# Part I

# Incompleteness

Material in this part covers the incompleteness theorems. It depends on material in the parts on first-order logic (esp., the proof system), the material on recursive functions (in the computability part). It is based on Jeremy Avigad's notes with revisions by Richard Zach.

# Chapter 1

# Introduction to Incompleteness

## 1.1 Historical Background

inc:int:bgr:sec

In this section, we will briefly discuss historical developments that will help put the incompleteness theorems in context. In particular, we will give a very sketchy overview of the history of mathematical logic; and then say a few words about the history of the foundations of mathematics.

digression

The phrase "mathematical logic" is ambiguous. One can interpret the word "mathematical" as describing the subject matter, as in, "the logic of mathematics," denoting the principles of mathematical reasoning; or as describing the methods, as in "the mathematics of logic," denoting a mathematical study of the principles of reasoning. The account that follows involves mathematical logic in both senses, often at the same time.

The study of logic began, essentially, with Aristotle, who lived approximately 384–322 BCE. His *Categories*, *Prior analytics*, and *Posterior analytics* include systematic studies of the principles of scientific reasoning, including a thorough and systematic study of the syllogism.

Aristotle's logic dominated scholastic philosophy through the middle ages; indeed, as late as eighteenth century Kant maintained that Aristotle's logic was perfect and in no need of revision. But the theory of the syllogism is far too limited to model anything but the most superficial aspects of mathematical reasoning. A century earlier, Leibniz, a contemporary of Newton's, imagined a complete "calculus" for logical reasoning, and made some rudimentary steps towards designing such a calculus, essentially describing a version of propositional logic.

The nineteenth century was a watershed for logic. In 1854 George Boole wrote *The Laws of Thought*, with a thorough algebraic study of propositional logic that is not far from modern presentations. In 1879 Gottlob Frege published his *Begriffsschrift* (Concept writing) which extends propositional logic with quantifiers and relations, and thus includes first-order logic. In fact, Frege's logical systems included higher-order logic as well, and more. In his *Basic Laws of Arithmetic*, Frege set out to show that all of arithmetic could

be derived in his Begriffsschrift from purely logical assumption. Unfortunately, these assumptions turned out to be inconsistent, as Russell showed in 1902. But setting aside the inconsistent axiom, Frege more or less invented modern logic singlehandedly, a startling achievement. Quantificational logic was also developed independently by algebraically-minded thinkers after Boole, including Peirce and Schröder.

Let us now turn to developments in the foundations of mathematics. Of course, since logic plays an important role in mathematics, there is a good deal of interaction with the developments I just described. For example, Frege developed his logic with the explicit purpose of showing that all of mathematics could be based solely on his logical framework; in particular, he wished to show that mathematics consists of a priori *analytic* truths instead of, as Kant had maintained, a priori *synthetic* ones.

Many take the birth of mathematics proper to have occurred with the Greeks. Euclid's *Elements*, written around 300 B.C., is already a mature representative of Greek mathematics, with its emphasis on rigor and precision. The definitions and proofs in Euclid's *Elements* survive more or less in tact in high school geometry textbooks today (to the extent that geometry is still taught in high schools). This model of mathematical reasoning has been held to be a paradigm for rigorous argumentation not only in mathematics but in branches of philosophy as well. (Spinoza even presented moral and religious arguments in the Euclidean style, which is strange to see!)

Calculus was invented by Newton and Leibniz in the seventeenth century. (A fierce priority dispute raged for centuries, but most scholars today hold that the two developments were for the most part independent.) Calculus involves reasoning about, for example, infinite sums of infinitely small quantities; these features fueled criticism by Bishop Berkeley, who argued that belief in God was no less rational than the mathematics of his time. The methods of calculus were widely used in the eighteenth century, for example by Leonhard Euler, who used calculations involving infinite sums with dramatic results.

In the nineteenth century, mathematicians tried to address Berkeley's criticisms by putting calculus on a firmer foundation. Efforts by Cauchy, Weierstrass, Bolzano, and others led to our contemporary definitions of limits, continuity, differentiation, and integration in terms of "epsilons and deltas," in other words, devoid of any reference to infinitesimals. Later in the century, mathematicians tried to push further, and explain all aspects of calculus, including the real numbers themselves, in terms of the natural numbers. (Kronecker: "God created the whole numbers, all else is the work of man.") In 1872, Dedekind wrote "Continuity and the irrational numbers," where he showed how to "construct" the real numbers as sets of rational numbers (which, as you know, can be viewed as pairs of natural numbers); in 1888 he wrote "Was sind und was sollen die Zahlen" (roughly, "What are the natural numbers, and what should they be?") which aimed to explain the natural numbers in purely "logical" terms. In 1887 Kronecker wrote "Über den Zahlbegriff" ("On the concept of number") where he spoke of representing all mathematical object in terms of the integers; in 1889 Giuseppe Peano gave formal, symbolic axioms

for the natural numbers.

The end of the nineteenth century also brought a new boldness in dealing with the infinite. Before then, infinitary objects and structures (like the set of natural numbers) were treated gingerly; "infinitely many" was understood as "as many as you want," and "approaches in the limit" was understood as "gets as close as you want." But Georg Cantor showed that it was possible to take the infinite at face value. Work by Cantor, Dedekind, and others help to introduce the general set-theoretic understanding of mathematics that is now widely accepted.

This brings us to twentieth century developments in logic and foundations. In 1902 Russell discovered the paradox in Frege's logical system. In 1904 Zermelo proved Cantor's well-ordering principle, using the so-called "axiom of choice"; the legitimacy of this axiom prompted a good deal of debate. Between 1910 and 1913 the three volumes of Russell and Whitehead's *Principia Mathematica* appeared, extending the Fregean program of establishing mathematics on logical grounds. Unfortunately, Russell and Whitehead were forced to adopt two principles that seemed hard to justify as purely logical: an axiom of infinity and an axiom of "reducibility." In the 1900's Poincaré criticized the use of "impredicative definitions" in mathematics, and in the 1910's Brouwer began proposing to refound all of mathematics in an "intuitionistic" basis, which avoided the use of the law of the excluded middle ($\varphi \vee \neg\varphi$).

Strange days indeed! The program of reducing all of mathematics to logic is now referred to as "logicism," and is commonly viewed as having failed, due to the difficulties mentioned above. The program of developing mathematics in terms of intuitionistic mental constructions is called "intuitionism," and is viewed as posing overly severe restrictions on everyday mathematics. Around the turn of the century, David Hilbert, one of the most influential mathematicians of all time, was a strong supporter of the new, abstract methods introduced by Cantor and Dedekind: "no one will drive us from the paradise that Cantor has created for us." At the same time, he was sensitive to foundational criticisms of these new methods (oddly enough, now called "classical"). He proposed a way of having one's cake and eating it too:

1. Represent classical methods with formal axioms and rules; represent mathematical questions as formulas in an axiomatic system.

2. Use safe, "finitary" methods to prove that these formal deductive systems are consistent.

Hilbert's work went a long way toward accomplishing the first goal. In 1899, he had done this for geometry in his celebrated book *Foundations of geometry*. In subsequent years, he and a number of his students and collaborators worked on other areas of mathematics to do what Hilbert had done for geometry. Hilbert himself gave axiom systems for arithmetic and analysis. Zermelo gave an axiomatization of set theory, which was expanded on by Fraenkel, Skolem, von Neumann, and others. By the mid-1920s, there were two approaches that

laid claim to the title of an axiomatization of "all" of mathematics, the *Principia mathematica* of Russell and Whitehead, and what came to be known as Zermelo-Fraenkel set theory.

In 1921, Hilbert set out on a research project to establish the goal of proving these systems to be consistent. He was aided in this project by several of his students, in particular Bernays, Ackermann, and later Gentzen. The basic idea for accomplishing this goal was to cast the question of the possibility of a derivation of an inconsistency in mathmatics as a combinatorial problem about possible sequences of symbols, namely possible sequences of sentences which meet the criterion of being a correct derivation of, say, $\varphi \wedge \neg\varphi$ from the axioms of an axiom system for arithmetic, analysis, or set theory. A proof of the impossibility of such a sequence of symbols would—since it is itself a mathematical proof—be formalizable in these axiomatic systems. In other words, there would be some sentence $\mathsf{Con}$ which states that, say, arithmetic is consistent. Moreover, this sentence should be provable in the systems in question, especially if its proof requires only very restricted, "finitary" means.

The second aim, that the axiom systems developed would settle every mathematical question, can be made precise in two ways. In one way, we can formulate it as follows: For any sentence $\varphi$ in the language of an axiom system for mathematics, either $\varphi$ or $\neg\varphi$ is provable from the axioms. If this were true, then there would be no sentences which can neither be proved nor refuted on the basis of the axioms, no questions which the axioms do not settle. An axiom system with this property is called *complete*. Of course, for any given sentence it might still be a difficult task to determine which of the two alternatives holds. But in principle there should be a method to do so. In fact, for the axiom and derivation systems considered by Hilbert, completeness would imply that such a method exists—although Hilbert did not realize this. The second way to interpret the question would be this stronger requirement: that there be a mechanical, computational method which would determine, for a given sentence $\varphi$, whether it is derivable from the axioms or not.

In 1931, Gödel proved the two "incompleteness theorems," which showed that this program could not succeed. There is no axiom system for mathematics which is complete, specifically, the sentence that expresses the consistency of the axioms is a sentence which can neither be proved nor refuted.

This struck a lethal blow to Hilbert's original program. However, as is so often the case in mathematics, it also opened up exciting new avenues for research. If there is no one, all-encompassing formal system of mathematics, it makes sense to develop more circumscribesd systems and investigate what can be proved in them. It also makes sense to develop less restricted methods of proof for establishing the consistency of these systems, and to find ways to measure how hard it is to prove their consistency. Since Gödel showed that (almost) every formal system has questions it cannot settle, it makes sense to look for "interesting" questions a given formal system cannot settle, and to figure out how strong a formal system has to be to settle them. To the present day, logicians have been pursuing these questions in a new mathematical discipline, the theory of proofs.

## 1.2 Definitions

inc:int:def:sec

In order to carry out Hilbert's project of formalizing mathematics and showing that such a formalization is consistent and complete, the first order of business would be that of picking a language, logical framework, and a system of axioms. For our purposes, let us suppose that mathematics can be formalized in a first-order language, i.e., that there is some set of constant symbols, function symbols, and predicate symbols which, together with the connectives and quatifiers of first-order logic, allow us to express the claims of mathematics. Most people agree that such a language exists: the language of set theory, in which $\in$ is the only non-logical symbol. That such a simple language is so expressive is of course a very implausible claim at first sight, and it took a lot of work to establish that practically of all mathematics can be expressed in this very austere vocabulary. To keep things simple, for now, let's restrict our discussion to arithmetic, so the part of mathematics that just deals with the natural numbers $\mathbb{N}$. The natural language in which to express facts of arithmetic is $\mathcal{L}_A$. $\mathcal{L}_A$ contains a single two-place predicate symbol $<$, a single constant symbol $\mathsf{0}$, one one-place function symbol $\prime$, and two two-place function symbols $+$ and $\times$.

**Definition 1.1.** A set of sentences $\Gamma$ is a *theory* if it is closed under entailment, i.e., if $\Gamma = \{\varphi : \Gamma \vDash \varphi\}$.

There are two easy ways to specify theories. One is as the set of sentences true in some structure. For instance, consider the structure for $\mathcal{L}_A$ in which the domain is $\mathbb{N}$ and all non-logical symbols are interpreted as you would expect.

**Definition 1.2.** The *standard model of arithmetic* is the structure $\mathfrak{N}$ defined as follows:

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

Note the difference between $\times$ and $\cdot$: $\times$ is a symbol in the language of arithmetic. Of course, we've chosen it to remind us of multiplication, but $\times$ is not the multiplication operation but a two-place function symbol (officially, $f^2_1$. By contrast, $\cdot$ *is* the ordinary multiplication function. When you see something like $n \cdot m$, we mean the product of the numbers $n$ and $m$; when you see something like $x \times y$ we are talking about a term in the language of arithmetic. In the standard model, the function symbol times is interpreted as the function $\cdot$ on the natural numbers. For addition, we use $+$ as both the

function symbol of the language of arithmetic, and the addition function on the natural numbers. Here you have to use the context to determine what is meant.

**Definition 1.3.** The theory of *true arithmetic* is the set of sentences satisfied in the standard model of arithmetic, i.e.,

$$\mathbf{TA} = \{\varphi : \mathfrak{N} \vDash \varphi\}.$$

**TA** is a theory, for whenever $\mathbf{TA} \vDash \varphi$, $\varphi$ is satisfied in every structure which satisfies **TA**. Since $\mathfrak{M} \vDash \mathbf{TA}$, $\mathfrak{M} \vDash \varphi$, and so $\varphi \in \mathbf{TA}$.

The other way to specify a theory $\Gamma$ is as the set of sentences entailed by some set of sentences $\Gamma_0$. In that case, $\Gamma$ is the "closure" of $\Gamma_0$ under entailment. Specifying a theory this way is only interesting if $\Gamma_0$ is explicitly specified, e.g., if the elements of $\Gamma_0$ are listed. At the very least, $\Gamma_0$ has to be decidable, i.e., there has to be a computable test for when a sentence counts as an element of $\Gamma_0$ or not. We call the sentences in $\Gamma_0$ *axioms* for $\Gamma$, and $\Gamma$ *axiomatized* by $\Gamma_0$.

**Definition 1.4.** A theory $\Gamma$ is *axiomatized* by $\Gamma_0$ iff

$$\Gamma = \{\varphi : \Gamma_0 \vDash \varphi\}$$

**Definition 1.5.** The theory **Q** axiomatized by the following sentences is known as "Robinson's **Q**" and is a very simple theory of arithmetic.

$$\forall x\, \forall y\, (x' = y' \to x = y) \tag{$Q_1$}$$
$$\forall x\, \mathsf{o} \neq x' \tag{$Q_2$}$$
$$\forall x\, (x \neq \mathsf{o} \to \exists y\, x = y') \tag{$Q_3$}$$
$$\forall x\, (x + \mathsf{o}) = x \tag{$Q_4$}$$
$$\forall x\, \forall y\, (x + y') = (x + y)' \tag{$Q_5$}$$
$$\forall x\, (x \times \mathsf{o}) = \mathsf{o} \tag{$Q_6$}$$
$$\forall x\, \forall y\, (x \times y') = ((x \times y) + x) \tag{$Q_7$}$$
$$\forall x\, \forall y\, (x < y \leftrightarrow \exists z\, (x + z' = y)) \tag{$Q_8$}$$

The set of sentences $\{Q_1, \ldots, Q_8\}$ are the axioms of **Q**, so **Q** consists of all sentences entailed by them:

$$\mathbf{Q} = \{\varphi : \{Q_1, \ldots, Q_8\} \vDash \varphi\}.$$

**Definition 1.6.** Suppose $\varphi(x)$ is a formula in $\mathcal{L}_A$ with free variables $x$ and $y_1$, $\ldots$, $y_n$. Then any sentence of the form

$$\forall y_1\, \ldots \forall y_n\, ((\varphi(\mathsf{o}) \wedge \forall x\, (\varphi(x) \to \varphi(x'))) \to \forall x\, \varphi(x))$$

is an instance of the *induction schema*.

*Peano arithmetic* **PA** is the theory axiomatized by the axioms of **Q** together with all instances of the induction schema.

Every instance of the induction schema is true in $\mathfrak{N}$. This is easiest to see if the formula $\varphi$ only has one free variable $x$. Then $\varphi(x)$ defines a subset $X_A$ of $\mathbb{N}$ in $\mathfrak{N}$. $X_A$ is the set of all $n \in \mathbb{N}$ such that $\mathfrak{N}, s \vDash \varphi(x)$ when $s(x) = n$. The corresponding instance of the induction schema is explanation

$$((\varphi(\mathsf{o}) \wedge \forall x\,(\varphi(x) \to \varphi(x'))) \to \forall x\,\varphi(x))$$

If its antecedent is true in $\mathfrak{N}$, then $0 \in X_A$ and, whenever $n \in X_A$, so is $n+1$. Since $0 \in X_A$, we get $1 \in X_A$. With $1 \in X_A$ we get $2 \in X_A$. And so on. So for every $n \in \mathbb{N}$, $n \in X_A$. But this means that $\forall x\,\varphi(x)$ is satisfied in $\mathfrak{N}$.

Both **Q** and **PA** are axiomatized theories. The big question is, how strong are they? For instance, can **PA** prove all the truths about $\mathbb{N}$ that can be expressed in $\mathcal{L}_A$? Specifically, do the axioms of **PA** settle all the questions that can be formulated in $\mathcal{L}_A$?

Another way to put this is to ask: Is $\mathbf{PA} = \mathbf{TA}$? **TA** obviously does prove (i.e., it includes) all the truths about $\mathbb{N}$, and it settles all the questions that can be formulated in $\mathcal{L}_A$, since if $\varphi$ is a sentence in $\mathcal{L}_A$, then either $\mathfrak{N} \vDash \varphi$ or $\mathfrak{N} \vDash \neg\varphi$, and so either $\mathbf{TA} \vDash \varphi$ or $\mathbf{TA} \vDash \neg\varphi$. Call such a theory complete.

**Definition 1.7.** A theory $\Gamma$ is *complete* iff for every sentence $\varphi$ in its language, either $\Gamma \vDash \varphi$ or $\Gamma \vDash \neg\varphi$.

By the Completeness Theorem, $\Gamma \vDash \varphi$ iff $\Gamma \vdash \varphi$, so $\Gamma$ is complete iff for every sentence $\varphi$ in its language, either $\Gamma \vdash \varphi$ or $\Gamma \vdash \neg\varphi$. explanation

Another question we are led to ask is this: Is there a computational procedure we can use to test if a sentence is in **TA**, in **PA**, or even just in **Q**? We can make this more precise by defining when a set (e.g., a set of sentences) is decidable.

**Definition 1.8.** A set $X$ is *decidable* iff there is a computational procedure which on input $x$ returns 1 if $x \in X$ and 0 otherwise.

So our question becomes: Is **TA** (**PA**, **Q**) decidable?

The answer to all these questions will be: no. None of these theories are decidable. However, this phenomenon is not specific to these particular theories. In fact, *any* theory that satisfies certain conditions is subject to the same results. One of these conditions, which **Q** and **PA** satisfy, is that they are axiomatized by a decidable set of axioms.

**Definition 1.9.** A theory is *axiomatizable* if it is axiomatized by a decidable set of axioms.

**Example 1.10.** Any theory axiomatized by a finite set of sentences is axiomatizable, since any finite set is decidable. Thus, **Q**, for instance, is axiomatizable.

