurs free. We define the sentences δ_n by induction on n.

Let c_0 be the first constant symbol among the d_i we added to $\mathcal{L}$ which does not occur in $\alpha_0(x_0)$. Assuming that δ_0, ..., δ_{n-1} have already been defined, let c_n be the first among the new constant symbols d_i that occurs neither in δ_0, ..., δ_{n-1} nor in $\alpha_n(x_n)$.

Now let δ_n be the wff $\neg\forall x_n\,\alpha_n(x_n) \to \neg\alpha_n(c_n)$.

LEMMA 64D Every consistent set Γ can be extended to a saturated consistent set Γ'.

Proof. Given a consistent set of sentences Γ in a language $\mathcal{L}$, expand the language by adding a denumerable set of new constant symbols to form $\mathcal{L}'$. By Proposition 64A, Γ is still consistent in the richer language. Further, let δ_i be as in Definition 64C. Let

$$\Gamma_0 = \Gamma$$
$$\Gamma_{n+1} = \Gamma_n \cup \{\delta_n\}$$

i.e., $\Gamma_{n+1} = \Gamma \cup \{\delta_0, \ldots, \delta_n\}$, and let $\Gamma' = \bigcup_n \Gamma_n$. Γ' is clearly saturated.

If Γ' were inconsistent, then for some n, Γ_n would be inconsistent (Exercise: explain why). So to show that Γ' is consistent it suffices to show, by induction on n, that each set Γ_n is consistent.

The induction basis is simply the claim that $\Gamma_0 = \Gamma$ is consistent, which is the hypothesis of the theorem. For the induction step, suppose that Γ_n is consistent but $\Gamma_{n+1} = \Gamma_n \cup \{\delta_n\}$ is inconsistent. Recall that δ_n is $\neg\forall x_n\,\alpha_n(x_n) \to \neg\alpha_n(c_n)$, where $\alpha_n(x_n)$ is a wff of $\mathcal{L}'$ with only the variable x_n free. By the way we've chosen the c_n (see Definition 64C), c_n does not occur in $A_n(x_n)$ nor in Γ_n.

If $\Gamma_n \cup \{\delta_n\}$ is inconsistent, then $\Gamma_n \vdash \neg\delta_n$, and hence both of the following hold:

$$\Gamma_n \vdash \neg\forall x_n\,\alpha_n(x_n) \qquad \Gamma_n \vdash \alpha_n(c_n)$$

Since c_n does not occur in Γ_n or in $\alpha_n(x_n)$, Theorems 411A and 510A applies. From $\Gamma_n \vdash \alpha_n(c_n)$, we obtain $\Gamma_n \vdash \forall x_n\,\alpha_n(x_n)$. Thus we have that both $\Gamma_n \vdash$

$\neg\forall x_n\, \alpha_n(x_n)$ and $\Gamma_n \vdash \forall x_n\, \alpha_n(x_n)$, so Γ_n itself is inconsistent. Contradiction: Γ_n was supposed to be consistent. Hence $\Gamma_n \cup \{\delta_n\}$ is consistent. □

We'll now show that *complete*, consistent sets which are saturated have the property that it contains a universally quantified sentence iff it contains all its instancesit contains an existentially quantified sentence iff it contains at least one instance. We'll use this to show that the structure we'll generate from a complete, consistent, saturated set makes all its quantified sentences true.

PROPOSITION 64E Suppose Γ is complete, consistent, and saturated. $\forall x\, \alpha(x) \in \Gamma$ iff $\alpha(t) \in \Gamma$ for all closed terms t.

Proof. Suppose that $\alpha(t) \in \Gamma$ for all closed terms t. By way of contradiction, assume $\forall x\, \alpha(x) \notin \Gamma$. Since Γ is complete, $\neg\forall x\, \alpha(x) \in \Gamma$. By saturation, $(\neg\forall x\, \alpha(x) \to \neg\alpha(c)) \in \Gamma$ for some constant symbol c. By assumption, since c is a closed term, $\alpha(c) \in \Gamma$. But this would make Γ inconsistent. (Exercise: give the derivation that shows

$$\neg\forall x\, \alpha(x), \neg\forall x\, \alpha(x) \to \neg\alpha(c), \alpha(c)$$

is inconsistent.)

For the reverse direction, we do not need saturation: Suppose $\forall x\, \alpha(x) \in \Gamma$. Then $\Gamma \vdash \alpha(t)$ by Theorem 411B and Proposition 510B, item (2). We get $\alpha(t) \in \Gamma$ by Proposition 63B. □

6.5 Lindenbaum's Lemma

We now prove a lemma that shows that any consistent set of sentences is contained in some set of sentences which is not just consistent, but also complete. The proof works by adding one sentence at a time, guaranteeing at each step that the set remains consistent. We do this so that for every α, either α or $\neg\alpha$ gets added at some stage. The union of all stages in that construction then contains either α or its negation $\neg\alpha$ and is thus complete. It is also consistent, since we made sure at each stage not to introduce an inconsistency.

LEMMA 65A (LINDENBAUM'S LEMMA) Every consistent set Γ' in a language $\mathcal{L}'$ can be extended to a complete and consistent set Γ^*.

Proof. Let Γ' be consistent. Let α_0, α_1, ... be an enumeration of all the wffs of $\mathcal{L}'$. Define $\Gamma_0 = \Gamma'$, and

$$\Gamma_{n+1} = \begin{cases} \Gamma_n \cup \{\alpha_n\} & \text{if } \Gamma_n \cup \{\alpha_n\} \text{ is consistent;} \\ \Gamma_n \cup \{\neg\alpha_n\} & \text{otherwise.} \end{cases}$$

Let $\Gamma^* = \bigcup_{n \geq 0} \Gamma_n$.

Each Γ_n is consistent: Γ_0 is consistent by definition. If $\Gamma_{n+1} = \Gamma_n \cup \{\alpha_n\}$, this is because the latter is consistent. If it isn't, $\Gamma_{n+1} = \Gamma_n \cup \{\neg\alpha_n\}$. We have to verify that $\Gamma_n \cup \{\neg\alpha_n\}$ is consistent. Suppose it's not. Then *both* $\Gamma_n \cup \{\alpha_n\}$

and $\Gamma_n \cup \{\neg\alpha_n\}$ are inconsistent. This means that Γ_n would be inconsistent by Propositions 49C and 58C, contrary to the induction hypothesis.

Every finite subset of Γ^* is a subset of Γ_n for some n, since each $\beta \in \Gamma^*$ not already in Γ' is added at some stage i. If n is the last one of these, then all β in the finite subset are in Γ_n. So, every finite subset of Γ^* is consistent. By Propositions 48H and 57H, Γ^* is consistent.

Every sentence of $\mathrm{Frm}(\mathcal{L}')$ appears on the list used to define Γ^*. If $\alpha_n \notin \Gamma^*$, then that is because $\Gamma_n \cup \{\alpha_n\}$ was inconsistent. But then $\neg\alpha_n \in \Gamma^*$, so Γ^* is complete. □

6.6 Construction of a Model

Right now we are not concerned about $=$, i.e., we only want to show that a consistent set Γ of sentences not containing $=$ is satisfiable. We first extend Γ to a consistent, complete, and saturated set Γ^*. In this case, the definition of a model $\mathfrak{M}(\mathfrak{d}^*)$ is simple: We take the set of closed terms of $\mathcal{L}'$ as the domain. We assign every constant symbol to itself, and make sure that more generally, for every closed term t, $t^{\mathfrak{A}(\Gamma^*)} = t$. The predicate symbols are assigned extensions in such a way that an atomic sentence is true in $\mathfrak{M}(\mathfrak{d}^*)$ iff it is in Γ^*. This will obviously make all the atomic sentences in Γ^* true in $\mathfrak{M}(\mathfrak{d}^*)$. The rest are true provided the Γ^* we start with is consistent, complete, and saturated.