Schematically axiomatized theories like **PA** are also axiomatizable. For to test if $\psi$ is among the axioms of **PA**, i.e., to compute the function $\chi_X$ where $\chi_X(\psi) = 1$ if $\psi$ is an axiom of **PA** and $= 0$ otherwise, we can do the following: First, check if $\psi$ is one of the axioms of **Q**. If it is, the answer is "yes" and the

value of $\chi_X(\psi) = 1$. If not, test if it is an instance of the induction schema. This can be done systematically; in this case, perhaps it's easiest to see that it can be done as follows: Any instance of the induction schema begins with a number of universal quantifiers, and then a sub-formula that is a conditional. The consequent of that conditional is $\forall x\, \varphi(x, y_1, \ldots, y_n)$ where $x$ and $y_1, \ldots, y_n$ are all the free variables of $\varphi$ and the initial quantifiers of $\psi$ bind the variables $y_1, \ldots, y_n$. Once we have extracted this $\varphi$ and checked that its free variables match the variables bound by the universal qauntifiers at the front and $\forall x$, we go on to check that the antecedent of the conditional matches

$$\varphi(\mathsf{0}, y_1, \ldots, y_n) \wedge \forall x\, (\varphi(x, y_1, \ldots, y_n) \to \varphi(x', y_1, \ldots, y_n))$$

Again, if it does, $\psi$ is an instance of the induction schema, and if it doesn't, $\psi$ isn't.

In answering this question—and the more general question of which theories are complete or decidable—it will be useful to consider also the following definition. Recall that a set $X$ is enumerable iff it is empty or if there is a surjective function $f\colon \mathbb{N} \to X$. Such a function is called an enumeration of $X$.

**Definition 1.11.** A set $X$ is called *computably enumerable* (c.e. for short) iff it is empty or it has a computable enumeration.

In addition to axiomatizability, another condition on theories to which the incompleteness theorems apply will be that they are strong enough to prove basic facts about computable functions and decidable relations. By "basic facts," we mean sentences which express what the values of computable functions are for each of their arguments. And by "strong enough" we mean that the theories in question count these sentences among its theorems. For instance, consider a prototypical computable function: addition. The value of $+$ for arguments 2 and 3 is 5, i.e., $2 + 3 = 5$. A sentence in the language of arithmetic that expresses that the value of $+$ for arguments 2 and 3 is 5 is: $(\overline{2} + \overline{3}) = \overline{5}$. And, e.g., **Q** proves this sentence. More generally, we would like there to be, for each computable function $f(x_1, x_2)$ a formula $\varphi_f(x_1, x_2, y)$ in $\mathcal{L}_A$ such that $\mathbf{Q} \vdash \varphi_f(\overline{n_1}, \overline{n_2}, \overline{m})$ whenever $f(n_1, n_2) = m$. In this way, **Q** proves that the value of $f$ for arguments $n_1$, $n_2$ is $m$. In fact, we require that it proves a bit more, namely that no other number is the value of $f$ for arguments $n_1, n_2$. And the same goes for decidable relations. This is made precise in the following two definitions.

**Definition 1.12.** A formula $\varphi(x_1, \ldots, x_k, y)$ *represents* the function $f\colon \mathbb{N}^k \to \mathbb{N}$ in $\Gamma$ iff whenever $f(n_1, \ldots, n_k) = m$, then

1. $\Gamma \vdash \varphi(\overline{n_1}, \ldots, \overline{n_k}, \overline{m})$, and
2. $\Gamma \vdash \forall y(\varphi(\overline{n_1}, \ldots, \overline{n_k}, y) \to y = \overline{m})$.

**Definition 1.13.** A formula $\varphi(x_1, \ldots, x_k)$ *represents* the relation $R \subseteq \mathbb{N}^k$ iff,

1. whenever $R(n_1, \ldots, n_k)$, $\Gamma \vdash \varphi(\overline{n_1}, \ldots, \overline{n_k})$, and
2. whenever not $R(n_1, \ldots, n_k)$, $\Gamma \vdash \neg\varphi(\overline{n_1}, \ldots, \overline{n_k})$.

A theory is "strong enough" for the incompleteness theorems to apply if it represents all computable functions and all decidable relations. **Q** and its extensions satisfy this condition, but it will take us a while to establish this—it's a non-trivial fact about the kinds of things **Q** can prove, and it's hard to show because **Q** has only a few axioms from which we'll have to prove all these facts. However, **Q** is a very weak theory. So although it's hard to prove that **Q** represents all computable functions, most interesting theories are stronger than **Q**, i.e., prove more than **Q** does. And if **Q** proves something, any stronger theory does; since **Q** represents all computable functions, every stronger theory does. This means that many interesting theories meet this condition of the incompleteness theorems. So our hard work will pay off, since it shows that the incompletess theorems apply to a wide range of theories. Certainly, any theory aiming to formalize "all of mathematics" must prove everything that **Q** proves, since it should at the very least be able to capture the results of elementary computations. So any theory that is a candidate for a theory of "all of mathematics" will be one to which the incompleteness theorems apply.

## 1.3 Overview of Incompleteness Results

inc:int:ovr: sec

Hilbert expected that mathematics could be formalized in an axiomatizable theory which it would be possible to prove complete and decidable. Moreover, he aimed to prove the consistency of this theory with very weak, "finitary," means, which would defend classical mathematics agianst the challenges of intuitionism. Gödel's incompleteness theorems showed that these goals cannot be achieved.

Gödel's first incompleteness theorem showed that a version of Russell and Whitehead's *Principia Mathematica* is not complete. But the proof was actually very general and applies to a wide variety of theories. This means that it wasn't just that *Principia Mathematica* did not manage to completely capture mathematics, but that *no* acceptable theory does. It took a while to isolate the features of theories that suffice for the incompleteness theorems to apply, and to generalize Gödel's proof to apply make it depend only on these features. But we are now in a position to state a very general version of the first incompleteness theorem for theories in the language $\mathcal{L}_A$ of arithmetic.

**Theorem 1.14.** *If $\Gamma$ is a consistent and axiomatizable theory in $\mathcal{L}_A$ which represents all computable functions and decidable relations, then $\Gamma$ is not complete.*

To say that $\Gamma$ is not complete is to say that for at least one sentence $\varphi$, $\Gamma \nvdash \varphi$ and $\Gamma \nvdash \neg\varphi$. Such a sentence is called *independent* (of $\Gamma$). We can in fact relatively quickly prove that there must be independent sentences. But the power of Gödel's proof of the theorem lies in the fact that it exhibits a

*specific example* of such an independent sentence. The intriguing construction produces a sentence $G_\Gamma$, called a *Gödel sentence* for $\Gamma$, which is unprovable because in $\Gamma$, $G_\Gamma$ is equivalent to the claim that $G_\Gamma$ is unprovable in $\Gamma$. It does so *constructively*, i.e., given an axiomatization of $\Gamma$ and a description of the proof system, the proof gives a method for actually writing down $G_\Gamma$.

The construction in Gödel's proof requires that we find a way to express in $\mathcal{L}_A$ the properties of and operations on terms and formulas of $\mathcal{L}_A$ itself. These include properties such as "$\varphi$ is a sentence," "$\delta$ is a derivation of $\varphi$," and operations such as $\varphi[t/x]$. This way must (a) express these properties and relations via a "coding" of symbols and sequences thereof (which is what terms, formulas, derivations, etc. are) as natural numbers (which is what $\mathcal{L}_A$ can talk about). It must (b) do this in such a way that $\Gamma$ will prove the relevant facts, so we must show that these properties are coded by decidable properties of natural numbers and the operations correspond to computable functions on natural numbers. This is called "arithmetization of syntax."

Before we investigate how syntax can be arithmetized, however, we will consider the condition that $\Gamma$ is "strong enough," i.e., represents all computable functions and decidable relations. This requires that we give a precise definition of "computable." This can be done in a number of ways, e.g., via the model of Turing machines, or as those functions computable by programs in some general-purpose programming language. Since our aim is to represent these functions and relations in a theory in the language $\mathcal{L}_A$, however, it is best to pick a simple definition of computability of just numerical functions. This is the notion of *recursive function*. So we will first discuss the recursive functions. We will then show that **Q** already represents all recursive functions and relations. This will allow us to apply the incompleteness theorem to specific theories such as **Q** and **PA**, since we will have established that these are examples of theories that are "strong enough."

The end result of the arithmetization of syntax is a formula $\mathsf{Prov}_\Gamma(x)$ which, via the coding of formulas as numbers, expresses provability from the axioms of $\Gamma$. Specifically, if $\varphi$ is coded by the number $n$, and $\Gamma \vdash \varphi$, then $\Gamma \vdash \mathrm{Prov}_\Gamma(\overline{n})$. This "provability predicate" for $\Gamma$ allows us also to express, in a certain sense, the consistency of $\Gamma$ as a sentence of $\mathcal{L}_A$: let the "consistency statemetn" for $\Gamma$ be the sentence $\neg\mathrm{Prov}_\Gamma(\overline{n})$, where we take $n$ to be the code of a contradiction, e.g., of $\bot$. The second incompleteness theorem states that consistent axiomatizable theories also do not prove their own consistency statements. The conditions required for this theorem to apply are a bit more stringent than just that the theory represents all computable functions and decidable relations, but we will show that **PA** satisifes them.

## 1.4 Undecidability and Incompleteness

inc:int:dec:sec

Gödel's proof of the incompleteness theorems require arithmetization of syntax. But even without that we can obtain some nice results just on the assumtion

that a theory represents all decidable relations. The proof is a diagonal argument similar to the proof of the undecidability of the halting problem.

**Theorem 1.15.** *If $\Gamma$ is a consistent theory that represents every decidable relation, then $\Gamma$ is not decidable.*

*Proof.* Suppose $\Gamma$ were decidable. We show that if $\Gamma$ represents every decidable relation, it must be inconsistent.

Decidable properties (one-place relations) are represented by formulas with one free variable. Let $\varphi_0(x)$, $\varphi_1(x)$, ..., be a computable enumeration of all such formulas. Now consider the following set $D \subseteq \mathbb{N}$:

$$D = \{n : \Gamma \vdash \neg\varphi_n(\overline{n})\}$$

The set $D$ is decidable, since we can test if $n \in D$ by first computing $\varphi_n(x)$, and from this $\neg\varphi_n(\overline{n})$. Obviously, substituting the term $\overline{n}$ for every free occurrence of $x$ in $\varphi_n(x)$ and prefixing $\varphi(\overline{n})$ by $\neg$ is a mechanical matter. By assumption, $\Gamma$ is decidable, so we can test if $\neg\varphi(\overline{n}) \in \Gamma$. If it is, $n \in D$, and if it isn't, $n \notin D$. So $D$ is likewise decidable.

Since $\Gamma$ represents all decidable properties, it represents $D$. And the formulas which represent $D$ in $\Gamma$ are all among $\varphi_0(x)$, $\varphi_1(x)$, .... So let $d$ be a number such that $\varphi_d(x)$ represents $D$ in $\Gamma$. If $d \notin D$, then, since $\varphi_d(x)$ represents $D$, $\Gamma \vdash \neg\varphi_d(\overline{d})$. But that means that $d$ meets the defining condition of $D$, and so $d \in D$. This contradicts $d \notin D$. So by indirect proof, $d \in D$.

Since $d \in D$, by the definition of $D$, $\Gamma \vdash \neg\varphi_d(\overline{d})$. On the other hand, since $\varphi_d(x)$ represents $D$ in $\Gamma$, $\Gamma \vdash \varphi_d(\overline{d})$. Hence, $\Gamma$ is inconsistent. □

The preceding theorem shows that no theory that represents all decidable relations can be decidable. We will show that **Q** does represent all decidable relations; this means that all theories that include **Q**, such as **PA** and **TA**, also do, and hence also are not decidable. explanation

We can also use this result to obtain a weak version of the first incompleteness theorem. Any theory that is axiomatizable and complete is decidable. Consistent theories that are axiomatizable and represent all decidable properties then cannot be complete.

**Theorem 1.16.** *If $\Gamma$ is axiomatizable and complete it is decidable.*

*Proof.* Any inconsistent theory is decidable, since inconsistent theories contain all sentences, so the answer to the question "is $\varphi \in \Gamma$" is always "yes," i.e., can be decided.

So suppose $\Gamma$ is consistent, and furthermore is axiomatizable, and complete. Since $\Gamma$ is axiomatizable, it is computably enumerable. For we can enumerate all the correct derivations from the axioms of $\Gamma$ by a computable function. From a correct derivation we can compute the sentence it derives, and so together there is a computable function that enumerates all theorems of $\Gamma$. A sentence is a theorem of $\Gamma$ iff $\neg\varphi$ is not a theorem, since $\Gamma$ is consistent and complete. We can therefore decide if $\varphi \in \Gamma$ as follows. Enumerate all theorems of $\Gamma$.

When $\varphi$ appears on this list, we know that $\Gamma \vdash \varphi$. When $\neg\varphi$ appears on this list, we know that $\Gamma \nvdash \varphi$. Since $\Gamma$ is complete, one of these cases eventually obtains, so the procedure eventually produces and answer. □

**Corollary 1.17.** *If $\Gamma$ is consistent, axiomatizable, and represents every decidable property, it is not complete.* inc:int:dec: cor:incompleteness

*Proof.* If $\Gamma$ were complete, it would be decidable by the previous theorem (since it is axiomatizable and consistent). But since $\Gamma$ represents every decidable property, it is not decidable, by the first theorem. □

**Problem 1.1.** Show that $\mathbf{TA} = \{\varphi : \mathfrak{N} \vDash \varphi\}$ is not axiomatizable. You may assume that **TA** represents all decidable properties.

Once we have established that, e.g., **Q**, represents all decidable properties, the corollary tells us that **Q** must be incomplete. However, its proof does not provide an example of an independent sentence; it merely shows that such a sentence must exist. For this, we have to arithmetize syntax and follow Gödel's original proof idea. And of course, we still have to show the first claim, namely that **Q** does, in fact, represent all decidable properties.

It should be noted that not every *interesting* theory is incomplete or undecidable. There are many theories that are sufficiently strong to describe interesting mathematical facts that do not satisify the conditions of Gödel's result. For instance, $\mathbf{Pres} = \{\varphi \in \mathcal{L}_{A^+} : \mathbb{N} \vDash \varphi\}$, the set of sentences of the language of arithmetic without $\times$ true in the standard model, is both complete and decidable. This theory is called Presburger arithmetic, and proves all the truths about natural numbers that can be formulated just with $0$, $'$, and $+$.

# Chapter 2

# Arithmetization of Syntax

Note that arithmetization for signed tableaux is not yet available.

## 2.1 Introduction

inc:art:int:sec

In order to connect computability and logic, we need a way to talk about the objects of logic (symbols, terms, formulas, derivations), operations on them, and their properties and relations, in a way amenable to computational treatment. We can do this directly, by considering computable functions and relations on symbols, sequences of symbols, and other objects built from them. Since the objects of logical syntax are all finite and built from an enumerable sets of symbols, this is possible for some models of computation. But other models of computation—such as the recursive functions—-are restricted to numbers, their relations and functions. Moreover, ultimately we also want to be able to deal with syntax within certain theories, specifically, in theories formulated in the language of arithmetic. In these cases it is necessary to *arithmetize* syntax, i.e., to represent syntactic objects, operations on them, and their relations, as numbers, arithmetical functions, and arithmetical relations, respectively. The idea, which goes back to Leibniz, is to assign numbers to syntactic objects.

It is relatively straightforward to assign numbers to symbols as their "codes." Some symbols pose a bit of a challenge, since, e.g., there are infinitely many variables, and even infinitely many function symbols of each arity $n$. But of course it's possible to assign numbers to symbols systematically in such a way that, say, $v_2$ and $v_3$ are assigned different codes. Sequences of symbols (such as terms and formulas) are a bigger challenge. But if can deal with sequences of numbers purely arithmetically (e.g., by the powers-of-primes coding of sequences), we can extend the coding of individual symbols to coding of sequences of symbols, and then further to sequences or other arrangements of formulas, such as derivations. This extended coding is called "Gödel numbering." Every term, formula, and derivation is assigned a Gödel number.

By coding sequences of symbols as sequences of their codes, and by chosing a system of coding sequences that can be dealt with using computable functions, we can then also deal with Gödel numbers using computable functions. In practice, all the relevant functions will be primitive recursive. For instance, computing the length of a sequence and computing the $i$-th element of a sequence from the code of the sequence are both primitive recursive. If the number coding the sequence is, e.g., the Gödel number of a formula $\varphi$, we immediately see that the length of a formula and the (code of the) $i$-th symbol in a formula can also be computed from the Gödel number of $\varphi$. It is a bit harder to prove that, e.g., the property of being the Gödel number of a correctly formed term, of being the Gödel number of a coret derivation is primitive recursive. It is nevertheless possible, because the sequences of interest (terms, formulas, derivations) are inductively defined.

As an example, consider the operation of substitution. If $\varphi$ is a formula, $x$ a variable, and $t$ a term, then $\varphi[t/x]$ is the result of replacing every free occurrence of $x$ in $\varphi$ by $t$. Now suppose we have assigned Gödel numbers to $\varphi$, $x$, $t$—say, $k$, $l$, and $m$, respectively. The same scheme assigns a Gödel number to $\varphi[t/x]$, say, $n$. This mapping—of $k$, $l$, and $m$ to $n$—is the arithmetical analog of the substitution operation. When the substitution operation maps $\varphi$, $x$, $t$ to $\varphi[t/x]$, the arithmetized substitution functions maps the Gödel numbers $k$, $l$, $m$ to the Gödel number $n$. We will see that this function is primitive recursive.

Arithmetization of syntax is not just of abstract interest, although it was originally a non-trivial insight that languages like the language of arithmetic, which do not come with mechanisms for "talking about" languages can, after all, formalize complex properties of expressions. It is then just a small step to ask what a theory in this language, such as Peano arithmetic, can *prove* about its own language (including, e.g., whether sentences are provable or true). This leads us to the famous limitative theorems of Gödel (about unprovability) and Tarski (the undefinability of truth). But the trick of arithmetizing syntax is also important in order to prove some important results in computability theory, e.g., about the computational prower of theories or the relationship between different models of computability. The arithmetization of syntax serves as a model for arithmetizing other objects and properties. For instance, it is similarly possible to arithmetize configurations and computations (say, of Turing machines). This makes it possible to simulate computations in one model (e.g., Turing machines) in another (e.g., recursive functions).

## 2.2 Coding Symbols

inc:art:cod:sec

The basic language $\mathcal{L}$ of first order logic makes use of the symbols

$$\bot \quad \neg \quad \vee \quad \wedge \quad \to \quad \forall \quad \exists \quad = \quad ( \quad ) \quad ,$$

together with enumerable sets of variables and constant symbols, and enumerable sets of function symbols and predicate symbols of arbitrary arity. We can assign *codes* to each of these symbols in such a way that every symbol is as-

signed a unique number as its code, and no two different symbols are assigned the same number. We know that this is possible since the set of all symbols is enumerable and so there is a bijection between it and the set of natural numbers. But we want to make sure that we can recover the symbol (as well as some information about it, e.g., the arity of a function symbol) from its code in a computable way. There are many possible ways of doing this, of course. Here is one such way, which uses primitive recursive functions. (Recall that $\langle n_0, \ldots, n_k \rangle$ is the number coding the sequence of numbers $n_0, \ldots, n_k$.)