Definition 66A (Term model) Let Γ^* be a complete and consistent, saturated set of sentences in a language $\mathcal{L}$. The *term model* $\mathfrak{A}(\Gamma^*)$ of Γ^* is the structure defined as follows:

1. The domain $|\mathfrak{A}(\Gamma^*)|$ is the set of all closed terms of $\mathcal{L}$.
2. The interpretation of a constant symbol c is c itself: $c^{\mathfrak{A}(\Gamma^*)} = c$.
3. The function symbol f is assigned the function which, given as arguments the closed terms $t_1, \ldots, t_n$, has as value the closed term $f(t_1, \ldots, t_n)$:
$$f^{\mathfrak{A}(\Gamma^*)}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$$
4. If R is an n-place predicate symbol, then
$$\langle t_1, \ldots, t_n\rangle \in R^{\mathfrak{A}(\Gamma^*)} \text{ iff } Rt_1 \ldots t_n \in \Gamma^*.$$

A structure $\mathfrak{A}$ may make all instances $\alpha(t)$ of a universally quantified sentence $\forall x\, \alpha(x)$ true, without making $\forall x\, \alpha(x)$ true. This is because in general not every element of $|\mathfrak{A}|$ is the value of a closed term ($\mathfrak{A}$ may not be covered). This is the reason the satisfaction relation is defined via variable assignments. However, for our term model $\mathfrak{M}(\mathfrak{d}^*)$ this wouldn't be necessary—because it is covered. This is the content of the next result.

Proposition 66B Let $\mathfrak{A}(\Gamma^*)$ be the term model of Definition 66A. $\models_{\mathfrak{A}(\Gamma^*)} \forall x\, \alpha(x)$ iff $\models_{\mathfrak{A}} \alpha(t)$ for all terms t.

Proof. By Proposition 112F, $\models_{\mathfrak{A}(\Gamma^*)} \forall x\, \alpha(x)$ iff for every variable assignment s, $\models_{\mathfrak{A}(\Gamma^*)} \alpha(x)[s]$. Recall that $|\mathfrak{A}(\Gamma^*)|$ consists of the closed terms of $\mathcal{L}$, so for every closed term t, $s(x) = t$ is such a variable assignment, and for any variable assignment, $s(x)$ is some closed term t. By Proposition 113D, $\models_{\mathfrak{A}(\Gamma^*)} \alpha(x)[s]$ iff $\models_{\mathfrak{A}(\Gamma^*)} \alpha(t)[s]$, where $s(x) = t$. By Proposition 112E, $\models_{\mathfrak{A}(\Gamma^*)} \alpha(t)[s]$ iff $\models_{\mathfrak{A}(\Gamma^*)} \alpha(t)$, since $\alpha(t)$ is a sentence. □

Lemma 66C (Truth Lemma) Suppose α does not contain $=$. Then $\models_{\mathfrak{A}(\Gamma^*)} \alpha$ iff $\alpha \in \Gamma^*$.

Proof. We prove both directions simultaneously, and by induction on α.

1. $\alpha \equiv R(t_1, \ldots, t_n)$: $\models_{\mathfrak{A}(\Gamma^*)} Rt_1 \ldots t_n$ iff $\langle t_1, \ldots, t_n \rangle \in R^{\mathfrak{A}(\Gamma^*)}$ (by the definition of satisfaction) iff $R(t_1, \ldots, t_n) \in \Gamma^*$ (by the construction of $\mathfrak{A}(\Gamma^*)$).

2. $\alpha \equiv \neg\beta$: $\models_{\mathfrak{A}(\Gamma^*)} \alpha$ iff $\not\models_{\mathfrak{A}(\Gamma^*)} \beta$ (by definition of satisfaction). By induction hypothesis, $\not\models_{\mathfrak{A}(\Gamma^*)} \beta$ iff $\beta \notin \Gamma^*$. Since Γ^* is consistent and complete, $\beta \notin \Gamma^*$ iff $\neg\beta \in \Gamma^*$.

3. $\alpha \equiv \beta \to \gamma$: $\models_{\mathfrak{A}(\Gamma^*)} \alpha$ iff $\not\models_{\mathfrak{A}(\Gamma^*} \beta$ or $\models_{\mathfrak{A}(\Gamma^*)} \gamma$ (by definition of satisfaction) iff $\beta \notin \Gamma^*$ or $\gamma \in \Gamma^*$ (by induction hypothesis). This is the case iff $(\beta \to \gamma) \in \Gamma^*$ (by Proposition 63B(2)).

4. $\alpha \equiv \forall x\, \beta(x)$: $\models_{\mathfrak{A}(\Gamma^*)} \alpha$ iff $\models_{\mathfrak{A}(\Gamma^*)} \beta(t)$ for all terms t (Proposition 66B). By induction hypothesis, this is the case iff $\beta(t) \in \Gamma^*$ for all terms t, by Proposition 64E, this in turn is the case iff $\forall x\, \alpha(x) \in \Gamma^*$.

□

6.7 Identity

The construction of the term model given in the preceding section is enough to establish completeness for first-order logic for sets Γ that do not contain $=$. The term model satisfies every $\alpha \in \Gamma^*$ which does not contain $=$ (and hence all $\alpha \in \Gamma$). It does not work, however, if $=$ is present. The reason is that Γ^* then may contain a sentence $t = t'$, but in the term model the value of any term is that term itself. Hence, if t and t' are different terms, their values in the term model—i.e., t and t', respectively—are different, and so $t = t'$ is false. We can fix this, however, using a construction known as "factoring."

Definition 67A Let Γ^* be a consistent and complete set of sentences in $\mathcal{L}$. We define the relation $\approx$ on the set of closed terms of $\mathcal{L}$ by

$$t \approx t' \quad \text{iff} \quad t = t' \in \Gamma^*$$

Proposition 67B The relation $\approx$ has the following properties:

1. $\approx$ is reflexive.

2. $\approx$ is symmetric.

3. $\approx$ is transitive.

4. If $t \approx t'$, f is a function symbol, and $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n$ are terms, then
$$ft_1 \ldots t_{i-1} t t_{i+1} \ldots t_n \approx ft_1 \ldots t_{i-1} t' t_{i+1} \ldots t_n.$$

5. If $t \approx t'$, R is a predicate symbol, and $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n$ are terms, then
$$Rt_1 \ldots t_{i-1} t t_{i+1} \ldots t_n \in \Gamma^* \text{ iff } Rt_1 \ldots t_{i-1} t' t_{i+1} \ldots t_n \in \Gamma^*.$$

Proof. Since Γ^* is consistent and complete, $t = t' \in \Gamma^*$ iff $\Gamma^* \vdash t = t'$. Thus it is enough to show the following:

1. $\Gamma^* \vdash t = t$ for all terms t.

2. If $\Gamma^* \vdash t = t'$ then $\Gamma^* \vdash t' = t$.

3. If $\Gamma^* \vdash t = t'$ and $\Gamma^* \vdash t' = t''$, then $\Gamma^* \vdash t = t''$.

4. If $\Gamma^* \vdash t = t'$, then
$$\Gamma^* \vdash ft_1 \ldots t_{i-1} t t_{i+1} \ldots t_n = ft_1 \ldots t_{i-1} t' t_{i+1} \ldots t_n$$
for every n-place function symbol f and terms $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n$.

5. If $\Gamma^* \vdash t = t'$ and $\Gamma^* \vdash Rt_1 \ldots t_{i-1} t t_{i+1} \ldots t_n$, then $\Gamma^* \vdash Rt_1 \ldots t_{i-1} t' t_{i+1} \ldots t_n$ for every n-place predicate symbol R and terms $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n$.

□

DEFINITION 67C Suppose Γ^* is a consistent and complete set in a language $\mathcal{L}$, t is a term, and $\approx$ as in the previous definition. Then:
$$[t]_\approx = \{t' : t' \in \mathrm{Trm}(\mathcal{L}), t \approx t'\}$$
and $\mathrm{Trm}(\mathcal{L})/_\approx = \{[t]_\approx : t \in \mathrm{Trm}(\mathcal{L})\}$.