**Definition 2.1.** If $s$ is a symbol of $\mathcal{L}$, let the *symbol code* $\mathrm{c}_s$ be defined as follows:

1. If $s$ is among the logical symbols, $\mathrm{c}_s$ is given by the following table:

| $\bot$ | $\neg$ | $\vee$ | $\wedge$ | $\to$ | $\forall$ |
|---|---|---|---|---|---|
| $\langle 0,0 \rangle$ | $\langle 0,1 \rangle$ | $\langle 0,2 \rangle$ | $\langle 0,3 \rangle$ | $\langle 0,4 \rangle$ | $\langle 0,5 \rangle$ |
| $\exists$ | $=$ | $($ | $)$ | $,$ | |
| $\langle 0,6 \rangle$ | $\langle 0,7 \rangle$ | $\langle 0,8 \rangle$ | $\langle 0,9 \rangle$ | $\langle 0,10 \rangle$ | |

2. If $s$ is the $i$-th variable $v_i$, then $\mathrm{c}_s = \langle 1, i \rangle$.
3. If $s$ is the $i$-th constant symbol $c^n_i$, then $\mathrm{c}_s = \langle 2, i \rangle$.
4. If $s$ is the $i$-th $n$-ary function symbol $f^n_i$, then $\mathrm{c}_s = \langle 3, n, i \rangle$.
5. If $s$ is the $i$-th $n$-ary predicate symbol $P^n_i$, then $\mathrm{c}_s = \langle 4, n, i \rangle$.

**Proposition 2.2.** *The following relations are primitive recursive:*

1. $\mathrm{Fn}(x, n)$ *iff* $x$ *is the code of* $f^n_i$ *for some* $i$*, i.e.,* $x$ *is the code of an* $n$*-ary function symbol.*
2. $\mathrm{Pred}(x, n)$ *iff* $x$ *is the code of* $P^n_i$ *for some* $i$ *or* $x$ *is the code of* $=$ *and* $n = 2$*, i.e.,* $x$ *is the code of an* $n$*-ary predicate symbol.*

**Definition 2.3.** If $s_0, \ldots, s_{n-1}$ is a sequence of symbols, its *Gödel number* is $\langle \mathrm{c}_{s_0}, \ldots, \mathrm{c}_{s_{n-1}} \rangle$.

Note that *codes* and *Gödel numbers* are different things. For instance, the variable $v_5$ has a code $\mathrm{c}_{v_5} = \langle 1, 5 \rangle = 2^2 \cdot 3^6$. But the variable $v_5$ considered as a term is also a sequence of symbols (of length 1). The *Gödel number* $\ulcorner v_5 \urcorner$ of the *term* $v_5$ is $\langle \mathrm{c}_{v_5} \rangle = 2^{\mathrm{c}_{v_5}+1} = 2^{2^2 \cdot 3^6 + 1}$. explanation

**Example 2.4.** Recall that if $k_0, \ldots, k_{n-1}$ is a sequence of numbers, then the code of the sequence $\langle k_0, \ldots, k_{n-1} \rangle$ in the power-of-primes coding is

$$2^{k_0+1} \cdot 3^{k_1+1} \cdot \ldots \cdot p_{n-1}^{k_{n-1}},$$

where $p_i$ is the $i$-th prime (starting with $p_0 = 2$). So for instance, the formula $v_0 = \mathsf{0}$, or, more explicitly, $=(v_0, c_0)$, has the Gödel number

$$\langle \mathrm{c}_=, \mathrm{c}_(, \mathrm{c}_{v_0}, \mathrm{c}_,, \mathrm{c}_{c_0}, \mathrm{c}_) \rangle.$$

Here, $c_=$ is $\langle 0, 7\rangle = 2^{0+1} \cdot 3^{7=1}$, $c_{v_0}$ is $\langle 1, 0\rangle = 2^{1+1} \cdot 3^{0+1}$, etc. So $\ulcorner = (v_0, c_0)\urcorner$ is

$$2^{c_=+1} \cdot 3^{c_(+1} \cdot 5^{c_{v_0}+1} \cdot 7^{c_,+1} \cdot 11^{c_{c_0}+1} \cdot 13^{c_)+1} =$$
$$2^{2^1\cdot 3^8+1} \cdot 3^{2^1\cdot 3^9+1} \cdot 5^{2^2\cdot 3^1+1} \cdot 7^{2^1\cdot 3^{11}+1} \cdot 11^{2^3\cdot 3^1+1} \cdot 13^{2^1\cdot 3^{10}+1} =$$
$$2^{13\,123} \cdot 3^{39\,367} \cdot 5^{13} \cdot 7^{354\,295} \cdot 11^{25} \cdot 13^{118\,099}.$$

## 2.3 Coding Terms

explanation

A term is simply a certain kind of sequence of symbols: it is built up inductively from constants and variables according to the formation rules for terms. Since sequences of symbols can be coded as numbers—using a coding scheme for the symbols plus a way to code sequences of numbers—assigning Gödel numbers to terms is not difficult. The challenge is rather to show that the property a number has if it is the Gödel number of a correctly formed term is computable, or in fact primitive recursive.

inc:art:trm:sec

Variables and constant symbols are the simplest terms, and testing whether $x$ is the Gödel number of such a term is easy: $\mathrm{Var}(x)$ holds if $x$ is $\ulcorner v_i \urcorner$ for some $i$. In other words, $x$ is a sequence of length 1 and its single element $(x)_0$ is the code of some variable $v_i$, i.e., $x$ is $\langle\langle 1, i\rangle\rangle$ for some $i$. Similarly, $\mathrm{Const}(x)$ holds if $x$ is $\ulcorner c_i \urcorner$ for some $i$. Both of these relations are primitive recursive, since if such an $i$ exists, it must be $< x$:

$$\mathrm{Var}(x) \Leftrightarrow (\exists i < x)\ x = \langle\langle 1, i\rangle\rangle$$
$$\mathrm{Const}(x) \Leftrightarrow (\exists i < x)\ x = \langle\langle 2, i\rangle\rangle$$

**Proposition 2.5.** *The relations* $\mathrm{Term}(x)$ *and* $\mathrm{ClTerm}(x)$ *which hold iff* $x$ *is the Gödel number of a term or a closed term, respectively, are primitive recursive.*

inc:art:trm:prop:term-primrec

*Proof.* A sequence of symbols $s$ is a term iff there is a sequence $s_0, \ldots, s_{k-1} = s$ of terms which records how the term $s$ was formed from constant symbols and variables according to the formation rules for terms. To express that such a putative formation sequence follows the formation rules it has to be the case that, for each $i < k$, either

1. $s_i$ is a variable $v_j$, or
2. $s_i$ is a constant symbol $c_j$, or
3. $s_i$ is built from $n$ terms $t_1, \ldots, t_n$ occurring prior to place $i$ using an $n$-place function symbol $f^n_j$.

To show that the corresponding relation on Gödel numbers is primitive recursive, we have to express this condition primitive recursively, i.e., using primitive recursive functions, relations, and bounded quantification.

Suppose $y$ is the number that codes the sequence $s_0, \ldots, s_{k-1}$, i.e., $y = \langle {}^{\ulcorner}s_0{}^{\urcorner}, \ldots, {}^{\ulcorner}s_k{}^{\urcorner}\rangle$. It codes a formation sequence for the term with Gödel number $x$ iff for all $i < k$:

1. $\mathrm{Var}((y)_i)$, or
2. $\mathrm{Const}((y)_i)$, or
3. there is an $n$ and a number $z = \langle z_1, \ldots, z_n\rangle$ such that each $z_l$ is equal to some $(y)_{i'}$ for $i' < i$ and

$$(y)_i = {}^{\ulcorner}f_j^n{}^{\urcorner}(^{\frown} \mathrm{flatten}(z) {}^{\frown}{}^{\ulcorner})^{\urcorner},$$

and moreover $(y)_{k-1} = x$. (The function $\mathrm{flatten}(z)$ turns the sequence $\langle {}^{\ulcorner}t_1{}^{\urcorner}, \ldots, {}^{\ulcorner}t_n{}^{\urcorner}\rangle$ into ${}^{\ulcorner}t_1, \ldots, t_n{}^{\urcorner}$ and is primitive recursive.)

The indices $j$, $n$, the Gödel numbers $z_l$ of the terms $t_l$, and the code $z$ of the sequence $\langle z_1, \ldots, z_n\rangle$, in (3) are all less than $y$. We can replace $k$ above with $\mathrm{len}(y)$. Hence we can express "$y$ is the code of a formation sequence of the term with Gödel number $x$" in a way that shows that this relation is primitive recursive.

We now just have to convince ourselves that there is a primitive recursive bound on $y$. But if $x$ is the Gödel number of a term, it must have a formation sequence with at most $\mathrm{len}(x)$ terms (since every term in the formation sequence of $s$ must start at some place in $s$, and no two subterms can start at the same place). The Gödel number of each subterm of $s$ is of course $\leq x$. Hence, there always is a formation sequence with code $\leq x^{\mathrm{len}(x)}$.

For ClTerm, simply leave out the clause for variables. □

**Problem 2.1.** Show that the function $\mathrm{flatten}(z)$, which turns the sequence $\langle {}^{\ulcorner}t_1{}^{\urcorner}, \ldots, {}^{\ulcorner}t_n{}^{\urcorner}\rangle$ into ${}^{\ulcorner}t_1, \ldots, t_n{}^{\urcorner}$, is primitive recursive.

inc:art:trm: prop:num-primrec

**Proposition 2.6.** *The function* $\mathrm{num}(n) = {}^{\ulcorner}\overline{n}{}^{\urcorner}$ *is primitive recursive.*

*Proof.* We define $\mathrm{num}(n)$ by primitive recursion:

$$\begin{aligned} \mathrm{num}(0) &= {}^{\ulcorner}\mathsf{0}{}^{\urcorner} \\ \mathrm{num}(n+1) &= {}^{\ulcorner}\mathsf{\prime}({}^{\urcorner} {}^{\frown} \mathrm{num}(n) {}^{\frown} {}^{\ulcorner}){}^{\urcorner}. \end{aligned}$$

□

## 2.4 Coding Formulas

inc:art:frm: sec

**Proposition 2.7.** *The relation* $\mathrm{Atom}(x)$ *which holds iff* $x$ *is the Gödel number of an atomic formula, is primitive recursive.*

*Proof.* The number $x$ is the Gödel number of an atomic formula iff one of the following holds:

1. There are $n$, $j < x$, and $z < x$ such that for each $i < n$, $\mathrm{Term}((z)_i)$ and $x =$
$$\ulcorner P_j^n(\# \frown \mathrm{flatten}(z) \frown \urcorner)^\#.$$

2. There are $z_1, z_2 < x$ such that $\mathrm{Term}(z_1)$, $\mathrm{Term}(z_2)$, and $x =$
$$\ulcorner =(\# \frown z_1 \frown \urcorner, \# \frown z_2 \frown \urcorner)^\#.$$

3. $x = \ulcorner \bot \urcorner^\#$.

4. $x = \ulcorner \top \urcorner^\#$.

□

**Proposition 2.8.** *The relation* $\mathrm{Frm}(x)$ *which holds iff* $x$ *is the Gödel number of a formula is primitive recursive.* *inc:art:frm: prop:frm-primrec*

*Proof.* A sequence of symbols $s$ is a formula iff there is formation sequence $s_0$, $\ldots$, $s_{k-1} = s$ of formula which records how $s$ was formed from atomic formulas according to the formation rules. The code for each $s_i$ (and indeed of the code of the sequence $\langle s_0, \ldots, s_{k-1} \rangle$ is less than the code $x$ of $s$. □

**Problem 2.2.** Give a detailed proof of Proposition 2.8 along the lines of the first proof of Proposition 2.5

**Problem 2.3.** Give a detailed proof of Proposition 2.8 along the lines of the alternate proof of Proposition 2.5

**Proposition 2.9.** *The relation* $\mathrm{FreeOcc}(x, z, i)$*, which holds iff the* $i$*-th symbol of the formula with Gödel number* $x$ *is a free occurrence of the variable with Gödel number* $z$*, is primitive recursive.* *inc:art:frm: prop:freeocc-primrec*

*Proof.* Exercise. □

**Problem 2.4.** Prove Proposition 2.9. You may make use of the fact that any substring of a formula which is a formula is a sub-formula of it.

**Proposition 2.10.** *The property* $\mathrm{Sent}(x)$ *which holds iff* $x$ *is the Gödel number of a sentence is primitive recursive.*

*Proof.* A sentence is a formula without free occurrences of variables. So $\mathrm{Sent}(x)$ holds iff
$$(\forall i < \mathrm{len}(x))\ (\forall z < x)\ ((\exists j < z)\ z = \ulcorner v_j \urcorner^\# \to \neg \mathrm{FreeOcc}(x, z, i)).$$

□

## 2.5 Substitution

inc:art:sub:sec

Recall that substitution is the operation of replacing all free occurrences of a variable $u$ in a formula $\varphi$ by a term $t$, written $\varphi[t/u]$. This operation, when carried out on Gödel numbers of variables, formulas, and terms, is primitive recursive.

inc:art:sub:prop:subst-primrec

**Proposition 2.11.** *There is a primitive recursive function* $\mathrm{Subst}(x, y, z)$ *with the property that*

$$\mathrm{Subst}({}^{\ulcorner}\varphi^{\urcorner}, {}^{\ulcorner}t^{\urcorner}, {}^{\ulcorner}u^{\urcorner}) = {}^{\ulcorner}\varphi[t/u]^{\urcorner}$$

*Proof.* We can then define a function hSubst by primitive recursion as follows:

$$\begin{aligned}\mathrm{hSubst}(x, y, z, 0) &= \Lambda \\ \mathrm{hSubst}(x, y, z, i+1) &= \\ &\begin{cases} \mathrm{hSubst}(x, y, z, i) \frown y & \text{if } \mathrm{FreeOcc}(x, z, i) \\ \mathrm{append}(\mathrm{hSubst}(x, y, z, i), (x)_i) & \text{otherwise.} \end{cases}\end{aligned}$$

$\mathrm{Subst}(x, y, z)$ can now be defined as $\mathrm{hSubst}(x, y, z, \mathrm{len}(x))$. □

inc:art:sub:prop:free-for

**Proposition 2.12.** *The relation* $\mathrm{FreeFor}(x, y, z)$*, which holds iff the term with Gödel number* $y$ *is free for the variable with Gödel number* $z$ *in the formula with Gödel number* $x$*, is primitive recursive.*

*Proof.* Exercise. □

**Problem 2.5.** Prove Proposition 2.12

## 2.6 Derivations in LK

inc:art:plk:sec

The definition of Deriv is not actually primitive recursive and needs to be rewritten.

explanation

In order to arithmetize derivations, we must represent derivations as numbers. Since derivations are trees of sequents where each inference carries also a label, a recursive representation is the most obvious approach: we represent a derivation as a tuple, the components of which are the end-sequent, the label, and the representations of the sub-derivations leading to the premises of the last inference.

**Definition 2.13.** If $\Gamma$ is a finite sequence of sentences, $\Gamma = \langle \varphi_1, \ldots, \varphi_n \rangle$, then ${}^{\ulcorner}\Gamma^{\urcorner} = \langle {}^{\ulcorner}\varphi_1{}^{\urcorner}, \ldots, {}^{\ulcorner}\varphi_n{}^{\urcorner} \rangle$.

If $\Gamma \Rightarrow \Delta$ is a sequent, then a Gödel number of $\Gamma \Rightarrow \Delta$ is

$$^\#\Gamma \Rightarrow \Delta^\# = \langle {}^\#\Gamma^\#, {}^\#\Delta^\# \rangle$$

If $\pi$ is a derivation in **LK**, then ${}^\#\pi^\#$ is

1. $\langle 0, {}^\#\Gamma \Rightarrow \Delta^\# \rangle$ if $\pi$ consists only of the initial sequent $\Gamma \Rightarrow \Delta$.

2. $\langle 1, {}^\#\Gamma \Rightarrow \Delta^\#, k, {}^\#\pi'^\# \rangle$ if $\pi$ ends in an inference with one premise, $k$ is given by the following table according to which rule was used in the last inference, and $\pi'$ is the immediate subproof ending in the premise of the last inference.

| Rule: | WL | WR | CL | CR | XL | XR |
|---|---|---|---|---|---|---|
| $k$: | 1 | 2 | 3 | 4 | 5 | 6 |

| Rule: | $\neg$L | $\neg$R | $\wedge$L | $\vee$R | $\rightarrow$R |
|---|---|---|---|---|---|
| $k$: | 7 | 8 | 9 | 10 | 11 |

| Rule: | $\forall$L | $\forall$R | $\exists$L | $\exists$R | $=$ |
|---|---|---|---|---|---|
| $k$: | 12 | 13 | 14 | 15 | 16 |

3. $\langle 2, {}^\#\Gamma \Rightarrow \Delta^\#, k, {}^\#\pi'^\#, {}^\#\pi''^\# \rangle$ if $\pi$ ends in an inference with two premises, $k$ is given by the following table according to which rule was used in the last inference, and $\pi'$, $\pi''$ are the immediate subproof ending in the left and right premise of the last inference, respectively.

| Rule: | Cut | $\wedge$R | $\vee$L | $\rightarrow$L |
|---|---|---|---|---|
| $k$: | 1 | 2 | 3 | 4 |

explanation

Having settled on a representation of derivations, we must also show that we can manipulate such derivations primitive recursively, and express their essential properties and relations so. Some operations are simple: e.g., given a Gödel number $d$ of a derivation, $(s)_1$ gives us the Gödel number of its end-sequent. Some are much harder. We'll at least sketch how to do this. The goal is to show that the relation "$\pi$ is a derivation of $\varphi$ from $\Gamma$" is a primitive recursive relation of the Gödel numbers of $\pi$ and $\varphi$.

**Proposition 2.14.** *The following relations are primitive recursive:*

*inc:art:plk: prop:followsby*

1. *$\Gamma \Rightarrow \Delta$ is an initial sequent.*

2. *$\Gamma \Rightarrow \Delta$ follows from $\Gamma' \Rightarrow \Delta'$ (and $\Gamma'' \Rightarrow \Delta''$) by a rule of* **LK**.

3. *$\pi$ is a correct* **LK***-derivation.*

*Proof.* We have to show that the corresponding relations between Gödel numbers of formulas, sequences of Gödel numbers of formulas (which code sequences of formulas), and Gödel numbers of sequents, are primitive recursive.