DEFINITION 67D Let $\mathfrak{A} = \mathfrak{A}(\Gamma^*)$ be the term model for Γ^*. Then $\mathfrak{A}/_\approx$ is the following structure:

1. $|\mathfrak{A}/_\approx| = \mathrm{Trm}(\mathcal{L})/_\approx$.

2. $c^{\mathfrak{A}/\approx} = [c]_\approx$

3. $f^{\mathfrak{A}/\approx}([t_1]_\approx, \ldots, [t_n]_\approx) = [ft_1 \ldots t_n]_\approx$

4. $\langle [t_1]_\approx, \ldots, [t_n]_\approx \rangle \in R^{\mathfrak{A}/\approx}$ iff $\models_\mathfrak{A} Rt_1 \ldots t_n$.

Note that we have defined $f^{\mathfrak{A}/\approx}$ and $R^{\mathfrak{A}/\approx}$ for elements of $\mathrm{Trm}(\mathcal{L})/\approx$ by referring to them as $[t]_\approx$, i.e., via *representatives* $t \in [t]_\approx$. We have to make sure that these definitions do not depend on the choice of these representatives, i.e., that for some other choices t' which determine the same equivalence classes ($[t]_\approx = [t']_\approx$), the definitions yield the same result. For instance, if R is a one-place predicate symbol, the last clause of the definition says that $[t]_\approx \in R^{\mathfrak{A}/\approx}$ iff $\models_{\mathfrak{A}} Rt$. If for some other term t' with $t \approx t'$, $\not\models_{\mathfrak{A}} Rt$, then the definition would require $[t']_\approx \notin R^{\mathfrak{A}/\approx}$. If $t \approx t'$, then $[t]_\approx = [t']_\approx$, but we can't have both $[t]_\approx \in R^{\mathfrak{A}/\approx}$ and $[t]_\approx \notin R^{\mathfrak{A}/\approx}$. However, Proposition 67B guarantees that this cannot happen.

PROPOSITION 67E $\mathfrak{A}/\approx$ is well defined, i.e., if $t_1, \ldots, t_n, t'_1, \ldots, t'_n$ are terms, and $t_i \approx t'_i$ then

1. $[ft_1 \ldots t_n]_\approx = [ft'_1 \ldots t'_n]_\approx$, i.e.,

$$ft_1 \ldots t_n \approx ft'_1 \ldots t'_n$$

   and

2. $\models_{\mathfrak{A}} Rt_1 \ldots t_n$ iff $\models_{\mathfrak{A}} Rt'_1 \ldots t'_n$, i.e.,

$$Rt_1 \ldots t_n \in \Gamma^* \text{ iff } Rt'_1 \ldots t'_n \in \Gamma^*.$$

Proof. Follows from Proposition 67B by induction on n. □

LEMMA 67F $\models_{\mathfrak{A}/\approx} \alpha$ iff $\alpha \in \Gamma^*$ for all sentences α.

Proof. By induction on α, just as in the proof of Lemma 66C. The only case that needs additional attention is when $\alpha \equiv t = t'$.

$$\begin{aligned} \models_{\mathfrak{A}/\approx} t = t' &\text{ iff } [t]_\approx = [t']_\approx \text{ (by definition of } \mathfrak{M}/\approx) \\ &\text{iff } t \approx t' \text{ (by definition of } [t]_\approx) \\ &\text{iff } t = t' \in \Gamma^* \text{ (by definition of } \approx). \end{aligned}$$

□

Note that while $\mathfrak{M}(\mathfrak{d}^*)$ is always enumerable and infinite, $\mathfrak{M}/\approx$ may be finite, since it may turn out that there are only finitely many classes $[t]_\approx$. This is to be expected, since Γ may contain sentences which require any structure in which they are true to be finite. For instance, $\forall x \forall y\, x = y$ is a consistent sentence, but is satisfied only in structures with a domain that contains exactly one element.

6.8 The Completeness Theorem

Let's combine our results: we arrive at Gödel's completeness theorem.

THEOREM 68A (COMPLETENESS THEOREM) Let Γ be a set of sentences. If Γ is consistent, it is satisfiable.

Proof. Suppose Γ is consistent. By Lemma 64D, there is a saturated consistent set $\Gamma' \supseteq \Gamma$. By Lemma 65A, there is a $\Gamma^* \supseteq \Gamma'$ which is consistent and complete. Since $\Gamma' \subseteq \Gamma^*$, for each wff α, Γ^* contains a wff of the form $\neg\forall x\, \alpha \to \neg\alpha(c)$ and so Γ^* is saturated.

If Γ does not contain $=$, then by Lemma 66C, $\models_{\mathfrak{A}(\Gamma^*)} \alpha$ iff $\alpha \in \Gamma^*$. From this it follows in particular that for all $\alpha \in \Gamma$, $\models_{\mathfrak{A}(\Gamma^*)} \alpha$, so Γ is satisfiable. If Γ does contain $=$, then by Lemma 67F, $\models_{\mathfrak{A}/\approx} \alpha$ iff $\alpha \in \Gamma^*$ for all sentences α. In particular, $\models_{\mathfrak{A}/\approx} \alpha$ for all $\alpha \in \Gamma$, so Γ is satisfiable. □

COROLLARY 6.8.2 (COMPLETENESS THEOREM, SECOND VERSION) For all Γ and α sentences: if $\Gamma \vDash \alpha$ then $\Gamma \vdash \alpha$.

Proof. Note that the Γ's in corollary 6.8.2 and Theorem 68A are universally quantified. To make sure we do not confuse ourselves, let us restate Theorem 68A using a different variable: for any set of sentences Δ, if Δ is consistent, it is satisfiable. By contraposition, if Δ is not satisfiable, then Δ is inconsistent. We will use this to prove the corollary.

Suppose that $\Gamma \vDash \alpha$. Then $\Gamma \cup \{\neg\alpha\}$ is unsatisfiable by Proposition 114E. Taking $\Gamma \cup \{\neg\alpha\}$ as our Δ, the previous version of Theorem 68A gives us that $\Gamma \cup \{\neg\alpha\}$ is inconsistent. By Propositions 49B and 58B, $\Gamma \vdash \alpha$. □

6.9 The Compactness Theorem

One important consequence of the completeness theorem is the compactness theorem. The compactness theorem states that if each *finite* subset of a set of sentences is satisfiable, the entire set is satisfiable—even if the set itself is infinite. This is far from obvious. There is nothing that seems to rule out, at first glance at least, the possibility of there being infinite sets of sentences which are contradictory, but the contradiction only arises, so to speak, from the infinite number. The compactness theorem says that such a scenario can be ruled out: there are no unsatisfiable infinite sets of sentences each finite subset of which is satisfiable. Like the completeness theorem, it has a version related to entailment: if an infinite set of sentences entails something, already a finite subset does.

DEFINITION 69A A set Γ of wffs is *finitely satisfiable* if and only if every finite $\Gamma_0 \subseteq \Gamma$ is satisfiable.

THEOREM 69B (COMPACTNESS THEOREM) The following hold for any sentences Γ and α:

1. $\Gamma \vDash \alpha$ iff there is a finite $\Gamma_0 \subseteq \Gamma$ such that $\Gamma_0 \vDash \alpha$.

2. Γ is satisfiable if and only if it is finitely satisfiable.

Proof. We prove (2). If Γ is satisfiable, then there is a structure $\mathfrak{A}$ such that $\models_{\mathfrak{A}} \alpha$ for all $\alpha \in \Gamma$. Of course, this $\mathfrak{A}$ also satisfies every finite subset of Γ, so Γ is finitely satisfiable.