1. $\Gamma \Rightarrow \Delta$ is an initial sequent if either there is a sentence $\varphi$ such that $\Gamma \Rightarrow \Delta$ is $\varphi \Rightarrow \varphi$, or there is a term $t$ such that $\Gamma \Rightarrow \Delta$ is $\emptyset \Rightarrow t = t$. In terms of Gödel numbers, $\mathrm{InitSeq}(s)$ holds iff

$$
\begin{aligned}
&(\exists x < s)\ (\mathrm{Sent}(x) \wedge s = \langle \langle x \rangle, \langle x \rangle \rangle) \vee \\
&(\exists t < s)\ (\mathrm{Term}(t) \wedge s = \langle 0, \langle {}^{\#}{=}({}^{\#} \frown t \frown {}^{\#}, {}^{\#} \frown t \frown {}^{\#}){}^{\#} \rangle \rangle).
\end{aligned}
$$

2. Here we have to show that for each rule of inference $R$ the relation $\mathrm{FollowsBy}_R(s, s')$ which holds if $s$ and $s'$ are the Gödel numbers of conclusion and premise of a correct application of $R$ is primitive recursive. If $R$ has two premises, $\mathrm{FollowsBy}_R$ of course has three arguments.

   For instance, $\Gamma \Rightarrow \Delta$ follows correctly from $\Gamma' \Rightarrow \Delta'$ by $\exists\mathrm{R}$ iff $\Gamma = \Gamma'$ and there is a sequence of formulas $\Delta''$, a formula $\varphi$, a variable $x$ and a closed term $t$ such that $\Delta' = \Delta'', \varphi[t/x]$ and $\Delta = \Delta'', \exists x\, \varphi$. We just have to translate this into Gödel numbers. If $s = {}^{\#}\Gamma \Rightarrow \Delta^{\#}$ then $(s)_0 = {}^{\#}\Gamma^{\#}$ and $(s)_1 = {}^{\#}\Delta^{\#}$. So, $\mathrm{FollowsBy}_{\exists\mathrm{R}}(s, s')$ holds iff

$$
\begin{aligned}
&(s)_0 = (s')_0 \wedge \\
&(\exists d < s)\ (\exists f < s)\ (\exists x < s)\ (\exists t < s')\ (\mathrm{Frm}(f) \wedge \mathrm{Var}(y) \wedge \mathrm{Term}(t) \wedge \\
&\qquad (s')_1 = d \frown \langle \mathrm{Subst}(f, t, x) \rangle \wedge \\
&\qquad (s)_1 = d \frown \langle \#(\exists) \frown y \frown f \rangle
\end{aligned}
$$

   The individual lines express, respectively, "$\Gamma = \Gamma$," "there is a sequence ($\Delta''$) with Gödel number $d$, a formula ($\varphi$) with Gödel number $f$, a variable with Gödel number $x$, and a term with Gödel number $t$," "$\Delta' = \Delta'', \varphi[t/x]$," and "$\Delta = \Delta'', \exists x\, \varphi$". (Remember that ${}^{\#}\Delta^{\#}$ is the number of a sequence of Gödel numbers of formulas in $\Delta$.)

3. We first define a helper relation $\mathrm{hDeriv}(s, n)$ which holds if $s$ codes a correct derivation to at least $n$ inferences up from the end sequent. If $n = 0$ we let the relation be satisfied by default. Otherwise, $\mathrm{hDeriv}(s, n+1)$ iff either $s$ consists just of an initial sequent, or it ends in a correct inference

and the codes of the immediate subderivations satisfy $\mathrm{hDeriv}(s, n)$.

$$
\begin{aligned}
\mathrm{hDeriv}(s, 0) &\Leftrightarrow \mathrm{true} \\
\mathrm{hDeriv}(s, n+1) &\Leftrightarrow \\
&\quad ((s)_0 = 0 \wedge \mathrm{InitialSeq}((s)_1)) \vee \\
&\quad ((s)_0 = 1 \wedge \\
&\qquad ((s)_2 = 1 \wedge \mathrm{FollowsBy}_{\mathrm{CL}}((s)_1, ((s)_3)_1)) \vee \\
&\qquad\quad \vdots \\
&\qquad ((s)_2 = 16 \wedge \mathrm{FollowsBy}_{=}((s)_1, ((s)_3)_1)) \wedge \\
&\qquad \mathrm{hDeriv}((s)_3, n)) \vee \\
&\quad ((s)_0 = 2 \wedge \\
&\qquad ((s)_2 = 1 \wedge \mathrm{FollowsBy}_{\mathrm{Cut}}((s)_1, ((s)_3)_1), ((s)_4)_1)) \vee \\
&\qquad\quad \vdots \\
&\qquad ((s)_2 = 4 \wedge \mathrm{FollowsBy}_{\to\mathrm{L}}((s)_1, ((s)_3)_1), ((s)_4)_1)) \wedge \\
&\qquad \mathrm{hDeriv}((s)_3, n) \wedge \mathrm{hDeriv}((s)_4, n))
\end{aligned}
$$

This is a primitive recursive definition. If the number $n$ is large enough, e.g., larger than the maximum number of inferences between an initial sequent and the end sequent in $s$, it holds of $s$ iff $s$ is the Gödel number of a correct derivation. The number $s$ itself is larger than that maximum number of inferences. So we can now define $\mathrm{Deriv}(s)$ by $\mathrm{hDeriv}(s, s)$.

□

**Problem 2.6.** Define the following relations as in Proposition 2.14:

1. $\mathrm{FollowsBy}_{\wedge\mathrm{R}}(s, s', s'')$,
2. $\mathrm{FollowsBy}_{=}(s, s')$,
3. $\mathrm{FollowsBy}_{\forall\mathrm{R}}(s, s')$.

**Proposition 2.15.** *Suppose $\Gamma$ is a primitive recursive set of sentences. Then the relation $\mathrm{Prf}_\Gamma(x, y)$ expressing "$x$ is the code of a derivation $\pi$ of $\Gamma_0 \Rightarrow \varphi$ for some finite $\Gamma_0 \subseteq \Gamma$ and $x$ is the Gödel number of $\varphi$" is primitive recursive.*

*Proof.* Suppose "$y \in \Gamma$" is given by the primitive recursive predicate $R_\Gamma(y)$. We have to show that $\mathrm{Prf}_\Gamma(x, y)$ which holds iff $y$ is the Gödel number of a sentence $\varphi$ and $x$ is the code of an **LK**-derivation with end sequent $\Gamma_0 \Rightarrow \varphi$ is primitive recursive.

By the previous proposition, the property $\mathrm{Deriv}(x)$ which holds iff $x$ is the code of a correct derivation $\pi$ in **LK** is primitive recursive. If $x$ is such a code, then $(x)_1$ is the code of the end sequent of $\pi$, and so $((x)_1)_0$ is the code of the left side of the end sequent and $((x)_1)_1$ the right side. So we can express "the

right side of the end sequent of $\pi$ is $\varphi$" as $\mathrm{len}(((x)_1)_1) = 1 \wedge (((x)_1)_1)_0 = x$. The left side of the end sequent of $\pi$ is of course automatically finite, we just have to express that every sentence in it is in $\Gamma$. Thus we can define $\mathrm{Prf}_\Gamma(x, y)$ by

$$\begin{aligned}\mathrm{Prf}_\Gamma(x, y) \Leftrightarrow \mathrm{Sent}(y) \wedge \mathrm{Deriv}(x) \wedge \\ (\forall i < \mathrm{len}(((x)_1)_0))\ R_\Gamma((((x)_1)_0)_i) \wedge \\ \mathrm{len}(((x)_1)_1) = 1 \wedge (((x)_1)_1)_0 = x\end{aligned}$$

□

## 2.7 Derivations in Natural Deduction

inc:art:pnd: sec

In order to arithmetize derivations, we must represent derivations as numbers. Since derivations are trees of formulas where each inference carries one or two labels, a recursive representation is the most obvious approach: we represent a derivation as a tuple, the components of which are the number of immediate sub-derivations leading to the premises of the last inference, the representations of these sub-derivations, and the end-formula, the discharge label of the last inference, and a number indicating the type of the last inference.

explanation

**Definition 2.16.** If $\delta$ is a derivation in natural deduction, then ${}^\ulcorner\delta^\urcorner$ is defined inductively as follows:

1. If $\delta$ consists only of the assumption $\varphi$, then ${}^\ulcorner\delta^\urcorner$ is $\langle 0, {}^\ulcorner\varphi^\urcorner, n\rangle$. The number $n$ is 0 if it is an undischarged assumption, and the numerical label otherwise.

2. If $\delta$ ends in an inference with one, two, or three premises, then ${}^\ulcorner\delta^\urcorner$ is $\langle 1, {}^\ulcorner\delta_1{}^\urcorner, {}^\ulcorner\varphi^\urcorner, n, k\rangle$, $\langle 2, {}^\ulcorner\delta_1{}^\urcorner, {}^\ulcorner\delta_2{}^\urcorner, {}^\ulcorner\varphi^\urcorner, n, k\rangle$, or $\langle 3, {}^\ulcorner\delta_1{}^\urcorner, {}^\ulcorner\delta_2{}^\urcorner, {}^\ulcorner\delta_3{}^\urcorner, {}^\ulcorner\varphi^\urcorner, n, k\rangle$, respectively. Here $\delta_1$, $\delta_2$, $\delta_3$ are the sub-derivations ending in the premise(s) of the last inference in $\delta$, $\varphi$ is the conclusion of the last inference in $\delta$, $n$ is the discharge label of the last inference (0 if the inference does not discharge any assumptions), and $k$ is given by the following table according to which rule was used in the last inference.

| Rule: | ∧Intro | ∧Elim | ∨Intro | ∨Elim |
|---|---|---|---|---|
| $k$: | 1 | 2 | 3 | 4 |
| Rule: | →Intro | →Elim | ¬Intro | ¬Elim |
| $k$: | 5 | 6 | 7 | 8 |
| Rule: | $\bot_I$ | $\bot_C$ | ∀Intro | ∀Elim |
| $k$: | 9 | 10 | 11 | 12 |
| Rule: | ∃Intro | ∃Elim | =Intro | =Elim |
| $k$: | 13 | 14 | 15 | 16 |

**Example 2.17.** Consider the very simple derivation

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

The Gödel number of the assumption would be $d_0 = \langle 0, {}^{\ulcorner}\varphi \wedge \psi{}^{\urcorner}, 1\rangle$. The Gödel number of the derivation ending in the conclusion of $\wedge$Elim would be $d_1 = \langle 1, d_0, {}^{\ulcorner}\varphi{}^{\urcorner}, 0, 2\rangle$ (1 since $\wedge$Elim has one premise, the Gödel number of conclusion $\varphi$, 0 because no assumption is discharged, and 2 is the number coding $\wedge$Elim). The Gödel number of the entire derivation then is $\langle 1, d_1, {}^{\ulcorner}((\varphi \wedge \psi) \to \varphi){}^{\urcorner}, 1, 5\rangle$, i.e.,

$$\langle 1, \langle 1, \langle 0, {}^{\ulcorner}(\varphi \wedge \psi){}^{\urcorner}, 1\rangle, {}^{\ulcorner}\varphi{}^{\urcorner}, 0, 2\rangle, {}^{\ulcorner}((\varphi \wedge \psi) \to \varphi){}^{\urcorner}, 1, 5\rangle.$$

explanation

Having settled on a representation of derivations, we must also show that we can manipulate Gödel numbers of such derivations primitive recursively, and express their essential properties and relations. Some operations are simple: e.g., given a Gödel number $d$ of a derivation, $\mathrm{EndFmla}(d) = (d)_{(d)_0+1}$ gives us the Gödel number of its end-formula, $\mathrm{DischargeLabel}(d) = (d)_{(d)_0+2}$ gives us the discharge label and $\mathrm{LastRule}(d) = (d)_{(d)_0+3}$ the number indicating the type of the last inference. Some are much harder. We'll at least sketch how to do this. The goal is to show that the relation "$\delta$ is a derivation of $\varphi$ from $\Gamma$" is a primitive recursive relation of the Gödel numbers of $\delta$ and $\varphi$.

**Proposition 2.18.** *The following relations are primitive recursive:*

*inc:art:pnd: prop:followsby*

1. *$\varphi$ occurs as an assumption in $\delta$ with label $n$.*
2. *All assumptions in $\delta$ with label $n$ are of the form $\varphi$ (i.e., we can discharge the assumption $\varphi$ using label $n$ in $\delta$).*

*Proof.* We have to show that the corresponding relations between Gödel numbers of formulas and Gödel numbers of derivations are primitive recursive.

1. We want to show that $\mathrm{Assum}(x, d, n)$, which holds if $x$ is the Gödel number of an assumption of the derivation with Gödel number $d$ labelled $n$, is primitive recursive. This is the case if the derivation with Gödel number $\langle 0, x, n\rangle$ is a sub-derivation of $d$. Note that the way we code derivations is a special case of the coding of trees introduced in **??**, so the primitive recursive function $\mathrm{SubtreeSeq}(d)$ gives a sequence of Gödel numbers of all sub-derivations of $d$ (of length a most $d$). So we can define

$$\mathrm{Assum}(x, d, n) \Leftrightarrow (\exists i < d)\ (\mathrm{SubtreeSeq}(d))_i = \langle 0, x, n\rangle.$$

2. We want to show that $\mathrm{Discharge}(x, d, n)$, which holds if all assumptions with label $n$ in the derivation with Gödel number $d$ all are the formula with Gödel number $x$. But this relation holds iff $(\forall y < d)\ (\mathrm{Assum}(y, d, n) \to y = x)$.

□

**Proposition 2.19.** *The relation* Correct$(d)$ *which holds if the last inference in the derivation $\delta$ with Gödel number $d$ is correct, is primitive recursive.*

*Proof.* Here we have to show that for each rule of inference $R$ the relation FollowsBy$_R(d)$ is primitive recursive, where FollowsBy$_R(d)$ holds iff $d$ is the Gödel number of derivation $\delta$, and the end-formula of $\delta$ follows by a correct application of $R$ from the immediate sub-derivations of $\delta$.

A simple case is that of the $\wedge$Intro rule. If $\delta$ ends in a correct $\wedge$Intro inference, it looks like this:

$$\dfrac{\begin{array}{cc}\vdots & \vdots \\ A & B\end{array}}{A \wedge B}\ \wedge\text{Intro}$$

Then the Gödel number $d$ of $\delta$ is $\langle 2, d_1, d_2, {}^{\ulcorner}(A \wedge B)^{\urcorner}, 0, k\rangle$ where EndFmla$(d_1) = {}^{\ulcorner}A^{\urcorner}$, EndFmla$(d_2) = {}^{\ulcorner}B^{\urcorner}$, $n = 0$, and $k = 1$. So we can define FollowsBy$_{\wedge\text{Intro}}(d)$ as

$$\begin{aligned}&(d)_0 = 2 \wedge \text{DischargeLabel}(d) = 0 \wedge \text{LastRule}(d) = 1 \wedge \\ &\text{EndFmla}(d) = {}^{\ulcorner}(^{\urcorner} \frown \text{EndFmla}((d)_1) \frown {}^{\ulcorner}\wedge^{\urcorner} \frown \text{EndFmla}((d)_2) \frown {}^{\ulcorner})^{\urcorner}.\end{aligned}$$

Another simple example if the $=$Intro rule. Here the premise is an empty derivation, i.e., $(d)_1 = 0$, and no discharge label, i.e., $n = 0$. However, $\varphi$ must be of the form $t = t$, for a closed term $t$. Here, a primitive recursive definition is

$$\begin{aligned}&(d)_0 = 1 \wedge (d)_1 = 0 \wedge \text{DischargeLabel}(d) = 0 \wedge \\ &\qquad (\exists t < d)\ (\text{ClTerm}(t) \wedge \text{EndFmla}(d) = {}^{\ulcorner}=(^{\urcorner} \frown t \frown {}^{\ulcorner},^{\urcorner} \frown t \frown {}^{\ulcorner})^{\urcorner})\end{aligned}$$

For a more complicated example, FollowsBy$_{\to\text{Intro}}(d)$ holds iff the end-formula of $\delta$ is of the form $(\varphi \to \psi)$, where the end-formula of $\delta_1$ is $\psi$, and any assumption in $\delta$ labelled $n$ is of the form $\varphi$. We can express this primitive recursively by

$$\begin{aligned}&(d)_0 = 1 \wedge \\ &\qquad (\exists a < d)\ (\text{Discharge}(a, (d)_1, \text{DischargeLabel}(d)) \wedge \\ &\qquad \text{EndFmla}(d) = ({}^{\ulcorner}(^{\urcorner} \frown a \frown {}^{\ulcorner}\to^{\urcorner} \frown \text{EndFmla}((d)_1) \frown {}^{\ulcorner})^{\urcorner}))\end{aligned}$$

(Think of $a$ as the Gödel number of $\varphi$).

For another example, consider $\exists$Intro. Here, the last inference in $\delta$ is correct iff there is a formula $\varphi$, a closed term $t$ and a variable $x$ such that $\varphi[t/x]$ is the

end-formula of the derivation $\delta_1$ and $\exists x\, \varphi$ is the conclusion of the last inference. So, $\text{FollowsBy}_{\exists\text{Intro}}(d)$ holds iff

$$\begin{aligned}(d)_0 = 1 \wedge \text{DischargeLabel}(d) = 0 \wedge \\ (\exists a < d)\ (\exists x < d)\ (\exists t < d)\ (\text{ClTerm}(t) \wedge \text{Var}(x) \wedge \\ \text{Subst}(a, t, x) = \text{EndFmla}((d)_1) \wedge \text{EndFmla}(d) = ({}^{\#}\exists^{\#} \frown x \frown a))\end{aligned}$$

We then define $\text{Correct}(d)$ as

$$\begin{aligned}\text{Sent}(\text{EndFmla}(d)) \wedge (\text{LastRule}(d) = 1 \wedge \text{FollowsBy}_{\wedge\text{Intro}}(d)) \vee \cdots \vee \\ (\text{LastRule}(d) = 16 \wedge \text{FollowsBy}_{=\text{Elim}}(d)) \vee \\ (\exists n < d)\ (\exists x < d)\ (d = \langle 0, x, n\rangle)\end{aligned}$$

The first line ensures that the end-formula of $d$ is a sentence. The last line covers the case where $d$ is just an assumption. □

**Problem 2.7.** Define the following relations as in Proposition 2.18:

1. $\text{FollowsBy}_{\to\text{Elim}}(d)$,
2. $\text{FollowsBy}_{=\text{Elim}}(d)$,
3. $\text{FollowsBy}_{\vee\text{Elim}}(d)$,
4. $\text{FollowsBy}_{\forall\text{Intro}}(d)$.

For the last one, you will have to also show that you can test primitive recursively if the the last inference of the derivation with Gödel number $d$ satisfies the eigenvariable condition, i.e., the eigenvariable $a$ of the $\forall$Intro inference occurs neither in the end-formula of $d$ nor in an open assumption of $d$. You may use the primitive recursive predicate OpenAssum from Proposition 2.21 for this.