Now suppose that Γ is finitely satisfiable. Then every finite subset $\Gamma_0 \subseteq \Gamma$ is satisfiable. By soundness (**??** 5.11.3**??** 4.12.5), every finite subset is consistent. Then Γ itself must be consistent by Propositions 48H and 57H. By completeness (Theorem 68A), since Γ is consistent, it is satisfiable. □

Example 6.9.3. In every model $\mathfrak{A}$ of a theory Γ, each term t of course picks out an element of $|\mathfrak{A}|$. Can we guarantee that it is also true that every element of $|\mathfrak{A}|$ is picked out by some term or other? In other words, are there theories Γ all models of which are covered? The compactness theorem shows that this is not the case if Γ has infinite models. Here's how to see this: Let $\mathfrak{A}$ be an infinite model of Γ, and let c be a constant symbol not in the language of Γ. Let Δ be the set of all sentences $c \neq t$ for t a term in the language $\mathcal{L}$ of Γ, i.e.,

$$\Delta = \{c \neq t : t \in \mathrm{Trm}(\mathcal{L})\}.$$

A finite subset of $\Gamma \cup \Delta$ can be written as $\Gamma' \cup \Delta'$, with $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$. Since Δ' is finite, it can contain only finitely many terms. Let $a \in |\mathfrak{A}|$ be an element of $|\mathfrak{A}|$ not picked out by any of them, and let $\mathfrak{M}'$ be the structure that is just like $\mathfrak{A}$, but also $c^{\mathfrak{A}'} = a$. Since $a \neq t^{\mathfrak{A}}$ for all t occuring in Δ', $\models_{\mathfrak{A}'} \Delta'$. Since $\models_{\mathfrak{A}} \Gamma$, $\Gamma' \subseteq \Gamma$, and c does not occur in Γ, also $\models_{\mathfrak{A}'} \Gamma'$. Together, $\models_{\mathfrak{A}'} \Gamma' \cup \Delta'$ for every finite subset $\Gamma' \cup \Delta'$ of $\Gamma \cup \Delta$. So every finite subset of $\Gamma \cup \Delta$ is satisfiable. By compactness, $\Gamma \cup \Delta$ itself is satisfiable. So there are models $\models_{\mathfrak{A}} \Gamma \cup \Delta$. Every such $\mathfrak{A}$ is a model of Γ, but is not covered, since $c^{\mathfrak{A}} \neq t^{\mathfrak{A}}$ for all terms t of $\mathcal{L}$.

Example 6.9.4. Consider a language $\mathcal{L}$ containing the predicate symbol $<$, constant symbols 0, 1, and function symbols $+$, $\times$, $-$, $\div$. Let Γ be the set of all sentences in this language true in $\mathfrak{Q}$ with domain $\mathbb{Q}$ and the obvious interpretations. Γ is the set of all sentences of $\mathcal{L}$ true about the rational numbers. Of course, in $\mathbb{Q}$ (and even in $\mathbb{R}$), there are no numbers which are greater than 0 but less than $1/k$ for all $k \in \mathbb{Z}^+$. Such a number, if it existed, would be an *infinitesimal:* non-zero, but infinitely small. The compactness theorem shows that there are models of Γ in which infinitesimals exist: Let Δ be $\{0 < c\} \cup \{c < (1 \div \overline{k}) : k \in \mathbb{Z}^+\}$ (where $\overline{k} = (1 + (1 + \cdots + (1 + 1) \ldots))$ with k 1's). For any finite subset Δ_0 of Δ there is a K such that all the sentences $c < \overline{k}$ in Δ_0 have $k < K$. If we expand $\mathfrak{Q}$ to $\mathfrak{Q}'$ with $c^{\mathfrak{Q}'} = 1/K$ we have that $\models_{\mathfrak{Q}'} \Gamma \cup \Delta_0$, and so $\Gamma \cup \Delta$ is finitely satisfiable (Exercise: prove this in detail). By compactness, $\Gamma \cup \Delta$ is satisfiable. Any model $\mathfrak{S}$ of $\Gamma \cup \Delta$ contains an infinitesimal, namely $c^{\mathfrak{S}}$.

Example 6.9.5. We know that first-order logic with equality symbol can express that the size of the domain must have some minimal size: The sentence $\alpha_{\geq n}$

(which says "there are at least n distinct objects") is true only in structures where $|\mathfrak{A}|$ has at least n objects. So if we take

$$\Delta = \{\alpha_{\geq n} : n \geq 1\}$$

then any model of Δ must be infinite. Thus, we can guarantee that a theory only has infinite models by adding Δ to it: the models of $\Gamma \cup \Delta$ are all and only the infinite models of Γ.

So first-order logic can express infinitude. The compactness theorem shows that it cannot express finitude, however. For suppose some set of sentences Λ were satisfied in all and only finite structures. Then $\Delta \cup \Lambda$ is finitely satisfiable. Why? Suppose $\Delta' \cup \Lambda' \subseteq \Delta \cup \Lambda$ is finite with $\Delta' \subseteq \Delta$ and $\Lambda' \subseteq \Lambda$. Let n be the largest number such that $A_{\geq n} \in \Delta'$. Λ, being satisfied in all finite structures, has a model $\mathfrak{A}$ with finitely many but $\geq n$ elements. But then $\models_{\mathfrak{A}} \Delta' \cup \Lambda'$. By compactness, $\Delta \cup \Lambda$ has an infinite model, contradicting the assumption that Λ is satisfied only in finite structures.

6.10 A Direct Proof of the Compactness Theorem

We can prove the Compactness Theorem directly, without appealing to the Completeness Theorem, using the same ideas as in the proof of the completeness theorem. In the proof of the Completeness Theorem we started with a consistent set Γ of sentences, expanded it to a consistent, saturated, and complete set Γ^* of sentences, and then showed that in the term model $\mathfrak{M}(\mathfrak{d}^*)$ constructed from Γ^*, all sentences of Γ are true, so Γ is satisfiable.

We can use the same method to show that a finitely satisfiable set of sentences is satisfiable. We just have to prove the corresponding versions of the results leading to the truth lemma where we replace "consistent" with "finitely satisfiable."

PROPOSITION 610A Suppose Γ is complete and finitely satisfiable. Then:

1. $(\alpha \to \beta) \in \Gamma$ iff either $\alpha \notin \Gamma$ or $\beta \in \Gamma$.

LEMMA 610B Every finitely satisfiable set Γ can be extended to a saturated finitely satisfiable set Γ'.

PROPOSITION 610C Suppose Γ is complete, finitely satisfiable, and saturated. $\forall x\, \alpha(x) \in \Gamma$ iff $\alpha(t) \in \Gamma$ for all closed terms t.

LEMMA 610D Every finitely satisfiable set Γ' can be extended to a complete and finitely satisfiable set Γ^*.

THEOREM 610E (COMPACTNESS) Γ is satisfiable if and only if it is finitely satisfiable.

Proof. If Γ is satisfiable, then there is a structure $\mathfrak{A}$ such that $\models_{\mathfrak{A}} \alpha$ for all $\alpha \in \Gamma$. Of course, this $\mathfrak{A}$ also satisfies every finite subset of Γ, so Γ is finitely satisfiable.

Now suppose that Γ is finitely satisfiable. By Lemma 610B, there is a finitely satisfiable, saturated set $\Gamma' \supseteq \Gamma$. By Lemma 610D, Γ' can be extended to a complete and finitely satisfiable set Γ^*, and Γ^* is still saturated. Construct the term model $\mathfrak{M}(\mathfrak{d}^*)$ as in Definition 66A. Note that Proposition 66B did not rely on the fact that Γ^* is consistent (or complete or saturated, for that matter), but just on the fact that $\mathfrak{M}(\mathfrak{d}^*)$ is covered. The proof of the Truth Lemma (Lemma 66C) goes through if we replace references to Proposition 63B and Proposition 64E by references to Proposition 610A and Proposition 610C. □

6.11 The Löwenheim-Skolem Theorem

The Löwenheim-Skolem Theorem says that if a theory has an infinite model, then it also has a model that is at most denumerable. An immediate consequene of this fact is that first-order logic cannot express that the size of a structure is non-enumerable: any sentence or set of sentences satisfied in all non-enumerable structures is also satisfied in some enumerable structure.

THEOREM 611A If Γ is consistent then it has an enumerable model, i.e., it is satisfiable in a structure whose domain is either finite or denumerable.