**Proposition 2.20.** *The relation* $\text{Deriv}(d)$ *which holds if* $d$ *is the Gödel number of a correct derivation* $\delta$*, is primitive recursive.* *inc:art:pnd: prop:deriv*

*Proof.* A derivation $\delta$ is correct if every one of its inferences is a correct application of a rule, i.e., if every one of its sub-derivations ends in a correct inference. So, $\text{Deriv}(d)$ iff

$$(\forall i < \text{len}(\text{SubtreeSeq}(d)))\ \text{Correct}((\text{SubtreeSeq}(d))_i)$$

□

**Proposition 2.21.** *The relation* $\text{OpenAssum}(z, d)$ *that holds if* $z$ *is the Gödel number of an undischarged assumption* $\varphi$ *of the derivation* $\delta$ *with Gödel number* $d$*, is primitive recursive.* *inc:art:pnd: prop:openassum*

*Proof.* An occurrence of an assumption is discharged if it occurs with label $n$ in a sub-derivation of $\delta$ that ends in a rule with discharge label $n$. So $\varphi$ is an undischarged assumption of $\delta$ if at least one of its occurrences is not discharged in $\delta$. We must be careful: $\delta$ may contain both discharged and undischarged occurrences of $\varphi$.

Consider a sequence $\delta_0, \ldots, \delta_k$ where $\delta_0 = d$, $\delta_k$ is the assumption $[\varphi]^n$ (for some $n$), and $\delta_i$ is an immediate sub-derivation of $\delta_{i+1}$. If such a sequence exists in which no $\delta_i$ ends in an inference with discharge label $n$, then $\varphi$ is an undischarged assumption of $\delta$.

The primitive recursive function $\mathrm{SubtreeSeq}(d)$ provides us with a sequence of Gödel numbers of all sub-derivations of $\delta$. Any sequence of Gödel numbers of sub-derivations of $\delta$ is a subsequence of it. Being a subsequence of is a primitive recursive relation: $\mathrm{Subseq}(s, s')$ holds iff $(\forall i < \mathrm{len}(s))\ \exists j < \mathrm{len}(s')\, (s)_i = (s)_j$. Being an immediate sub-derivation is as well: $\mathrm{Subderiv}(d, d')$ iff $(\exists j < (d')_0)\ d = (d')_j$. So we can define $\mathrm{OpenAssum}(z, d)$ by

$$\begin{aligned}(\exists s < \mathrm{SubtreeSeq}(d))\ (\mathrm{Subseq}(s, \mathrm{SubtreeSeq}(d)) \wedge (s)_0 = d \wedge \\ (\exists n < d)\ ((s)_{\mathrm{len}(s) \dot{-} 1} = \langle 0, z, n\rangle \wedge \\ (\forall i < (\mathrm{len}(s) \dot{-} 1))\ (\mathrm{Subderiv}((s)_i, (s)_{i+1})] \wedge \\ \mathrm{DischargeLabel}((s)_{i+1}) \neq n))).\end{aligned}$$

□

**Proposition 2.22.** *Suppose $\Gamma$ is a primitive recursive set of sentences. Then the relation $\mathrm{Prf}_\Gamma(x, y)$ expressing "$x$ is the code of a derivation $\delta$ of $\varphi$ from undischarged assumptions in $\Gamma$ and $y$ is the Gödel number of $\varphi$" is primitive recursive.*

*Proof.* Suppose "$y \in \Gamma$" is given by the primitive recursive predicate $R_\Gamma(y)$. We have to show that $\mathrm{Prf}_\Gamma(x, y)$ which holds iff $y$ is the Gödel number of a sentence $\varphi$ and $x$ is the code of a natural deduction derivation with end formula $\varphi$ and all undischarged assumptions in $\Gamma$ is primitive recursive.

By Proposition 2.20, the property $\mathrm{Deriv}(x)$ which holds iff $x$ is the Gödel number of a correct derivation $\delta$ in natural deduction is primitive recursive. Thus we can define $\mathrm{Prf}_\Gamma(x, y)$ by

$$\begin{aligned}\mathrm{Deriv}(x) \wedge \mathrm{EndFmla}(x) = y \wedge \\ (\forall z < x)\ (\mathrm{OpenAssum}(z, x) \to R_\Gamma(z))\end{aligned}$$

□

## 2.8 Axiomatic Derivations

inc:art:pax:sec

In order to arithmetize axiomatic derivations, we must represent derivations as numbers. Since derivations are simply sequences of formulas, the obvious

explanation

approach is to code every derivation as the code of the sequence of codes of formulas in it.

**Definition 2.23.** If $\delta$ is an axiomatic derivation consisting of formulas $\varphi_1$, $\dots$, $\varphi_n$, then ${}^{\ulcorner}\delta^{\urcorner}$ is

$$\langle {}^{\ulcorner}\varphi_1{}^{\urcorner}, \dots, {}^{\ulcorner}\varphi_n{}^{\urcorner} \rangle.$$

**Example 2.24.** Consider the very simple derivation

1. $\psi \to (\psi \vee \varphi)$
2. $(\psi \to (\psi \vee \varphi)) \to (\varphi \to (\psi \to (\psi \vee \varphi)))$
3. $\varphi \to (\psi \to (\psi \vee \varphi))$

The Gödel number of this derivation would simply be

$$\langle {}^{\ulcorner}\psi \to (\psi \vee \varphi)^{\urcorner}, {}^{\ulcorner}(\psi \to (\psi \vee \varphi)) \to (\varphi \to (\psi \to (\psi \vee \varphi)))^{\urcorner}, {}^{\ulcorner}\varphi \to (\psi \to (\psi \vee \varphi))^{\urcorner} \rangle.$$

explanation

Having settled on a representation of derivations, we must also show that we can manipulate such derivations primitive recursively, and express their essential properties and relations so. Some operations are simple: e.g., given a Gödel number $d$ of a derivation, $(d)_{\mathrm{len}(d)-1}$ gives us the Gödel number of its end-formula. Some are much harder. We'll at least sketch how to do this. The goal is to show that the relation "$\delta$ is a derivation of $\varphi$ from $\Gamma$" is primitive recursive on the Gödel numbers of $\delta$ and $\varphi$.

**Proposition 2.25.** *The following relations are primitive recursive:*

inc:art:pax: prop:followsby

1. *$\varphi$ is an axiom.*
2. *The $i$th line in $\delta$ is justified by modus ponens*
3. *The $i$th line in $\delta$ is justified by* QR*.*
4. *$\delta$ is a correct derivation.*

*Proof.* We have to show that the corresponding relations between Gödel numbers of formulas and Gödel numbers of derivations are primitive recursive.

1. We have a given list of axiom schemas, and $\varphi$ is an axiom if it is of the form given by one of these schemas. Since the list of schemas is finite, it suffices to show that we can test primitive recursively, for each axiom schema, if $\varphi$ is of that form. For instance, consider the axiom schema

   $$\psi \to (\chi \to \psi).$$

   $\varphi$ is an instance of this axiom schema if there are formulas $\psi$ and $\chi$ such that we obtain $\varphi$ when we concatenate ( with $\psi$ with $\to$ with ( with $\chi$ with $\to$ with $\psi$ and with )). We can test the corresponding property of the Gödel number $n$ of $\varphi$, since concatenation of sequences is primitive recursive, and the Gödel numbers of $\psi$ and $C$ must be smaller than the

Gödel number of $\varphi$, since when the relation holds, both $\psi$ and $\chi$ are sub-formulas of $\varphi$. Hence, we can define

$$\begin{aligned}\mathrm{IsAx}_{\psi\to(\chi\to\psi)}(n) \Leftrightarrow (\exists b < n)\ (\exists c < n)\ (\mathrm{Sent}(b) \wedge \mathrm{Sent}(c) \wedge \\ n = {}^{\ulcorner}(^{\ulcorner} \frown b \frown {}^{\ulcorner}\to^{\ulcorner} \frown {}^{\ulcorner}(^{\ulcorner} \frown c \frown {}^{\ulcorner}\to^{\ulcorner} \frown b \frown {}^{\ulcorner}))^{\ulcorner}).\end{aligned}$$

If we have such a definition for each axiom schema, their disjunction defines the property $\mathrm{IsAx}(n)$, "$n$ is the Gödel number of an axiom."

2. The $i$th line in $\delta$ is justified by modus ponens iff there are lines $j$ and $k < i$ where the sentence on line $j$ is some formula $\varphi$, the sentence on line $k$ is $\varphi \to \psi$, and the sentence on line $i$ is $\psi$.

$$\begin{aligned}\mathrm{MP}(d, i) \Leftrightarrow (\exists j < i)\ (\exists k < i) \\ (d)_k = {}^{\ulcorner}(^{\ulcorner} \frown (d)_j \frown {}^{\ulcorner}\to^{\ulcorner} \frown (d)_i \frown {}^{\ulcorner})^{\ulcorner}\end{aligned}$$

Since bounded quantification, concatenation, and $=$ are primitive recursive, this defines a primitive recursive relation.

3. A line in $\delta$ is justified by QR if it is of the form $\psi \to \forall x\, \varphi(x)$, a preceding line is $\psi \to \varphi(c)$ for some constant symbol $c$, and $c$ does on occur in $\psi$. This is the case iff

   a) there is a sentence $\psi$ and
   b) a formula $\varphi(x)$ with a single variable $x$ free so that
   c) line $i$ contains $\psi \to \forall x\, \varphi(x)$
   d) some line $j < i$ contains $\psi \to \varphi[c/x]$ for a constant $c$
   e) which does not occur in $\psi$.

   All of these can be tested primitive recursively, since the Gödel numbers of $\psi$, $\varphi(x)$, and $x$ are less than the Gödel number of the formula on line $i$, and that of $a$ less than the Gödel number of the formula on line $j$:

$$\begin{aligned}\mathrm{QR}_1(d, i) \Leftrightarrow (\exists b < (d)_i)\ (\exists x < (d)_i)\ (\exists a < (d)_i)\ (\exists c < (d)_j)\ ( \\ \mathrm{Var}(x) \wedge \mathrm{Const}(c) \wedge \\ (d)_i = {}^{\ulcorner}(^{\ulcorner} \frown b \frown {}^{\ulcorner}\to^{\ulcorner} \frown {}^{\ulcorner}\forall^{\ulcorner} \frown x \frown a \frown {}^{\ulcorner})^{\ulcorner} \wedge \\ (d)_j = {}^{\ulcorner}(^{\ulcorner} \frown b \frown {}^{\ulcorner}\to^{\ulcorner} \frown \mathrm{Subst}(a, c, x) \frown {}^{\ulcorner})^{\ulcorner} \wedge \\ \mathrm{Sent}(b) \wedge \mathrm{Sent}(\mathrm{Subst}(a, c, x)) \wedge (\forall k < \mathrm{len}(b))\ (b)_k \neq (c)_0)\end{aligned}$$

   Here we assume that $c$ and $x$ are the Gödel numbers of the variable and constant considered as terms (i.e., not their symbol codes). We test that $x$ is the only free variable of $\varphi(x)$ by testing if $\varphi(x)[c/x]$ is a sentence, and ensure that $c$ does not occur in $\psi$ by requiring that every symbol of $\psi$ is different from $c$.

   We leave the other version of QR as an exercise.

4. $d$ is the Gödel number of a correct derivation iff every line in it is an axiom, or justified by modus ponens or QR. Hence:

$$\mathrm{Deriv}(d) \Leftrightarrow (\forall i < \mathrm{len}(d))\ (\mathrm{IsAx}((d)_i) \vee \mathrm{MP}(d, i) \vee \mathrm{QR}(d, i))$$

□

**Problem 2.8.** Define the following relations as in Proposition 2.25:

1. $\mathrm{IsAx}_{\varphi\to(\psi\to(\varphi\wedge\psi))}(n)$,
2. $\mathrm{IsAx}_{\forall x\,\varphi(x)\to\varphi(t)}(n)$,
3. $\mathrm{QR}_2(d, i)$ (for the other version of QR).

**Proposition 2.26.** *Suppose $\Gamma$ is a primitive recursive set of sentences. Then the relation $\mathrm{Prf}_\Gamma(x, y)$ expressing "$x$ is the code of a derivation $\delta$ of $\varphi$ from $\Gamma$ and $y$ is the Gödel number of $\varphi$" is primitive recursive.*

*Proof.* Suppose "$y \in \Gamma$" is given by the primitive recursive predicate $R_\Gamma(y)$. We have to show that the relation $\mathrm{Prf}_\Gamma(x, y)$ is primitive recursive, where $\mathrm{Prf}_\Gamma(x, y)$ holds iff $y$ is the Gödel number of a sentence $\varphi$ and $x$ is the code of a derivation of $\varphi$ from $\Gamma$.

By the previous proposition, the property $\mathrm{Deriv}(x)$ which holds iff $x$ is the code of a correct derivation $\delta$ is primitive recursive. However, that definition did not take into account the set $\Gamma$ as an additional way to justify lines in the derivation. Our primitive recursive test of whether a line is justified by QR also left out of consideration the requirement that the constant $c$ is not allowed to occur in $\Gamma$. It is possible to amend our definition so that it takes into account $\Gamma$ directly, but it is easier to use Deriv and the deduction theorem. $\Gamma \vdash \varphi$ iff there is some finite list of sentences $\psi_1, \dots, \psi_n \in \Gamma$ such that $\{\psi_1, \dots, \psi_n\} \vdash \varphi$. And by the deduction theorem, this is the case if $\vdash (\psi_1 \to (\psi_2 \to \cdots (\psi_n \to \varphi) \cdots))$. Whether a sentence with Gödel number $z$ is of this form can be tested primitive recursively. So, instead of considering $x$ as the Gödel number of a derivation of the sentence with Gödel number $y$ *from* $\Gamma$, we consider $x$ as the Gödel number of a derivation of a nested conditional of the above form from $\emptyset$.

First, if we have a sequence of sentences, we can primitive recursively form the conditional with all these sentences as antecedents and given sentence as consequent:

$$\begin{aligned}
\mathrm{hCond}(s, y, 0) &= y \\
\mathrm{hCond}(s, y, n+1) &= {}^{\frown}(^{\#} \frown (s)_n \frown {}^{\frown}\!\to^{\#} \frown \mathrm{Cond}(s, y, n) \frown {}^{\frown})^{\#} \\
\mathrm{Cond}(s, y) &= \mathrm{hCond}(s, y, \mathrm{len}(s))
\end{aligned}$$

So we can define $\mathrm{Prf}_\Gamma(x, y)$ by

$$\begin{aligned}\mathrm{Prf}_\Gamma(x, y) \Leftrightarrow (\exists s < \mathrm{sequenceBound}(x, x))\ ( & \\ & (x)_{\mathrm{len}(x)-1} = \mathrm{Cond}(s, y) \wedge \\ & (\forall i < \mathrm{len}(s))\ (s)_i \in \Gamma \wedge \\ & \mathrm{Deriv}(x)).\end{aligned}$$

The bound on $s$ is given by considering that each $(s)_i$ is the Gödel number of a subformula of the last line of the derivation, i.e., is less than $(x)_{\mathrm{len}(x)-1}$. The number of antecedents $\psi \in \Gamma$, i.e., the length of $s$, is less than the length of the last line of $x$. □

# Chapter 3

# Representability in Q

## 3.1 Introduction

inc:req:int:sec

The incompleteness theorems apply to theories in which basic facts about computable functions can be expressed and proved. We will describe a very minimal such theory called "**Q**" (or, sometimes, "Robinson's $Q$," after Raphael Robinson). We will say what it means for a function to be *representable* in **Q**, and then we will prove the following:

> A function is representable in **Q** if and only if it is computable.

For one thing, this provides us with another model of computability. But we will also use it to show that the set $\{\varphi : \mathbf{Q} \vdash \varphi\}$ is not decidable, by reducing the halting problem to it. By the time we are done, we will have proved much stronger things than this.

The language of **Q** is the language of arithmetic; **Q** consists of the following axioms (to be used in conjunction with the other axioms and rules of first-order logic with identity predicate):

$$\forall x\, \forall y\, (x' = y' \to x = y) \tag{$Q_1$}$$
$$\forall x\, \mathsf{o} \neq x' \tag{$Q_2$}$$
$$\forall x\, (x \neq \mathsf{o} \to \exists y\, x = y') \tag{$Q_3$}$$
$$\forall x\, (x + \mathsf{o}) = x \tag{$Q_4$}$$
$$\forall x\, \forall y\, (x + y') = (x + y)' \tag{$Q_5$}$$
$$\forall x\, (x \times \mathsf{o}) = \mathsf{o} \tag{$Q_6$}$$
$$\forall x\, \forall y\, (x \times y') = ((x \times y) + x) \tag{$Q_7$}$$
$$\forall x\, \forall y\, (x < y \leftrightarrow \exists z\, (z' + x) = y) \tag{$Q_8$}$$

For each natural number $n$, define the numeral $\overline{n}$ to be the term $0''^{\cdots'}$ where there are $n$ tick marks in all. So, $\overline{0}$ is the constant symbol $\mathsf{o}$ by itself, $\overline{1}$ is $\mathsf{o}'$, $\overline{2}$ is $\mathsf{o}''$, etc.

As a theory of arithmetic, **Q** is *extremely* weak; for example, you can't even prove very simple facts like $\forall x\, x \neq x'$ or $\forall x\, \forall y\, (x + y) = (y + x)$. But we will

see that much of the reason that **Q** is so interesting is *because* it is so weak. In fact, it is just barely strong enough for the incompleteness theorem to hold. Another reason **Q** is interesting is because it has a *finite* set of axioms.

A stronger theory than **Q** (called *Peano arithmetic* **PA**) is obtained by adding a schema of induction to **Q**:

$$(\varphi(\mathsf{0}) \wedge \forall x\,(\varphi(x) \to \varphi(x'))) \to \forall x\,\varphi(x)$$

where $\varphi(x)$ is any formula. If $\varphi(x)$ contains free variables other than $x$, we add universal quantifiers to the front to bind all of them (so that the corresponding instance of the induction schema is a sentence). For instance, if $\varphi(x, y)$ also contains the variable $y$ free, the corresponding instance is

$$\forall y\,((\varphi(\mathsf{0}) \wedge \forall x\,(\varphi(x) \to \varphi(x'))) \to \forall x\,\varphi(x))$$

Using instances of the induction schema, one can prove much more from the axioms of **PA** than from those of **Q**. In fact, it takes a good deal of work to find "natural" statements about the natural numbers that can't be proved in Peano arithmetic!

inc:req:int: defn:representable-fn

**Definition 3.1.** A function $f(x_0, \ldots, x_k)$ from the natural numbers to the natural numbers is said to be *representable in* **Q** if there is a formula $\varphi_f(x_0, \ldots, x_k, y)$ such that whenever $f(n_0, \ldots, n_k) = m$, **Q** proves

1. $\varphi_f(\overline{n_0}, \ldots, \overline{n_k}, \overline{m})$
2. $\forall y\,(\varphi_f(\overline{n_0}, \ldots, \overline{n_k}, y) \to \overline{m} = y)$.