Proof. If Γ is consistent, the structure $\mathfrak{A}$ delivered by the proof of the completeness theorem has a domain $|\mathfrak{A}|$ that is no larger than the set of the terms of the language $\mathcal{L}$. So $\mathfrak{A}$ is at most denumerable. □

THEOREM 611B If Γ is consistent set of sentences in the language of first-order logic without identity, then it has a denumerable model, i.e., it is satisfiable in a structure whose domain is infinite and enumerable.

Proof. If Γ is consistent and contains no sentences in which identity appears, then the structure $\mathfrak{A}$ delivered by the proof of the completness theorem has a domain $|\mathfrak{A}|$ identical to the set of terms of the language $\mathcal{L}'$. So $\mathfrak{A}$ is denumerable, since $\mathrm{Trm}(\mathcal{L}')$ is. □

Example 6.11.3 (Skolem's Paradox). Zermelo-Fraenkel set theory **ZFC** is a very powerful framework in which practically all mathematical statements can be expressed, including facts about the sizes of sets. So for instance, **ZFC** can prove that the set $\mathbb{R}$ of real numbers is non-enumerable, it can prove Cantor's Theorem that the power set of any set is larger than the set itself, etc. If **ZFC** is consistent, its models are all infinite, and moreover, they all contain elements about which the theory says that they are non-enumerable, such as the element that makes true the theorem of **ZFC** that the power set of the natural numbers exists. By the Löwenheim-Skolem Theorem, **ZFC** also has enumerable models—models that contain "non-enumerable" sets but which themselves are enumerable.

Chapter 7

Beyond First-order Logic

7.1 Overview

First-order logic is not the only system of logic of interest: there are many extensions and variations of first-order logic. A logic typically consists of the formal specification of a language, usually, but not always, a deductive system, and usually, but not always, an intended semantics. But the technical use of the term raises an obvious question: what do logics that are not first-order logic have to do with the word "logic," used in the intuitive or philosophical sense? All of the systems described below are designed to model reasoning of some form or another; can we say what makes them logical?

No easy answers are forthcoming. The word "logic" is used in different ways and in different contexts, and the notion, like that of "truth," has been analyzed from numerous philosophical stances. For example, one might take the goal of logical reasoning to be the determination of which statements are necessarily true, true a priori, true independent of the interpretation of the nonlogical terms, true by virtue of their form, or true by linguistic convention; and each of these conceptions requires a good deal of clarification. Even if one restricts one's attention to the kind of logic used in mathematics, there is little agreement as to its scope. For example, in the *Principia Mathematica*, Russell and Whitehead tried to develop mathematics on the basis of logic, in the *logicist* tradition begun by Frege. Their system of logic was a form of higher-type logic similar to the one described below. In the end they were forced to introduce axioms which, by most standards, do not seem purely logical (notably, the axiom of infinity, and the axiom of reducibility), but one might nonetheless hold that some forms of higher-order reasoning should be accepted as logical. In contrast, Quine, whose ontology does not admit "propositions" as legitimate objects of discourse, argues that second-order and higher-order logic are really manifestations of set theory in sheep's clothing; in other words, systems involving quantification over predicates are not purely logical.

For now, it is best to leave such philosophical issues for a rainy day, and simply think of the systems below as formal idealizations of various kinds of

reasoning, logical or otherwise.

7.2 Many-Sorted Logic

In first-order logic, variables and quantifiers range over a single domain. But it is often useful to have multiple (disjoint) domains: for example, you might want to have a domain of numbers, a domain of geometric objects, a domain of functions from numbers to numbers, a domain of abelian groups, and so on.

Many-sorted logic provides this kind of framework. One starts with a list of "sorts"—the "sort" of an object indicates the "domain" it is supposed to inhabit. One then has variables and quantifiers for each sort, and (usually) an equality symbol for each sort. Functions and relations are also "typed" by the sorts of objects they can take as arguments. Otherwise, one keeps the usual rules of first-order logic, with versions of the quantifier-rules repeated for each sort.

For example, to study international relations we might choose a language with two sorts of objects, French citizens and German citizens. We might have a unary relation, "drinks wine," for objects of the first sort; another unary relation, "eats wurst," for objects of the second sort; and a binary relation, "forms a multinational married couple," which takes two arguments, where the first argument is of the first sort and the second argument is of the second sort. If we use variables a, b, c to range over French citizens and x, y, z to range over German citizens, then

$$\forall a \, \forall x [(MarriedToax \to (DrinksWinea \vee \neg EatsWurstx)]]$$

asserts that if any French person is married to a German, either the French person drinks wine or the German doesn't eat wurst.

Many-sorted logic can be embedded in first-order logic in a natural way, by lumping all the objects of the many-sorted domains together into one first-order domain, using unary predicate symbols to keep track of the sorts, and relativizing quantifiers. For example, the first-order language corresponding to the example above would have unary predicate symbolss "$German$" and "$French$," in addition to the other relations described, with the sort requirements erased. A sorted quantifier $\forall x \, \alpha$, where x is a variable of the German sort, translates to

$$\forall x \, (Germanx \to \alpha).$$

We need to add axioms that insure that the sorts are separate—e.g., $\forall x \, \neg(Germanx \wedge Frenchx)$—as well as axioms that guarantee that "drinks wine" only holds of objects satisfying the predicate $Frenchx$, etc. With these conventions and axioms, it is not difficult to show that many-sorted sentences translate to first-order sentences, and many-sorted derivations translate to first-order derivations. Also, many-sorted structures "translate" to corresponding first-order structures and vice-versa, so we also have a completeness theorem for many-sorted logic.

7.3 Second-Order logic

The language of second-order logic allows one to quantify not just over a domain of individuals, but over relations on that domain as well. Given a first-order language $\mathcal{L}$, for each k one adds variables R which range over k-ary relations, and allows quantification over those variables. If R is a variable for a k-ary relation, and $t_1, \ldots, t_k$ are ordinary (first-order) terms, $Rt_1 \ldots t_k$ is an atomic wff. Otherwise, the set of wffs is defined just as in the case of first-order logic, with additional clauses for second-order quantification. Note that we only have the equality symbol for first-order terms: if R and S are relation variables of the same arity k, we can define $R = S$ to be an abbreviation for

$$\forall x_1 \ldots \forall x_k \, (Rx_1 \ldots x_k \leftrightarrow Sx_1 \ldots x_k).$$

The rules for second-order logic simply extend the quantifier rules to the new second order variables. Here, however, one has to be a little bit careful to explain how these variables interact with the predicate symbols of $\mathcal{L}$, and with wffs of $\mathcal{L}$ more generally. At the bare minimum, relation variables count as terms, so one has inferences of the form

$$\alpha(R) \vdash \exists R \, \alpha(R)$$

But if $\mathcal{L}$ is the language of arithmetic with a constant relation symbol $<$, one would also expect the following inference to be valid:

$$x < y \vdash \exists R \, Rxy$$

or for a given wff α,

$$\alpha(x_1, \ldots, x_k) \vdash \exists R \, Rx_1 \ldots x_k$$

More generally, we might want to allow inferences of the form

$$\alpha[\lambda \vec{x}. \, \beta(\vec{x})/R] \vdash \exists R \, \alpha$$

where $\alpha[\lambda \vec{x}. \, \beta(\vec{x})/R]$ denotes the result of replacing every atomic wff of the form $Rt_1, \ldots, t_k$ in α by $\beta(t_1, \ldots, t_k)$. This last rule is equivalent to having a *comprehension schema*, i.e., an axiom of the form

$$\exists R \, \forall x_1, \ldots, x_k \, (\alpha(x_1, \ldots, x_k) \leftrightarrow Rx_1 \ldots x_k),$$

one for each wff α in the second-order language, in which R is not a free variable. (Exercise: show that if R is allowed to occur in α, this schema is inconsistent!)

When logicians refer to the "axioms of second-order logic" they usually mean the minimal extension of first-order logic by second-order quantifier rules together with the comprehension schema. But it is often interesting to study weaker subsystems of these axioms and rules. For example, note that in its full generality the axiom schema of comprehension is *impredicative*: it allows

one to assert the existence of a relation $Rx_1 \dots x_k$ that is "defined" by a wff with second-order quantifiers; and these quantifiers range over the set of all such relations—a set which includes R itself! Around the turn of the twentieth century, a common reaction to Russell's paradox was to lay the blame on such definitions, and to avoid them in developing the foundations of mathematics. If one prohibits the use of second-order quantifiers in the wff α, one has a *predicative* form of comprehension, which is somewhat weaker.

From the semantic point of view, one can think of a second-order structure as consisting of a first-order structure for the language, coupled with a set of relations on the domain over which the second-order quantifiers range (more precisely, for each k there is a set of relations of arity k). Of course, if comprehension is included in the proof system, then we have the added requirement that there are enough relations in the "second-order part" to satisfy the comprehension axioms—otherwise the proof system is not sound! One easy way to insure that there are enough relations around is to take the second-order part to consist of *all* the relations on the first-order part. Such a structure is called *full*, and, in a sense, is really the "intended structure" for the language. If we restrict our attention to full structures we have what is known as the *full* second-order semantics. In that case, specifying a structure boils down to specifying the first-order part, since the contents of the second-order part follow from that implicitly.

To summarize, there is some ambiguity when talking about second-order logic. In terms of the proof system, one might have in mind either

1. A "minimal" second-order proof system, together with some comprehension axioms.

2. The "standard" second-order proof system, with full comprehension.

In terms of the semantics, one might be interested in either

1. The "weak" semantics, where a structure consists of a first-order part, together with a second-order part big enough to satisfy the comprehension axioms.

2. The "standard" second-order semantics, in which one considers full structures only.

When logicians do not specify the proof system or the semantics they have in mind, they are usually refering to the second item on each list. The advantage to using this semantics is that, as we will see, it gives us categorical descriptions of many natural mathematical structures; at the same time, the proof system is quite strong, and sound for this semantics. The drawback is that the proof system is *not* complete for the semantics; in fact, *no* effectively given proof system is complete for the full second-order semantics. On the other hand, we will see that the proof system *is* complete for the weakened semantics; this implies that if a sentence is not provable, then there is *some* structure, not necessarily the full one, in which it is false.

The language of second-order logic is quite rich. One can identify unary relations with subsets of the domain, and so in particular you can quantify over these sets; for example, one can express induction for the natural numbers with a single axiom

$$\forall R\,((R0 \wedge \forall x\,(Rx \to Rx')) \to \forall x\,Rx).$$

If one takes the language of arithmetic to have symbols $0, \prime, +, \times$ and $<$, one can add the following axioms to describe their behavior:

1. $\forall x\,\neg x' = 0$
2. $\forall x\,\forall y\,(s(x) = s(y) \to x = y)$
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. $\forall x\,\forall y\,(x < y \leftrightarrow \exists z\,y = (x + z'))$

It is not difficult to show that these axioms, together with the axiom of induction above, provide a categorical description of the structure $\mathfrak{B}$, the standard model of arithmetic, provided we are using the full second-order semantics. Given any structure $\mathfrak{A}$ in which these axioms are true, define a function f from $\mathbb{N}$ to the domain of $\mathfrak{A}$ using ordinary recursion on $\mathbb{N}$, so that $f(0) = 0^{\mathfrak{A}}$ and $f(x + 1) = \prime^{\mathfrak{A}}(f(x))$. Using ordinary induction on $\mathbb{N}$ and the fact that axioms (1) and (2) hold in $\mathfrak{A}$, we see that f is injective. To see that f is surjective, let P be the set of elements of $|\mathfrak{A}|$ that are in the range of f. Since $\mathfrak{A}$ is full, P is in the second-order domain. By the construction of f, we know that $0^{\mathfrak{A}}$ is in P, and that P is closed under $\prime^{\mathfrak{A}}$. The fact that the induction axiom holds in $\mathfrak{A}$ (in particular, for P) guarantees that P is equal to the entire first-order domain of $\mathfrak{A}$. This shows that f is a bijection. Showing that f is a homomorphism is no more difficult, using ordinary induction on $\mathbb{N}$ repeatedly.

In set-theoretic terms, a function is just a special kind of relation; for example, a unary function f can be identified with a binary relation R satisfying $\forall x\,\exists y\,R(x, y)$. As a result, one can quantify over functions too. Using the full semantics, one can then define the class of infinite structures to be the class of structures $\mathfrak{A}$ for which there is an injective function from the domain of $\mathfrak{A}$ to a proper subset of itself:

$$\exists f\,(\forall x\,\forall y\,(f(x) = f(y) \to x = y) \wedge \exists y\,\forall x\,f(x) \neq y).$$

The negation of this sentence then defines the class of finite structures.

In addition, one can define the class of well-orderings, by adding the following to the definition of a linear ordering:

$$\forall P\,(\exists x\,Px \to \exists x\,(Px \wedge \forall y\,(y < x \to \neg Py))).$$

This asserts that every non-empty set has a least element, modulo the identification of "set" with "one-place relation". For another example, one can express the notion of connectedness for graphs, by saying that there is no nontrivial separation of the vertices into disconnected parts:

$$\neg\exists A\,(\exists x\,A(x) \wedge \exists y\,\neg A(y) \wedge \forall w\,\forall z\,((Aw \wedge \neg Az) \rightarrow \neg Rwz)).$$

For yet another example, you might try as an exercise to define the class of finite structures whose domain has even size. More strikingly, one can provide a categorical description of the real numbers as a complete ordered field containing the rationals.

In short, second-order logic is much more expressive than first-order logic. That's the good news; now for the bad. We have already mentioned that there is no effective proof system that is complete for the full second-order semantics. For better or for worse, many of the properties of first-order logic are absent, including compactness and the Löwenheim-Skolem theorems.

On the other hand, if one is willing to give up the full second-order semantics in terms of the weaker one, then the minimal second-order proof system is complete for this semantics. In other words, if we read $\vdash$ as "proves in the minimal system" and $\models$ as "logically implies in the weaker semantics", we can show that whenever $\Gamma \models \alpha$ then $\Gamma \vdash \alpha$. If one wants to include specific comprehension axioms in the proof system, one has to restrict the semantics to second-order structures that satisfy these axioms: for example, if Δ consists of a set of comprehension axioms (possibly all of them), we have that if $\Gamma \cup \Delta \models \alpha$, then $\Gamma \cup \Delta \vdash \alpha$. In particular, if α is not provable using the comprehension axioms we are considering, then there is a model of $\neg\alpha$ in which these comprehension axioms nonetheless hold.

The easiest way to see that the completeness theorem holds for the weaker semantics is to think of second-order logic as a many-sorted logic, as follows. One sort is interpreted as the ordinary "first-order" domain, and then for each k we have a domain of "relations of arity k." We take the language to have built-in relation symbols "$true_k R x_1 \ldots x_k$" which is meant to assert that R holds of $x_1, \ldots, x_k$, where R is a variable of the sort "k-ary relation" and $x_1, \ldots, x_k$ are objects of the first-order sort.

With this identification, the weak second-order semantics is essentially the usual semantics for many-sorted logic; and we have already observed that many-sorted logic can be embedded in first-order logic. Modulo the translations back and forth, then, the weaker conception of second-order logic is really a form of first-order logic in disguise, where the domain contains both "objects" and "relations" governed by the appropriate axioms.

7.4 Higher-Order logic

Passing from first-order logic to second-order logic enabled us to talk about sets of objects in the first-order domain, within the formal language. Why stop there? For example, third-order logic should enable us to deal with sets of sets

of objects, or perhaps even sets which contain both objects and sets of objects. And fourth-order logic will let us talk about sets of objects of that kind. As you may have guessed, one can iterate this idea arbitrarily.

In practice, higher-order logic is often wffted in terms of functions instead of relations. (Modulo the natural identifications, this difference is inessential.) Given some basic "sorts" A, B, C, ... (which we will now call "types"), we can create new ones by stipulating

> If σ and τ are finite types then so is $\sigma \to \tau$.