There are other ways of stating the definition; for example, we could equivalently require that **Q** proves $\forall y\,(\varphi_f(\overline{n_0}, \ldots, \overline{n_k}, y) \leftrightarrow y = \overline{m})$.

inc:req:int: thm:representable-iff-comp

**Theorem 3.2.** *A function is representable in* **Q** *if and only if it is computable.*

There are two directions to proving the theorem. The left-to-right direction is fairly straightforward once arithmetization of syntax is in place. The other direction requires more work. Here is the basic idea: we pick "general recursive" as a way of making "computable" precise, and show that every general recursive function is representable in **Q**. Recall that a function is general recursive if it can be defined from zero, the successor function succ, and the projection functions $P^n_i$, using composition, primitive recursion, and regular minimization. So one way of showing that every general recursive function is representable in **Q** is to show that the basic functions are representable, and whenever some functions are representable, then so are the functions defined from them using composition, primitive recursion, and regular minimization. In other words, we might show that the basic functions are representable, and that the representable functions are "closed under" composition, primitive recursion, and regular minimization. This guarantees that every general recursive function is representable.

It turns out that the step where we would show that representable functions are closed under primitive recursion is hard. In order to avoid this step, we show first that in fact we can do without primitive recursion. That is, we show that every general recursive function can be defined from basic functions using composition and regular minimization alone. To do this, we show that primitive recursion can actually be done by a specific regular minimization. However, for this to work, we have to add some additional basic functions: addition, multiplication, and the characteristic function of the identity relation $\chi_=$. Then, we can prove the theorem by showing that all of *these* basic functions are representable in $\mathbf{Q}$, and the representable functions are closed under composition and regular minimization.

## 3.2 Functions Representable in Q are Computable

inc:req:rpc:sec

**Lemma 3.3.** *Every function that is representable in* $\mathbf{Q}$ *is computable.*

*Proof.* Let's first give the intuitive idea for why this is true. If $f(x_0, \ldots, x_k)$ is representable in $\mathbf{Q}$, there is a formula $\varphi(x_0, \ldots, x_k, y)$ such that

$$\mathbf{Q} \vdash \varphi_f(\overline{n_0}, \ldots, \overline{n_k}, \overline{m}) \quad \text{iff} \quad m = f(n_0, \ldots, n_k).$$

To compute $f$, we do the following. List all the possible derivations $\delta$ in the language of arithmetic. This is possible to do mechanically. For each one, check if it is a derivation of a formula of the form $\varphi_f(\overline{n_0}, \ldots, \overline{n_k}, \overline{m})$. If it is, $m$ must be $= f(n_0, \ldots, n_k)$ and we've found the value of $f$. The search terminates because $\mathbf{Q} \vdash \varphi_f(\overline{n_0}, \ldots, \overline{n_k}, \overline{f(n_0, \ldots, n_k)})$, so eventually we find a $\delta$ of the right sort.

This is not quite precise because our procedure operates on derivations and formulas instead of just on numbers, and we haven't explained exactly why "listing all possible derivations" is mechanically possible. But as we've seen, it is possible to code terms, formulas, and derivations by Gödel numbers. We've also introduced a precise model of computation, the general recursive functions. And we've seen that the relation $\mathrm{Prf}_{\mathbf{Q}}(d, y)$, which holds iff $d$ is the Gödel number of a derivation of the formula with Gödel number $x$ from the axioms of $\mathbf{Q}$, is (primitive) recursive. Other primitive recursive functions we'll need are num (Proposition 2.6) and Subst (Proposition 2.11). From these, it is possible to define $f$ by minimization; thus, $f$ is recursive.

First, define

$$\begin{aligned} A(n_0, \ldots, n_k, m) = \\ \mathrm{Subst}(\mathrm{Subst}(\ldots \mathrm{Subst}(&{}^{\ulcorner}\varphi_f{}^{\urcorner}, \mathrm{num}(n_0), {}^{\ulcorner}x_0{}^{\urcorner}), \\ &\ldots), \mathrm{num}(n_k), {}^{\ulcorner}x_k{}^{\urcorner}), \mathrm{num}(m), {}^{\ulcorner}y{}^{\urcorner}) \end{aligned}$$

This looks complicated, but it's just the function $A(n_0, \ldots, n_k, m) = {}^{\ulcorner}\varphi_f(\overline{n_0}, \ldots, \overline{n_k}, \overline{m})^{\urcorner}$.

Now, consider the relation $R(n_0, \ldots, n_k, s)$ which holds if $(s)_0$ is the Gödel number of a derivation from $\mathbf{Q}$ of $\varphi_f(\overline{n_0}, \ldots, \overline{n_k}, \overline{(s)_1})$:

$$R(n_0, \ldots, n_k, s) \quad \text{iff} \quad \mathrm{Prf}_{\mathbf{Q}}((s)_0, A(n_0, \ldots, n_k, (s)_1)$$

If we can find an $s$ such that $R(n_0, \ldots, n_k, s)$ hold, we have found a pair of numbers—$(s)_0$ and $(s_1)$—such that $(s)_0$ is the Gödel number of a derivation of $A_f(\overline{n_0}, \ldots, \overline{n_k}, (s)_1)$. So looking for $s$ is like looking for the pair $d$ and $m$ in the informal proof. And a computable function that "looks for" such an $s$ can be defined by regular minimization. Note that $R$ is regular: for every $n_0, \ldots, n_k$, there is a derivation $\delta$ of $\mathbf{Q} \vdash \varphi_f(\overline{n_0}, \ldots, \overline{n_k}, \overline{f(n_0, \ldots, n_k)})$, so $R(n_0, \ldots, n_k, s)$ holds for $s = \langle {}^{\#}\delta^{\#}, f(n_0, \ldots, n_k)\rangle$. So, we can write $f$ as

$$f(n_0, \ldots, n_k) = (\mu s\, R(n_0, \ldots, n_k, s))_1.$$

□

## 3.3 The Beta Function Lemma

inc:req:bet: sec

In order to show that we can carry out primitive recursion if addition, multiplication, and $\chi_=$ are available, we need to develop functions that handle sequences. (If we had exponentiation as well, our task would be easier.) When we had primitive recursion, we could define things like the "$n$-th prime," and pick a fairly straightforward coding. But here we do not have primitive recursion—in fact we want to show that we can do primitive recursion using minimization—so we need to be more clever.

inc:req:bet: lem:beta

**Lemma 3.4.** *There is a function $\beta(d, i)$ such that for every sequence $a_0, \ldots, a_n$ there is a number $d$, such that for every $i \leq n$, $\beta(d, i) = a_i$. Moreover, $\beta$ can be defined from the basic functions using just composition and regular minimization.*

Think of $d$ as coding the sequence $\langle a_0, \ldots, a_n\rangle$, and $\beta(d, i)$ returning the $i$-th element. (Note that this "coding" does *not* use the prower-of-primes coding we're already familiar with!). The lemma is fairly minimal; it doesn't say we can concatenate sequences or append elements, or even that we can *compute* $d$ from $a_0, \ldots, a_n$ using functions definable by composition and regular minimization. All it says is that there is a "decoding" function such that every sequence is "coded."

The use of the notation $\beta$ is Gödel's. To repeat, the hard part of proving the lemma is defining a suitable $\beta$ using the seemingly restricted resources, i.e., using just composition and minimization—however, we're allowed to use addition, multiplication, and $\chi_=$. There are various ways to prove this lemma, but one of the cleanest is still Gödel's original method, which used a number-theoretic fact called the Chinese Remainder theorem.

**Definition 3.5.** Two natural numbers $a$ and $b$ are *relatively prime* if their greatest common divisor is 1; in other words, they have no other divisors in common.

**Definition 3.6.** $a \equiv b \mod c$ means $c \mid (a - b)$, i.e., $a$ and $b$ have the same remainder when divided by $c$.

Here is the *Chinese Remainder theorem*:

**Theorem 3.7.** *Suppose $x_0, \ldots, x_n$ are (pairwise) relatively prime. Let $y_0, \ldots, y_n$ be any numbers. Then there is a number $z$ such that*

$$\begin{aligned} z &\equiv y_0 \mod x_0 \\ z &\equiv y_1 \mod x_1 \\ &\vdots \\ z &\equiv y_n \mod x_n. \end{aligned}$$

Here is how we will use the Chinese Remainder theorem: if $x_0, \ldots, x_n$ are bigger than $y_0, \ldots, y_n$ respectively, then we can take $z$ to code the sequence $\langle y_0, \ldots, y_n \rangle$. To recover $y_i$, we need only divide $z$ by $x_i$ and take the remainder. To use this coding, we will need to find suitable values for $x_0, \ldots, x_n$.

A couple of observations will help us in this regard. Given $y_0, \ldots, y_n$, let

$$j = \max(n, y_0, \ldots, y_n) + 1,$$

and let

$$\begin{aligned} x_0 &= 1 + j! \\ x_1 &= 1 + 2 \cdot j! \\ x_2 &= 1 + 3 \cdot j! \\ &\vdots \\ x_n &= 1 + (n+1) \cdot j! \end{aligned}$$

Then two things are true:

1. $x_0, \ldots, x_n$ are relatively prime.
2. For each $i$, $y_i < x_i$.

To see that (1) is true, note that if $p$ is a prime number and $p \mid x_i$ and $p \mid x_k$, then $p \mid 1 + (i+1)j!$ and $p \mid 1 + (k+1)j!$. But then $p$ divides their difference,

$$(1 + (i+1)j!) - (1 + (k+1)j!) = (i - k)j!.$$

Since $p$ divides $1+(i+1)j!$, it can't divide $j!$ as well (otherwise, the first division would leave a remainder of 1). So $p$ divides $i - k$, since $p$ divides $(i - k)j!$. But $|i - k|$ is at most $n$, and we have chosen $j > n$, so this implies that $p \mid j!$, again a contradiction. So there is no prime number dividing both $x_i$ and $x_k$. Clause (2) is easy: we have $y_i < j < j! < x_i$.

Now let us prove the $\beta$ function lemma. Remember that we can use 0, successor, plus, times, $\chi_=$, projections, and any function defined from them

using composition and minimization applied to regular functions. We can also use a relation if its characteristic function is so definable. As before we can show that these relations are closed under boolean combinations and bounded quantification; for example:

1. $\text{not}(x) = \chi_{=}(x, 0)$
2. $(\min x \leq z)\, R(x, y) = \mu x\, (R(x, y) \vee x = z)$
3. $(\exists x \leq z)\, R(x, y) \Leftrightarrow R((\min x \leq z)\, R(x, y), y)$

We can then show that all of the following are also definable without primitive recursion:

1. The pairing function, $J(x, y) = \frac{1}{2}[(x + y)(x + y + 1)] + x$
2. Projections
$$K(z) = (\min x \leq q)\, (\exists y \leq z\, [z = J(x, y)])$$
and
$$L(z) = (\min y \leq q)\, (\exists x \leq z\, [z = J(x, y)]).$$
3. $x < y$
4. $x \mid y$
5. The function $\text{rem}(x, y)$ which returns the remainder when $y$ is divided by $x$

Now define
$$\beta^*(d_0, d_1, i) = \text{rem}(1 + (i + 1)d_1, d_0)$$
and
$$\beta(d, i) = \beta^*(K(d), L(d), i).$$
This is the function we need. Given $a_0, \ldots, a_n$, as above, let
$$j = \max(n, a_0, \ldots, a_n) + 1,$$
and let $d_1 = j!$. By the observations above, we know that $1+d_1, 1+2d_1, \ldots, 1+(n+1)d_1$ are relatively prime and all are bigger than $a_0, \ldots, a_n$. By the Chinese Remainder theorem there is a value $d_0$ such that for each $i$,
$$d_0 \equiv a_i \mod (1 + (i + 1)d_1)$$
and so (because $d_1$ is greater than $a_i$),
$$a_i = \text{rem}(1 + (i + 1)d_1, d_0).$$
Let $d = J(d_0, d_1)$. Then for each $i \leq n$, we have
$$\begin{aligned} \beta(d, i) &= \beta^*(d_0, d_1, i) \\ &= \text{rem}(1 + (i + 1)d_1, d_0) \\ &= a_i \end{aligned}$$
which is what we need. This completes the proof of the $\beta$-function lemma.

## 3.4 Simulating Primitive Recursion

inc:req:pri:sec

Now we can show that definition by primitive recursion can be "simulated" by regular minimization using the beta function. Suppose we have $f(\vec{x})$ and $g(\vec{x}, y, z)$. Then the function $h(x, \vec{z})$ defined from $f$ and $g$ by primitive recursion is

$$h(\vec{x}, y) = f(\vec{z})$$
$$h(\vec{x}, y+1) = g(\vec{x}, y, h(\vec{x}, y)).$$

We need to show that $h$ can be defined from $f$ and $g$ using just composition and regular minimization, using the basic functions and functions defined from them using composition and regular minimization (such as $\beta$).

inc:req:pri:lem:prim-rec

**Lemma 3.8.** *If $h$ can be defined from $f$ and $g$ using primitive recursion, it can be defined from $f$, $g$, the functions* zero, succ, $P^n_i$, add, mult, $\chi_=$, *using composition and regular minimization.*

*Proof.* First, define an auxiliary function $\hat{h}(\vec{x}, y)$ which returns the least number $d$ such that $d$ codes a sequence which satisfies

1. $(d)_0 = f(\vec{x})$, and
2. for each $i < x$, $(d)_{i+1} = g(\vec{x}, i, (d)_i)$,

where now $(d)_i$ is short for $\beta(d, i)$. In other words, $\hat{h}$ returns the sequence $\langle h(\vec{x}, 0), h(\vec{x}, 1), \ldots, h(\vec{x}, y)\rangle$. We can write $\hat{h}$ as

$$\hat{h}(\vec{x}, y) = \mu d\, (\beta(d, 0) = f(\vec{x}) \wedge (\forall i < y)\, \beta(d, i+1) = g(\vec{x}, i, \beta(d, i))).$$

Note: no primitive recursion is needed here, just minimization. The function we minimize is regular because of the beta function lemma Lemma 3.4.

But now we have

$$h(\vec{x}, y) = \beta(\hat{h}(\vec{x}, y), y),$$

so $h$ can be defined from the basic functions using just composition and regular minimization. □

## 3.5 Basic Functions are Representable in Q

inc:req:bre:sec

First we have to show that all the basic functions are representable in **Q**. In the end, we need to show how to assign to each $k$-ary basic function $f(x_0, \ldots, x_{k-1})$ a formula $\varphi_f(x_0, \ldots, x_{k-1}, y)$ that represents it.

We will be able to represent zero, successor, plus, times, the characteristic function for equality, and projections. In each case, the appropriate representing function is entirely straightforward; for example, zero is represented by the formula $y = 0$, successor is represented by the formula $x'_0 = y$, and addition is represented by the formula $(x_0 + x_1) = y$. The work involves showing that

**Q** can prove the relevant sentences; for example, saying that addition is represented by the formula above involves showing that for every pair of natural numbers $m$ and $n$, **Q** proves

$$\overline{n} + \overline{m} = \overline{n+m} \text{ and}$$
$$\forall y\,((\overline{n} + \overline{m}) = y \to y = \overline{n+m}).$$

inc:req:bre: prop:rep-zero **Proposition 3.9.** *The zero function* $\mathrm{zero}(x) = 0$ *is represented in* **Q** *by* $y = \mathsf{0}$.

inc:req:bre: prop:rep-succ **Proposition 3.10.** *The successor function* $\mathrm{succ}(x) = x + 1$ *is represented in* **Q** *by* $y = x'$.

inc:req:bre: prop:rep-proj **Proposition 3.11.** *The projection function* $P^n_i(x_0, \ldots, x_{n-1}) = x_i$ *is represented in* **Q** *by* $y = x_i$.

**Problem 3.1.** Prove that $y = \mathsf{0}$, $y = x'$, and $y = x_i$ represent zero, succ, and $P^n_i$, respectively.

inc:req:bre: prop:rep-id **Proposition 3.12.** *The characteristic function of* $=$,

$$\chi_=(x_0, x_1) = \begin{cases} 1 & \text{if } x_0 = x_1 \\ 0 & \text{otherwise} \end{cases}$$

*is represented in* **Q** *by*

$$(x_0 = x_1 \wedge y = \overline{1}) \vee (x_0 \neq x_1 \wedge y = \overline{0}).$$

The proof requires the following lemma.

inc:req:bre: lem:q-proves-neq **Lemma 3.13.** *Given natural numbers* $n$ *and* $m$*, if* $n \neq m$*, then* $\mathbf{Q} \vdash \overline{n} \neq \overline{m}$.

*Proof.* Use induction on $n$ to show that for every $m$, if $n \neq m$, then $Q \vdash \overline{n} \neq \overline{m}$.

In the base case, $n = 0$. If $m$ is not equal to 0, then $m = k + 1$ for some natural number $k$. We have an axiom that says $\forall x\, 0 \neq x'$. By a quantifier axiom, replacing $x$ by $\overline{k}$, we can conclude $0 \neq \overline{k}'$. But $\overline{k}'$ is just $\overline{m}$.

In the induction step, we can assume the claim is true for $n$, and consider $n+1$. Let $m$ be any natural number. There are two possibilities: either $m = 0$ or for some $k$ we have $m = k + 1$. The first case is handled as above. In the second case, suppose $n + 1 \neq k + 1$. Then $n \neq k$. By the induction hypothesis for $n$ we have $\mathbf{Q} \vdash \overline{n} \neq \overline{k}$. We have an axiom that says $\forall x \forall y\, x' = y' \to x = y$. Using a quantifier axiom, we have $\overline{n}' = \overline{k}' \to \overline{n} = \overline{k}$. Using propositional logic, we can conclude, in **Q**, $\overline{n} \neq \overline{k} \to \overline{n}' \neq \overline{k}'$. Using modus ponens, we can conclude $\overline{n}' \neq \overline{k}'$, which is what we want, since $\overline{k}'$ is $\overline{m}$. □

Note that the lemma does not say much: in essence it says that **Q** can prove that different numerals denote different objects. For example, **Q** proves $0'' \neq 0'''$. But showing that this holds in general requires some care. Note also that although we are using induction, it is induction *outside* of **Q**. explanation

*Proof of Proposition 3.12.* If $n = m$, then $\overline{n}$ and $\overline{m}$ are the same term, and $\chi_=(n, m) = 1$. But $\mathbf{Q} \vdash (\overline{n} = \overline{m} \wedge \overline{1} = \overline{1})$, so it proves $\varphi_=(\overline{n}, \overline{m}, \overline{1})$. If $n \neq m$, then $\chi_=(n, m) = 0$. By Lemma 3.13, $\mathbf{Q} \vdash \overline{n} \neq \overline{m}$ and so also $(\overline{n} \neq \overline{m} \wedge \mathsf{o} = \mathsf{o})$. Thus $\mathbf{Q} \vdash \varphi_=(\overline{n}, \overline{m}, \overline{0})$.