Think of types as syntactic "labels," which classify the objects we want in our domain; $\sigma \to \tau$ describes those objects that are functions which take objects of type σ to objects of type τ. For example, we might want to have a type Ω of truth values, "true" and "false," and a type $\mathbb{N}$ of natural numbers. In that case, you can think of objects of type $\mathbb{N} \to \Omega$ as unary relations, or subsets of $\mathbb{N}$; objects of type $\mathbb{N} \to \mathbb{N}$ are functions from natural numers to natural numbers; and objects of type $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ are "functionals," that is, higher-type functions that take functions to numbers.

As in the case of second-order logic, one can think of higher-order logic as a kind of many-sorted logic, where there is a sort for each type of object we want to consider. But it is usually clearer just to define the syntax of higher-type logic from the ground up. For example, we can define a set of finite types inductively, as follows:

1. $\mathbb{N}$ is a finite type.
2. If σ and τ are finite types, then so is $\sigma \to \tau$.
3. If σ and τ are finite types, so is $\sigma \times \tau$.

Intuitively, $\mathbb{N}$ denotes the type of the natural numbers, $\sigma \to \tau$ denotes the type of functions from σ to τ, and $\sigma \times \tau$ denotes the type of pairs of objects, one from σ and one from τ. We can then define a set of terms inductively, as follows:

1. For each type σ, there is a stock of variables x, y, z, ... of type σ
2. 0 is a term of type $\mathbb{N}$
3. S (successor) is a term of type $\mathbb{N} \to \mathbb{N}$
4. If s is a term of type σ, and t is a term of type $\mathbb{N} \to (\sigma \to \sigma)$, then R_{st} is a term of type $\mathbb{N} \to \sigma$
5. If s is a term of type $\tau \to \sigma$ and t is a term of type τ, then $s(t)$ is a term of type σ
6. If s is a term of type σ and x is a variable of type τ, then $\lambda x.\, s$ is a term of type $\tau \to \sigma$.

7. If s is a term of type σ and t is a term of type τ, then $\langle s, t\rangle$ is a term of type $\sigma \times \tau$.

8. If s is a term of type $\sigma \times \tau$ then $p_1(s)$ is a term of type σ and $p_2(s)$ is a term of type τ.

Intuitively, R_{st} denotes the function defined recursively by

$$R_{st}(0) = s$$
$$R_{st}(x+1) = t(x, R_{st}(x)),$$

$\langle s, t\rangle$ denotes the pair whose first component is s and whose second component is t, and $p_1(s)$ and $p_2(s)$ denote the first and second elements ("projections") of s. Finally, $\lambda x.\, s$ denotes the function f defined by

$$f(x) = s$$

for any x of type σ; so item (6) gives us a form of comprehension, enabling us to define functions using terms. Wffs are built up from equality symbol statements $s = t$ between terms of the same type, the usual propositional connectives, and higher-type quantification. One can then take the axioms of the system to be the basic equations governing the terms defined above, together with the usual rules of logic with quantifiers and equality symbol.

If one augments the finite type system with a type Ω of truth values, one has to include axioms which govern its use as well. In fact, if one is clever, one can get rid of complex wffs entirely, replacing them with terms of type Ω! The proof system can then be modified accordingly. The result is essentially the *simple theory of types* set forth by Alonzo Church in the 1930s.

As in the case of second-order logic, there are different versions of higher-type semantics that one might want to use. In the full version, variables of type $\sigma \to \tau$ range over the set of *all* functions from the objects of type σ to objects of type τ. As you might expect, this semantics is too strong to admit a complete, effective proof system. But one can consider a weaker semantics, in which a structure consists of sets of elements T_τ for each type τ, together with appropriate operations for application, projection, etc. If the details are carried out correctly, one can obtain completeness theorems for the kinds of proof systems described above.

Higher-type logic is attractive because it provides a framework in which we can embed a good deal of mathematics in a natural way: starting with $\mathbb{N}$, one can define real numbers, continuous functions, and so on. It is also particularly attractive in the context of intuitionistic logic, since the types have clear "constructive" intepretations. In fact, one can develop constructive versions of higher-type semantics (based on intuitionistic, rather than classical logic) that clarify these constructive interpretations quite nicely, and are, in many ways, more interesting than the classical counterparts.

7.5 Intuitionistic Logic

In constrast to second-order and higher-order logic, intuitionistic first-order logic represents a restriction of the classical version, intended to model a more "constructive" kind of reasoning. The following examples may serve to illustrate some of the underlying motivations.

Suppose someone came up to you one day and announced that they had determined a natural number x, with the property that if x is prime, the Riemann hypothesis is true, and if x is composite, 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 here they seem to have reduced the problem to one of calculation, that is, to the determination of whether a specific number is prime or not.

What is the magic value of x? They describe it as follows: x is the natural number that is equal to 7 if the Riemann hypothesis is true, and 9 otherwise.

Angrily, you demand your money back. From a classical point of view, the description above does in fact determine a unique value of x; but what you really want is a value of x 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:

THEOREM 75A There are irrational numbers a and b such that a^b is rational.

Proof. 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 certainly 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 model a kind of reasoning where moves like the one in the first proof are disallowed. Proving the existence of an x satisfying $\alpha(x)$ means that you have to give a specific x, and a proof that it

satisfies α, like in the second proof. Proving that α or β holds requires that you can prove one or the other.

Formally speaking, intuitionistic first-order logic is what you get if you omit restrict a proof system for first-order logic in a certain way. Similarly, there are intuitionistic versions of second-order or higher-order logic. From the mathematical point of view, these are just formal deductive systems, but, as already noted, they are intended to model 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.

There is an informal constructive interpretation of the intuitionist connectives, usually known as the Brouwer-Heyting-Kolmogorov interpretation. It runs as follows: a proof of $\alpha \wedge \beta$ consists of a proof of α paired with a proof of β; a proof of $\alpha \vee \beta$ consists of either a proof of α, or a proof of β, where we have explicit information as to which is the case; a proof of $\alpha \to \beta$ consists of a procedure, which transforms a proof of α to a proof of β; a proof of $\forall x\, \alpha(x)$ consists of a procedure which returns a proof of $\alpha(x)$ for any value of x; and a proof of $\exists x\, \alpha(x)$ consists of a value of x, together with a proof that this value satisfies α. One can describe the interpretation in computational terms known as the "Curry-Howard isomorphism" or the "wffs-as-types paradigm": think of a wff as specifying a certain kind of data type, and proofs as computational objects of these data types that enable us to see that the corresponding wff is true.

Intuitionistic logic is often thought of as being classical logic "minus" the law of the excluded middle. This following theorem makes this more precise.

THEOREM 75B Intuitionistically, the following axiom schemata are equivalent:

1. $(\alpha \to \bot) \to \neg\alpha$.
2. $\alpha \vee \neg\alpha$
3. $\neg\neg\alpha \to \alpha$

Obtaining instances of one schema from either of the others is a good exercise in intuitionistic logic.

The first deductive systems for intuitionistic propositional logic, put forth as formalizations of Brouwer's intuitionism, are due, independently, to Kolmogorov, Glivenko, and Heyting. The first formalization of intuitionistic first-order logic (and parts of intuitionist mathematics) is due to Heyting. Though a number of classically valid schemata are not intuitionistically valid, many are.

The *double-negation translation* describes an important relationship between classical and intuitionist logic. It is defined inductively follows (think of α^N as the "intuitionist" translation of the classical wff α):

$$\begin{aligned}
\alpha^N &\equiv \neg\neg\alpha \quad \text{for atomic wffs } \alpha \\
(\alpha \wedge \beta)^N &\equiv (\alpha^N \wedge \beta^N) \\
(\alpha \vee \beta)^N &\equiv \neg\neg(\alpha^N \vee \beta^N) \\
(\alpha \to \beta)^N &\equiv (\alpha^N \to \beta^N) \\
(\forall x\, \alpha)^N &\equiv \forall x\, \alpha^N \\
(\exists x\, \alpha)^N &\equiv \neg\neg\exists x\, \alpha^N
\end{aligned}$$

Kolmogorov and Glivenko had versions of this translation for propositional logic; for predicate logic, it is due to Gödel and Gentzen, independently. We have

THEOREM 75C 1. $\alpha \leftrightarrow \alpha^N$ is provable classically

2. If α is provable classically, then α^N is provable intuitionistically.

We can now envision the following dialogue. Classical mathematician: "I've proved α!" Intuitionist mathematician: "Your proof isn't valid. What you've really proved is α^N." Classical mathematician: "Fine by me!" As far as the classical mathematician is concerned, the intuitionist is just splitting hairs, since the two are equivalent. But the intuitionist insists there is a difference.

Note that the above translation concerns pure logic only; it does not address the question as to what the appropriate *nonlogical* axioms are for classical and intuitionistic mathematics, or what the relationship is between them. But the following slight extension of the theorem above provides some useful information:

THEOREM 75D If Γ proves α classically, Γ^N proves α^N intuitionistically.

In other words, if α is provable from some hypotheses classically, then α^N is provable from their double-negation translations.

To show that a sentence or propositional wff is intuitionistically valid, all you have to do is provide a proof. But how can you show that it is not valid? For that purpose, we need a semantics that is sound, and preferrably complete. A semantics due to Kripke nicely fits the bill.

We can play the same game we did for classical logic: define the semantics, and prove soundness and completeness. It is worthwhile, however, to note the following distinction. In the case of classical logic, the semantics was the "obvious" one, in a sense implicit in the meaning of the connectives. Though one can provide some intuitive motivation for Kripke semantics, the latter does not offer the same feeling of inevitability. In addition, the notion of a classical structure is a natural mathematical one, so we can either take the notion of a structure to be a tool for studying classical first-order logic, or take classical first-order logic to be a tool for studying mathematical structures. In contrast,

Kripke structures can only be viewed as a logical construct; they don't seem to have independent mathematical interest.

A Kripke structure for a propositional langauge consists of a partial order $\mathrm{Mod}(P)$ with a least element, and an "monotone" assignment of propositional variables to the elements of $\mathrm{Mod}(P)$. The intuition is that the elements of $\mathrm{Mod}(P)$ represent "worlds," or "states of knowledge"; an element $p \geq q$ represents a "possible future state" of q; and the propositional variables assigned to p are the propositions that are known to be true in state p. The forcing relation $\mathfrak{P}, p \Vdash \alpha$ then extends this relationship to arbitrary wffs in the language; read $\mathfrak{P}, p \Vdash \alpha$ as "α is true in state p." The relationship is defined inductively, as follows:

1. $\mathfrak{P}, p \Vdash p_i$ iff p_i is one of the propositional variables assigned to p.
2. $\mathfrak{P}, p \nVdash \bot$.
3. $\mathfrak{P}, p \Vdash (\alpha \wedge \beta)$ iff $\mathfrak{P}, p \Vdash \alpha$ and $\mathfrak{P}, p \Vdash \beta$.
4. $\mathfrak{P}, p \Vdash (\alpha \vee \beta)$ iff $\mathfrak{P}, p \Vdash \alpha$ or $\mathfrak{P}, p \Vdash \beta$.
5. $\mathfrak{P}, p \Vdash (\alpha \to \beta)$ iff, whenever $q \geq p$ and $\mathfrak{P}, q \Vdash \alpha$, then $\mathfrak{P}, q \Vdash \beta$.

It is a good exercise to try to show that $\neg(p \wedge q) \to (\neg p \vee \neg q)$ is not intuitionistically valid, by cooking up a Kripke structure that provides a counterexample.

7.6 Modal Logics

Consider the following example of a conditional sentence:

> If Jeremy is alone in that room, then he is drunk and naked and dancing on the chairs.

This is an example of a conditional assertion that may be materially true but nonetheless misleading, since it seems to suggest that there is a stronger link between the antecedent and conclusion other than simply that either the antecedent is false or the consequent true. That is, the wording suggests that the claim is not only true in this particular world (where it may be trivially true, because Jeremy is not alone in the room), but that, moreover, the conclusion *would have* been true *had* the antecedent been true. In other words, one can take the assertion to mean that the claim is true not just in this world, but in any "possible" world; or that it is *necessarily* true, as opposed to just true in this particular world.

Modal logic was designed to make sense of this kind of necessity. One obtains modal propositional logic from ordinary propositional logic by adding a box operator; which is to say, if α is a wff, so is $\Box\alpha$. Intuitively, $\Box\alpha$ asserts that α is *necessarily* true, or true in any possible world. $\Diamond\alpha$ is usually taken to be an abbreviation for $\neg\Box\neg\alpha$, and can be read as asserting that α is *possibly* true. Of course, modality can be added to predicate logic as well.

Kripke structures can be used to provide a semantics for modal logic; in fact, Kripke first designed this semantics with modal logic in mind. Rather than restricting to partial orders, more generally one has a set of "possible worlds," P, and a binary "accessibility" relation Rxy between worlds. Intuitively, Rpq asserts that the world q is compatible with p; i.e., if we are "in" world p, we have to entertain the possibility that the world could have been like q.

Modal logic is sometimes called an "intensional" logic, as opposed to an "extensional" one. The intended semantics for an extensional logic, like classical logic, will only refer to a single world, the "actual" one; while the semantics for an "intensional" logic relies on a more elaborate ontology. In addition to structureing necessity, one can use modality to structure other linguistic constructions, reinterpreting $\Box$ and $\Diamond$ according to the application. For example:

1. In provability logic, $\Box\alpha$ is read "α is provable" and $\Diamond\alpha$ is read "α is consistent."
2. In epistemic logic, one might read $\Box\alpha$ as "I know α" or "I believe α."
3. In temporal logic, one can read $\Box\alpha$ as "α is always true" and $\Diamond\alpha$ as "α is sometimes true."

One would like to augment logic with rules and axioms dealing with modality. For example, the system **S4** consists of the ordinary axioms and rules of propositional logic, together with the following axioms:

$$\Box(\alpha \to \beta) \to (\Box\alpha \to \Box\beta)$$
$$\Box\alpha \to \alpha$$
$$\Box\alpha \to \Box\Box\alpha$$

as well as a rule, "from α conclude $\Box\alpha$." **S5** adds the following axiom:

$$\Diamond\alpha \to \Box\Diamond\alpha$$

Variations of these axioms may be suitable for different applications; for example, S5 is usually taken to characterize the notion of logical necessity. And the nice thing is that one can usually find a semantics for which the proof system is sound and complete by restricting the accessibility relation in the Kripke structures in natural ways. For example, **S4** corresponds to the class of Kripke structures in which the accessibility relation is reflexive and transitive. **S5** corresponds to the class of Kripke structures in which the accessibility relation is *universal*, which is to say that every world is accessible from every other; so $\Box\alpha$ holds if and only if α holds in every world.

7.7 Other Logics

As you may have gathered by now, it is not hard to design a new logic. You too can create your own a syntax, make up a deductive system, and fashion

a semantics to go with it. You might have to be a bit clever if you want the proof system to be complete for the semantics, and it might take some effort to convince the world at large that your logic is truly interesting. But, in return, you can enjoy hours of good, clean fun, exploring your logic's mathematical and computational properties.

Recent decades have witnessed a veritable explosion of formal logics. Fuzzy logic is designed to model reasoning about vague properties. Probabilistic logic is designed to model reasoning about uncertainty. Default logics and nonmonotonic logics are designed to model defeasible forms of reasoning, which is to say, "reasonable" inferences that can later be overturned in the face of new information. There are epistemic logics, designed to model reasoning about knowledge; causal logics, designed to model reasoning about causal relationships; and even "deontic" logics, which are designed to model reasoning about moral and ethical obligations. Depending on whether the primary motivation for introducing these systems is philosophical, mathematical, or computational, you may find such creatures studies under the rubric of mathematical logic, philosophical logic, artificial intelligence, cognitive science, or elsewhere.

The list goes on and on, and the possibilities seem endless. We may never attain Leibniz' dream of reducing all of human reason to calculation—but that can't stop us from trying.