For the second part, we also have two cases. If $n = m$, we have to show that that $\mathbf{Q} \vdash \forall(\varphi_=(\overline{n}, \overline{m}, y) \to y = \overline{1})$. Arguing informally, suppose $\varphi_=(\overline{n}, \overline{m}, y)$, i.e.,

$$(\overline{n} = \overline{n} \wedge y = \overline{1}) \vee (\overline{n} \neq \overline{n} \wedge y = \overline{0})$$

The left disjunct implies $y = \overline{1}$ by logic; the right contradicts $\overline{n} = \overline{n}$ which is provable by logic.

Suppose, on the other hand, that $n \neq m$. Then $\varphi_=(\overline{n}, \overline{m}, y)$ is

$$(\overline{n} = \overline{m} \wedge y = \overline{1}) \vee (\overline{n} \neq \overline{m} \wedge y = \overline{0})$$

Here, the left disjunct contradicts $\overline{n} \neq \overline{m}$, which is provable in $\mathbf{Q}$ by Lemma 3.13; the right disjunct entails $y = \overline{0}$. □

**Proposition 3.14.** *The addition function* $\mathrm{add}(x_0, x_1) = x_0 + x_1$ *is represented in* $\mathbf{Q}$ *by* inc:req:bre: prop:rep-add

$$y = (x_0 + x_1).$$

**Lemma 3.15.** $\mathbf{Q} \vdash (\overline{n} + \overline{m}) = \overline{n + m}$ inc:req:bre: lem:q-proves-add

*Proof.* We prove this by induction on $m$. If $m = 0$, the claim is that $\mathbf{Q} \vdash (\overline{n} + \mathsf{o}) = \overline{n}$. This follows by axiom $Q_4$. Now suppose the claim for $m$; let's prove the claim for $m + 1$, i.e., prove that $\mathbf{Q} \vdash (\overline{n} + \overline{m + 1}) = \overline{n + m + 1}$. Note that $\overline{m + 1}$ is just $\overline{m}'$, and $\overline{n + m + 1}$ is just $\overline{n + m}'$. By axiom $Q_5$, $\mathbf{Q} \vdash (\overline{n} + \overline{m}') = (\overline{n} + \overline{m})'$. By induction hypothesis, $\mathbf{Q} \vdash (\overline{n} + \overline{m}) = \overline{n + m}$. So $\mathbf{Q} \vdash (\overline{n} + \overline{m}') = \overline{n + m}'$. □

*Proof of Proposition 3.14.* The formula $\varphi_{\mathrm{add}}(x_0, x_1, y)$ representing add is $y = (x_0 + x_1)$. First we show that if $\mathrm{add}(n, m) = k$, then $\mathbf{Q} \vdash \varphi_{\mathrm{add}}(\overline{n}, \overline{m}, \overline{k})$, i.e., $\mathbf{Q} \vdash \overline{k} = (\overline{n} + \overline{m})$. But since $k = n + m$, $\overline{k}$ just is $\overline{n + m}$, and we've shown in Lemma 3.15 that $\mathbf{Q} \vdash (\overline{n} + \overline{m}) = \overline{n + m}$.

We also have to show that if $\mathrm{add}(n, m) = k$, then

$$\mathbf{Q} \vdash \forall y\, (\varphi_{\mathrm{add}}(\overline{n}, \overline{m}, y) \to y = \overline{k}).$$

Suppose we have $(\overline{n} + \overline{m}) = y$. Since

$$\mathbf{Q} \vdash (\overline{n} + \overline{m}) = \overline{n + m},$$

we can replace the left side with $\overline{n + m}$ and get $\overline{n + m} = y$, for arbitrary $y$. □

**Proposition 3.16.** *The multiplication function* $\mathrm{mult}(x_0, x_1) = x_0 \cdot x_1$ *is represented in* $\mathbf{Q}$ *by* inc:req:bre: prop:rep-mult

$$y = (x_0 \times x_1).$$

*Proof.* Exercise. □

inc:req:bre: lem:q-proves-mult **Lemma 3.17.** $\mathbf{Q} \vdash (\overline{n} \times \overline{m}) = \overline{n \cdot m}$

*Proof.* Exercise. □

**Problem 3.2.** Prove Lemma 3.17.

**Problem 3.3.** Use Lemma 3.17 to prove Proposition 3.16.

explanation

Recall that we use $\times$ for the function symbol of the language of arithmetic, and $\cdot$ for the ordinary multiplication operation on numbers. So $\cdot$ can appear between expressions for numbers (such as in $m \cdot n$) while $\times$ appears only between terms of the language of arithmetic (such as in $(\overline{m} \times \overline{n})$). EVen more confusingly, $+$ is used for both the function symbol and the addition operation. When it appears between terms—e.g., in $(\overline{n} + \overline{m})$—it is the 2-place function symbol of the language of arithmetic, and when it appears between numbers—e.g., in $n + m$—it is the operation. This includes the case $\overline{n + m}$: this is the standard numeral corresponding to the number $n + m$.

## 3.6 Composition is Representable in Q

inc:req:cmp: sec Suppose $h$ is defined by

$$h(x_0, \ldots, x_{l-1}) = f(g_0(x_0, \ldots, x_{l-1}), \ldots, g_{k-1}(x_0, \ldots, x_{l-1})).$$

where we have already found formulas $\varphi_f, \varphi_{g_0}, \ldots, \varphi_{g_{k-1}}$ representing the functions $f$, and $g_0$, …, $g_{k-1}$, respectively. We have to find a formula $\varphi_h$ representing $h$.

Let's start with a simple case, where all functions are 1-place, i.e., consider $h(x) = f(g(x))$. If $\varphi_f(y, z)$ represents $f$, and $\varphi_g(x, y)$ represents $g$, we need a formula $\varphi_h(x, z)$ that represents $h$. Note that $h(x) = z$ iff there is a $y$ such that both $z = f(y)$ and $y = g(x)$. (If $h(x) = z$, then $g(x)$ is such a $y$; if such a $y$ exists, then since $y = g(x)$ and $z = f(y)$, $z = f(g(x))$.) This suggests that $\exists y\,(\varphi_g(x, y) \wedge \varphi_f(y, z))$ is a good candidate for $\varphi_h(x, z)$. We just have to verify that $\mathbf{Q}$ proves the relevant formulas.

inc:req:cmp: prop:rep1 **Proposition 3.18.** *If $h(n) = m$, then $\mathbf{Q} \vdash \varphi_h(\overline{n}, \overline{m})$.*

*Proof.* Suppose $h(n) = m$, i.e., $f(g(n)) = m$. Let $k = g(n)$. Then

$$\mathbf{Q} \vdash \varphi_g(\overline{n}, \overline{k})$$

since $\varphi_g$ represents $g$, and

$$\mathbf{Q} \vdash \varphi_f(\overline{k}, \overline{m})$$

since $\varphi_f$ represents $f$. Thus,

$$\mathbf{Q} \vdash \varphi_g(\overline{n}, \overline{k}) \wedge \varphi_f(\overline{k}, \overline{m})$$

and consequently also

$$\mathbf{Q} \vdash \exists y\,(\varphi_g(\overline{n}, y) \wedge \varphi_f(y, \overline{m})),$$

i.e., $\mathbf{Q} \vdash \varphi_h(\overline{n}, \overline{m})$. □

**Proposition 3.19.** *If $h(n) = m$, then $\mathbf{Q} \vdash \forall z\,(\varphi_h(\overline{n}, z) \to z = \overline{m})$.* inc:req:cmp: prop:rep2

*Proof.* Suppose $h(n) = m$, i.e., $f(g(n)) = m$. Let $k = g(n)$. Then

$$\mathbf{Q} \vdash \forall y\,(\varphi_g(\overline{n}, y) \to y = \overline{k})$$

since $\varphi_g$ represents $g$, and

$$\mathbf{Q} \vdash \forall z\,(\varphi_f(\overline{k}, z) \to z = \overline{m})$$

since $\varphi_f$ represents $f$. Using just a little bit of logic, we can show that also

$$\mathbf{Q} \vdash \forall z\,(\exists y\,(\varphi_g(\overline{n}, y) \wedge \varphi_f(y, z)) \to z = \overline{m}).$$

i.e., $\mathbf{Q} \vdash \forall y\,(\varphi_h(\overline{n}, y) \to y = \overline{m})$. □

The same idea works in the more complex case where $f$ and $g_i$ have arity greater than 1.

**Proposition 3.20.** *If $\varphi_f(y_0, \ldots, y_{k-1}, z)$ represents $f(y_0, \ldots, y_{k-1})$ in $\mathbf{Q}$, and $\varphi_{g_i}(x_0, \ldots, x_{l-1}, y)$ represents $g_i(x_0, \ldots, x_{l-1})$ in $\mathbf{Q}$, then* inc:req:cmp: prop:rep-composition

$$\exists y_0, \ldots\, \exists y_{k-1}\,(\varphi_{g_0}(x_0, \ldots, x_{l-1}, y_0) \wedge \cdots \wedge \varphi_{g_{k-1}}(x_0, \ldots, x_{l-1}, y_{k-1}) \wedge \varphi_f(y_0, \ldots, y_{k-1}, z))$$

*represents*

$$h(x_0, \ldots, x_{k-1}) = f(g_0(x_0, \ldots, x_{k-1}), \ldots, g_0(x_0, \ldots, x_{k-1})).$$

*Proof.* Exercise. □

**Problem 3.4.** Using the proofs of Proposition 3.19 and Proposition 3.19 as a guide, carry out the proof of Proposition 3.20 in detail.

## 3.7 Regular Minimization is Representable in Q

Let's consider unbounded search. Suppose $g(x, z)$ is regular and representable in $\mathbf{Q}$, say by the formula $\varphi_g(x, z, y)$. Let $f$ be defined by $f(z) = \mu x\,[g(x, z) = 0]$. We would like to find a formula $\varphi_f(z, y)$ representing $f$. The value of $f(z)$ is that number $x$ which (a) satisfies $g(x, z) = 0$ and (b) is the least such, i.e., for any $w < x$, $g(w, z) \neq 0$. So the following is a natural choice: inc:req:min: sec

$$\varphi_f(z, y) \equiv \varphi_g(y, z, 0) \wedge \forall w\,(w < y \to \neg\varphi_g(w, z, 0)).$$

In the general case, of course, we would have to replace $z$ with $z_0, \ldots, z_k$.

The proof, again, will involve some lemmas about things $\mathbf{Q}$ is strong enough to prove.

inc:req:min: lem:succ

**Lemma 3.21.** *For every constant symbol $a$ and every natural number $n$,*

$$\mathbf{Q} \vdash (a' + \overline{n}) = (a + \overline{n})'.$$

*Proof.* The proof is, as usual, by induction on $n$. In the base case, $n = 0$, we need to show that $\mathbf{Q}$ proves $(a' + 0) = (a + 0)'$. But we have:

inc:req:min: step1
$$\mathbf{Q} \vdash (a' + 0) = a' \quad \text{by axiom } Q_4 \tag{3.1}$$

inc:req:min: step2
$$\mathbf{Q} \vdash (a + 0) = a \quad \text{by axiom } Q_4 \tag{3.2}$$

inc:req:min: step3
$$\mathbf{Q} \vdash (a + 0)' = a' \quad \text{by eq. (3.2)} \tag{3.3}$$

$$\mathbf{Q} \vdash (a' + 0) = (a + 0)' \quad \text{by eq. (3.1) and eq. (3.3)}$$

In the induction step, we can assume that we have shown that $\mathbf{Q} \vdash (a' + \overline{n}) = (a + \overline{n})'$. Since $\overline{n+1}$ is $\overline{n}'$, we need to show that $\mathbf{Q}$ proves $(a' + \overline{n}') = (a + \overline{n}')'$. We have:

inc:req:min: step5
$$\mathbf{Q} \vdash (a' + \overline{n}') = (a' + \overline{n})' \quad \text{by axiom } Q_5 \tag{3.4}$$

inc:req:min: step6
$$\mathbf{Q} \vdash (a' + \overline{n}') = (a + \overline{n}')' \quad \text{inductive hypothesis} \tag{3.5}$$

$$\mathbf{Q} \vdash (a' + \overline{n})' = (a + \overline{n}')' \quad \text{by eq. (3.4) and eq. (3.5).}$$

□

It is again worth mentioning that this is weaker than saying that $\mathbf{Q}$ proves $\forall x \, \forall y \, (x' + y) = (x + y)'$. Although this sentence is true in $\mathfrak{N}$, $\mathbf{Q}$ does not prove it.

inc:req:min: lem:less

**Lemma 3.22.**

1. $\mathbf{Q} \vdash \forall x \, \neg x < \mathsf{o}$.
2. *For every natural number $n$,*

$$\mathbf{Q} \vdash \forall x \, (x < \overline{n+1} \to (x = \mathsf{o} \vee \cdots \vee x = \overline{n})).$$

*Proof.* Let us do 1 and part of 2, informally (i.e., only giving hints as to how to construct the formal derivation).

For part 1, by the definition of $<$, we need to prove $\neg \exists y \, (y' + a) = \mathsf{o}$ in $\mathbf{Q}$, which is equivalent (using the axioms and rules of first-order logic) to $\forall b \, (y' + a) \neq 0$. Here is the idea: suppose $(y' + b) = \mathsf{o}$. If $a = \mathsf{o}$, we have $(y' + \mathsf{o}) = \mathsf{o}$. But by axiom $Q_4$ of $\mathbf{Q}$, we have $(b' + \mathsf{o}) = b'$, and by axiom $Q_2$ we have $b' \neq \mathsf{o}$, a contradiction. So $\forall y \, (y' + a) \neq \mathsf{o}$. If $a \neq \mathsf{o}$, by axiom $Q_3$, there is a $c$ such that $a = c'$. But then we have $(b' + c') = 0$. By axiom $Q_5$, we have $(b' + c)' = \mathsf{o}$, again contradicting axiom $Q_2$.

For part 2, use induction on $n$. Let us consider the base case, when $n = 0$. In that case, we need to show $a < \overline{1} \to a = \mathsf{o}$. Suppose $a < \overline{1}$. Then by the defining axiom for $<$, we have $\exists y \, (y' + a) = \mathsf{o}'$.

Suppose $b$ has that property, i.e., we have $b' + a = \mathsf{o}'$. We need to show $a = \mathsf{o}$. By axiom $Q_3$, if $a \neq \mathsf{o}$, we get $a = c'$ for some $z$. Then we have $(b' + c') = \mathsf{o}'$. By axiom $Q_5$ of $\mathbf{Q}$, we have $(b' + c)' = \mathsf{o}'$. By axiom $Q_1$, we have $(b' + c) = \mathsf{o}$. But this means, by definition, $z < \mathsf{o}$, contradicting part 1. □

**Lemma 3.23.** *For every $m \in \mathbb{N}$,* inc:req:min: lem:trichotomy

$$\mathbf{Q} \vdash \forall y\,((y < \overline{m} \vee \overline{m} < y) \vee y = \overline{m}).$$

*Proof.* By induction on $m$. First, consider the case $m = 0$. $\mathbf{Q} \vdash \forall y\,(y \neq \mathsf{o} \to \exists z\, y = z')$ by $Q_3$. But if $b = c'$, then $(c' + \mathsf{o}) = (b + \mathsf{o})$ by the logic of $=$. By $Q_4$, $(b + \mathsf{o}) = b$, so we have $(c' + \mathsf{o}) = b$, and hence $\exists z\,(z' + \mathsf{o}) = b$. By the definition of $<$ in $Q_8$, $\mathsf{o} < b$. If $\mathsf{o} < b$, then also $\mathsf{o} < b \vee b < \mathsf{o}$. We obtain: $b \neq \mathsf{o} \to (\mathsf{o} < b \vee b < \mathsf{o})$, which is equivalent to $(\mathsf{o} < b \vee b < \mathsf{o}) \vee b = \mathsf{o}$.

Now suppose we have

$$\mathbf{Q} \vdash \forall y\,((y < \overline{m} \vee \overline{m} < y) \vee y = \overline{m})$$

and we want to show

$$\mathbf{Q} \vdash \forall y\,((y < \overline{m+1} \vee \overline{m+1} < y) \vee y = \overline{m+1})$$

The first disjunct $b < \overline{m}$ is equivalent (by $Q_8$) to $\exists z\,(z' + b) = \overline{m}$. Suppose $c$ has this property. If $(c' + b) = \overline{m}$, then also $(c' + b)' = \overline{m}'$. By $Q_4$, $(c' + b)' = (c'' + b)$. Hence, $(c'' + b) = \overline{m}'$. We get $\exists u\,(u' + b) = \overline{m+1}$ by existentially generalizing on $c'$ and keeping in mind that $\overline{m}'$ is $\overline{m+1}$. Hence, if $b < \overline{m}$ then $b < \overline{m+1}$.

Now suppose $\overline{m} < b$, i.e., $\exists z\,(z' + \overline{m}) = b$. Suppose $c$ is such a $z$. By $Q_3$ and some logic, we have $c = \mathsf{o} \vee \exists u\, c = u'$. If $c = \mathsf{o}$, we have $(\mathsf{o}' + \overline{m}) = b$. Since $\mathbf{Q} \vdash (\mathsf{o}' + \overline{m}) = \overline{m+1}$, we have $b = \overline{m+1}$. Now suppose $\exists u\, c = u'$. Let $d$ be such a $u$. Then:

$$\begin{aligned}
b &= (c' + \overline{m}) && \text{by assumption}\\
(c' + \overline{m}) &= (d'' + \overline{m}) && \text{from } c = d'\\
(d'' + \overline{m}) &= (d' + \overline{m})' && \text{by Lemma 3.21}\\
(d' + \overline{m})' &= (d' + \overline{m}') && \text{by } Q_5\text{, so}\\
b &= (d' + \overline{m+1})
\end{aligned}$$

By existential generalization, $\exists u\,(u' + \overline{m+1}) = b$, i.e., $\overline{m+1} < b$. So, if $\overline{m} < b$, then $\overline{m+1} < b \vee b = \overline{m+1}$.

Finally, assume $b = \overline{m}$. Then, since $\mathbf{Q} \vdash (\mathsf{o}' + \overline{m}) = \overline{m+1}$, $(\mathsf{o}' + b) = \overline{m+1}$. From this we get $\exists z\,(z' + b) = \overline{m+1}$, or $b < \overline{m+1}$.

Hence, from each disjunct of the case for $m$, we can obtain the case for $m+1$. □

**Proposition 3.24.** *If $\varphi_g(x, z, y)$ represents $g(x, y)$ in $\mathbf{Q}$, then* inc:req:min: prop:rep-minimization

$$\varphi_f(z, y) \equiv \varphi_g(y, z, \mathsf{o}) \wedge \forall w\,(w < y \to \neg\varphi_g(w, z, \mathsf{o})).$$

*represents $f(z) = \mu x\,[g(x, z) = 0]$.*

*Proof.* First we show that if $f(n) = m$, then $\mathbf{Q} \vdash \varphi_f(\overline{n}, \overline{m})$, i.e.,

$$\mathbf{Q} \vdash \varphi_g(\overline{m}, \overline{n}, \mathsf{o}) \wedge \forall w\,(w < \overline{m} \to \neg\varphi_g(w, \overline{n}, \mathsf{o})).$$

Since $\varphi_g(x, z, y)$ represents $g(x, z)$ and $g(m, n) = 0$ if $f(n) = m$, we have

$$\mathbf{Q} \vdash \varphi_g(\overline{m}, \overline{n}, \mathsf{o}).$$

If $f(n) = m$, then for every $k < m$, $g(k, n) \neq 0$. So

$$\mathbf{Q} \vdash \neg\varphi_g(\overline{k}, \overline{n}, \mathsf{o}).$$

We get that

inc:req:min: rep-less

$$\mathbf{Q} \vdash \forall w\,(w < \overline{m} \to \neg\varphi_g(w, \overline{n}, \mathsf{o})). \tag{3.6}$$

by Lemma 3.22 (by (1) in case $m = 0$ and by (2) otherwise).

Now let's show that if $f(n) = m$, then $\mathbf{Q} \vdash \forall y\,(\varphi_f(\overline{n}, y) \to y = \overline{m})$. We again sketch the argument informally, leaving the formalization to the reader.

Suppose $\varphi_f(\overline{n}, b)$. From this we get (a) $\varphi_g(b, \overline{n}, \mathsf{o})$ and (b) $\forall w\,(w < b \to \neg\varphi_g(w, \overline{n}, \mathsf{o}))$. By Lemma 3.23, $(b < \overline{m} \vee \overline{m} < b) \vee b = \overline{m}$. We'll show that both $b < \overline{m}$ and $\overline{m} < b$ leads to a contradiction.

If $\overline{m} < b$, then $\neg\varphi_g(\overline{m}, \overline{n}, \mathsf{o})$ from (b). But $m = f(n)$, so $g(m, n) = 0$, and so $\mathbf{Q} \vdash \varphi_g(\overline{m}, \overline{n}, \mathsf{o})$ since $\varphi_g$ represents $g$. So we have a contradiction.

Now suppose $b < \overline{m}$. Then since $\mathbf{Q} \vdash \forall w\,(w < \overline{m} \to \neg\varphi_g(w, \overline{n}, \mathsf{o}))$ by eq. (3.6), we get $\neg\varphi_g(b, \overline{n}, \mathsf{o})$. This again contradicts (a). □

## 3.8 Computable Functions are Representable in Q

inc:req:crq: sec

**Theorem 3.25.** *Every computable function is representable in* **Q**.

*Proof.* For definiteness, and using the Church-Turing Thesis, let's say that a function is computable iff it is general recursive. The general recursive functions are those which can be defined from the zero function zero, the successor function succ, and the projection function $P^n_i$ using composition, primitive recursion, and regular minimization. By Lemma 3.8, any function $h$ that can be defined from $f$ and $g$ can also be defined using composition and regular minimization from $f$, $g$, and zero, succ, $P^n_i$, add, mult, $\chi_=$. Consequently, a function is general recursive iff it can be defined from zero, succ, $P^n_i$, add, mult, $\chi_=$ using composition and regular minimization.

We've furthermore shown that the basic functions in question are representable in **Q** (Propositions 3.9 to 3.12, 3.14 and 3.16), and that any function defined from representable functions by composition or regular minimization (Proposition 3.20, Proposition 3.24) is also representable. Thus every general recursive function is representable in **Q**. □

We have shown that the set of computable functions can be characterized as the set of functions representable in **Q**. In fact, the proof is more general. From the definition of representability, it is not hard to see that any theory

explanation

extending **Q** (or in which one can interpret **Q**) can represent the computable functions. But, conversely, in any proof system in which the notion of proof is computable, every representable function is computable. So, for example, the set of computable functions can be characterized as the set of functions representable in Peano arithmetic, or even Zermelo-Fraenkel set theory. As Gödel noted, this is somewhat surprising. We will see that when it comes to provability, questions are very sensitive to which theory you consider; roughly, the stronger the axioms, the more you can prove. But across a wide range of axiomatic theories, the representable functions are exactly the computable ones; stronger theories do not represent more functions as long as they are axiomatizable.

## 3.9 Representing Relations

inc:req:rel:sec

Let us say what it means for a *relation* to be representable.

inc:req:rel:defn:representing-relations

**Definition 3.26.** A relation $R(x_0, \ldots, x_k)$ on the natural numbers is *representable in* **Q** if there is a formula $\varphi_R(x_0, \ldots, x_k)$ such that whenever $R(n_0, \ldots, n_k)$ is true, **Q** proves $\varphi_R(\overline{n_0}, \ldots, \overline{n_k})$, and whenever $R(n_0, \ldots, n_k)$ is false, **Q** proves $\neg\varphi_R(\overline{n_0}, \ldots, \overline{n_k})$.

inc:req:rel:thm:representing-rels

**Theorem 3.27.** *A relation is representable in* **Q** *if and only if it is computable.*

*Proof.* For the forwards direction, suppose $R(x_0, \ldots, x_k)$ is represented by the formula $\varphi_R(x_0, \ldots, x_k)$. Here is an algorithm for computing $R$: on input $n_0$, $\ldots$, $n_k$, simultaneously search for a proof of $\varphi_R(\overline{n_0}, \ldots, \overline{n_k})$ and a proof of $\neg\varphi_R(\overline{n_0}, \ldots, \overline{n_k})$. By our hypothesis, the search is bound to find one or the other; if it is the first, report "yes," and otherwise, report "no."

In the other direction, suppose $R(x_0, \ldots, x_k)$ is computable. By definition, this means that the function $\chi_R(x_0, \ldots, x_k)$ is computable. By Theorem 3.2, $\chi_R$ is represented by a formula, say $\varphi_{\chi_R}(x_0, \ldots, x_k, y)$. Let $\varphi_R(x_0, \ldots, x_k)$ be the formula $\varphi_{\chi_R}(x_0, \ldots, x_k, \overline{1})$. Then for any $n_0$, $\ldots$, $n_k$, if $R(n_0, \ldots, n_k)$ is true, then $\chi_R(n_0, \ldots, n_k) = 1$, in which case **Q** proves $\varphi_{\chi_R}(\overline{n_0}, \ldots, \overline{n_k}, \overline{1})$, and so **Q** proves $\varphi_R(\overline{n_0}, \ldots, \overline{n_k})$. On the other hand, if $R(n_0, \ldots, n_k)$ is false, then $\chi_R(n_0, \ldots, n_k) = 0$. This means that **Q** proves

$$\forall y\, (\varphi_{\chi_R}(\overline{n_0}, \ldots, \overline{n_k}, y) \to y = \overline{0}).$$

Since **Q** proves $\overline{0} \neq \overline{1}$, **Q** proves $\neg\varphi_{\chi_R}(\overline{n_0}, \ldots, \overline{n_k}, \overline{1})$, and so it proves $\neg\varphi_R(\overline{n_0}, \ldots, \overline{n_k})$. □

**Problem 3.5.** Show that if $R$ is representable in **Q**, so is $\chi_R$.

## 3.10 Undecidability

inc:req:und:sec

We call a theory $\mathbf{T}$ *undecidable* if there is no computational procedure which, after finitely many steps and unfailingly, provides a correct answer to the question "does $\mathbf{T}$ prove $\varphi$?" for any sentence $\varphi$ in the language of $\mathbf{T}$. So $\mathbf{Q}$ would be decidable iff there were a computational procedure which decides, given a sentence $\varphi$ in the language of arithmetic, whether $\mathbf{Q} \vdash \varphi$ or not. We can make this more precise by asking: Is the relation $\mathrm{Prov}_{\mathbf{Q}}(y)$, which holds of $y$ iff $y$ is the Gödel number of a sentence provable in $\mathbf{Q}$, recursive? The answer is: no.

**Theorem 3.28.** $\mathbf{Q}$ *is undecidable, i.e., the relation*

$$\mathrm{Prov}_{\mathbf{Q}}(y) \Leftrightarrow \mathrm{Sent}(y) \wedge \exists x\, \mathrm{Prf}_{\mathbf{Q}}(x, y)$$

*is not recursive.*

*Proof.* Suppose it were. Then we could solve the halting problem as follows: Given $e$ and $n$, we know that $\varphi_e(n) \downarrow$ iff there is an $s$ such that $T(e, n, s)$, where $T$ is Kleene's predicate from **??**. Since $T$ is primitive recursive it is representable in $\mathbf{Q}$ by a formula $\psi_T$, that is, $\mathbf{Q} \vdash \psi_T(\overline{e}, \overline{n}, \overline{s})$ iff $T(e, n, s)$. If $\mathbf{Q} \vdash \psi_T(\overline{e}, \overline{n}, \overline{s})$ then also $\mathbf{Q} \vdash \exists y\, \psi_T(\overline{e}, \overline{n}, y)$. If no such $s$ exists, then $\mathbf{Q} \vdash \neg\psi_T(\overline{e}, \overline{n}, \overline{s})$ for every $s$. But $\mathbf{Q}$ is $\omega$-consistent, i.e., if $\mathbf{Q} \vdash \neg\varphi(\overline{n})$ for every $n \in \mathbb{N}$, then $\mathbf{Q} \nvdash \exists y\, \varphi(y)$. We know this because the axioms of $\mathbf{Q}$ are true in the standard model $\mathfrak{N}$. So, $\mathbf{Q} \nvdash \exists y\, \psi_T(\overline{e}, \overline{n}, y)$. In other words, $\mathbf{Q} \vdash \exists y\, \psi_T(\overline{e}, \overline{n}, y)$ iff there is an $s$ such that $T(e, n, s)$, i.e., iff $\varphi_e(n) \downarrow$. From $e$ and $n$ we can compute ${}^{\#}\exists y\, \psi_T(\overline{e}, \overline{n}, y)^{\#}$, let $g(e, n)$ be the primitive recursive function which does that. So

$$h(e, n) = \begin{cases} 1 & \text{if } \mathrm{Prov}_{\mathbf{Q}}(g(e, n)) \\ 0 & \text{otherwise.} \end{cases}$$

This would show that $h$ is recursive if $\mathrm{Prov}_{\mathbf{Q}}$ is. But $h$ is not recursive, by **??**, so $\mathrm{Prov}_{\mathbf{Q}}$ cannot be either. □

**Corollary 3.29.** *First-order logic is undecidable.*

*Proof.* If first-order logic were decidable, provability in $\mathbf{Q}$ would be as well, since $\mathbf{Q} \vdash \varphi$ iff $\vdash \omega \to \varphi$, where $\omega$ is the conjunction of the axioms of $\mathbf{Q}$. □

This chapter depends on material in the chapter on computability theory, but can be left out if that hasn't been covered. It's currently a basic conversion of Jeremy Avigad's notes, has not been revised, and is missing exercises.

# Chapter 4

# Theories and Computability

## 4.1 Introduction

inc:tcp:int:sec

This section should be rewritten.

We have the following:

1. A definition of what it means for a function to be representable in **Q** (Definition 3.1)
2. a definition of what it means for a relation to be representable in **Q** (Definition 3.26)
3. a theorem asserting that the representable functions of **Q** are exactly the computable ones (Theorem 3.2)
4. a theorem asserting that the representable relations of **Q** are exactly the computable ones Theorem 3.27)

A *theory* is a set of sentences that is deductively closed, that is, with the property that whenever $T$ proves $\varphi$ then $\varphi$ is in $T$. It is probably best to think of a theory as being a collection of sentences, together with all the things that these sentences imply. From now on, I will use **Q** to refer to the *theory* consisting of the set of sentences derivable from the eight axioms in section 3.1. Remember that we can code formula of **Q** as numbers; if $\varphi$ is such a formula, let $\ulcorner\varphi\urcorner$ denote the number coding $\varphi$. Modulo this coding, we can now ask whether various sets of formulas are computable or not.

## 4.2 Q is C.e.-Complete

inc:tcp:qce:sec

**Theorem 4.1.** **Q** *is c.e. but not decidable. In fact, it is a complete c.e. set.*

*Proof.* It is not hard to see that $\mathbf{Q}$ is c.e., since it is the set of (codes for) sentences $y$ such that there is a proof $x$ of $y$ in $\mathbf{Q}$:

$$Q = \{y : \exists x \, \mathrm{Prf}_{\mathbf{Q}}(x, y)\}.$$

But we know that $\mathrm{Prf}_{\mathbf{Q}}(x, y)$ is computable (in fact, primitive recursive), and any set that can be written in the above form is c.e.

Saying that it is a complete c.e. set is equivalent to saying that $K \leq_m Q$, where $K = \{x : \varphi_x(x) \downarrow\}$. So let us show that $K$ is reducible to $\mathbf{Q}$. Since Kleene's predicate $T(e, x, s)$ is primitive recursive, it is representable in $\mathbf{Q}$, say, by $\varphi_T$. Then for every $x$, we have

$$\begin{aligned} x \in K &\to \exists s \, T(x, x, s) \\ &\to \exists s \, (\mathbf{Q} \vdash \varphi_T(\overline{x}, \overline{x}, \overline{s})) \\ &\to \mathbf{Q} \vdash \exists s \, \varphi_T(\overline{x}, \overline{x}, s). \end{aligned}$$

Conversely, if $\mathbf{Q} \vdash \exists s \, \varphi_T(\overline{x}, \overline{x}, s)$, then, in fact, for some natural number $n$ the formula $\varphi_T(\overline{x}, \overline{x}, \overline{n})$ must be true. Now, if $T(x, x, n)$ were false, $\mathbf{Q}$ would prove $\neg\varphi_T(\overline{x}, \overline{x}, \overline{n})$, since $\varphi_T$ represents $T$. But then $\mathbf{Q}$ proves a false formula, which is a contradiction. So $T(x, x, n)$ must be true, which implies $\varphi_x(x) \downarrow$.

In short, we have that for every $x$, $x$ is in $K$ if and only if $\mathbf{Q}$ proves $\exists s \, T(\overline{x}, \overline{x}, s)$. So the function $f$ which takes $x$ to (a code for) the sentence $\exists s \, T(\overline{x}, \overline{x}, s)$ is a reduction of $K$ to $\mathbf{Q}$. □

## 4.3 $\omega$-Consistent Extensions of Q are Undecidable

inc:tcp:oqn: sec

The proof that $\mathbf{Q}$ is c.e.-complete relied on the fact that any sentence provable in $\mathbf{Q}$ is "true" of the natural numbers. The next definition and theorem strengthen this theorem, by pinpointing just those aspects of "truth" that were needed in the proof above. Don't dwell on this theorem too long, though, because we will soon strengthen it even further. We include it mainly for historical purposes: Gödel's original paper used the notion of $\omega$-consistency, but his result was strengthened by replacing $\omega$-consistency with ordinary consistency soon after.

explanation

inc:tcp:oqn: thm:oconsis-q

**Definition 4.2.** A theory $\mathbf{T}$ is $\omega$-consistent if the following holds: if $\exists x \, \varphi(x)$ is any sentence and $\mathbf{T}$ proves $\neg\varphi(\overline{0})$, $\neg\varphi(\overline{1})$, $\neg\varphi(\overline{2})$, ... then $\mathbf{T}$ does not prove $\exists x \, \varphi(x)$.

**Theorem 4.3.** *Let $\mathbf{T}$ be any $\omega$-consistent theory that includes $\mathbf{Q}$. Then $\mathbf{T}$ is not decidable.*

*Proof.* If $\mathbf{T}$ includes $\mathbf{Q}$, then $\mathbf{T}$ represents the computable functions and relations. We need only modify the previous proof. As above, if $x \in K$, then $\mathbf{T}$ proves $\exists s \, \varphi_T(\overline{x}, \overline{x}, s)$. Conversely, suppose $\mathbf{T}$ proves $\exists s \, \varphi_T(\overline{x}, \overline{x}, s)$. Then $x$ must be in $K$: otherwise, there is no halting computation of machine $x$ on input $x$; since $\varphi_T$ represents Kleene's $T$ relation, $\mathbf{T}$ proves $\neg\varphi_T(\overline{x}, \overline{x}, \overline{0})$, $\neg\varphi_T(\overline{x}, \overline{x}, \overline{1})$, ..., making $\mathbf{T}$ $\omega$-inconsistent. □

## 4.4 Consistent Extensions of Q are Undecidable

explanation

Remember that a theory is *consistent* if it does not prove both $\varphi$ and $\neg\varphi$ for any formula $\varphi$. Since anything follows from a contradiction, an inconsistent theory is trivial: every sentence is provable. Clearly, if a theory if $\omega$-consistent, then it is consistent. But being consistent is a weaker requirement (i.e., there are theories that are consistent but not $\omega$-consistent.). We can weaken the assumption in Definition 4.2 to simple consistency to obtain a stronger theorem.

inc:tcp:cqn:sec

**Lemma 4.4.** *There is no "universal computable relation." That is, there is no binary computable relation $R(x, y)$, with the following property: whenever $S(y)$ is a unary computable relation, there is some $k$ such that for every $y$, $S(y)$ is true if and only if $R(k, y)$ is true.*

*Proof.* Suppose $R(x, y)$ is a universal computable relation. Let $S(y)$ be the relation $\neg R(y, y)$. Since $S(y)$ is computable, for some $k$, $S(y)$ is equivalent to $R(k, y)$. But then we have that $S(k)$ is equivalent to both $R(k, k)$ and $\neg R(k, k)$, which is a contradiction. □

**Theorem 4.5.** *Let* **T** *be any consistent theory that includes* **Q**. *Then* **T** *is not decidable.*

*Proof.* Suppose **T** is a consistent, decidable extension of **Q**. We will obtain a contradiction by using **T** to define a universal computable relation.

Let $R(x, y)$ hold if and only if

> $x$ codes a formula $\theta(u)$, and **T** proves $\theta(\overline{y})$.

Since we are assuming that **T** is decidable, $R$ is computable. Let us show that $R$ is universal. If $S(y)$ is any computable relation, then it is representable in **Q** (and hence **T**) by a formula $\theta_S(u)$. Then for every $n$, we have

$$\begin{aligned} S(\overline{n}) &\to T \vdash \theta_S(\overline{n}) \\ &\to R(^{\ulcorner}\theta_S(u)^{\urcorner}, n) \end{aligned}$$

and

$$\begin{aligned} \neg S(\overline{n}) &\to T \vdash \neg\theta_S(\overline{n}) \\ &\to T \nvdash \theta_S(\overline{n}) \quad \text{(since } \mathbf{T} \text{ is consistent)} \\ &\to \neg R(^{\ulcorner}\theta_S(u)^{\urcorner}, n). \end{aligned}$$

That is, for every $y$, $S(y)$ is true if and only if $R(^{\ulcorner}\theta_S(u)^{\urcorner}, y)$ is. So $R$ is universal, and we have the contradiction we were looking for. □

Let "true arithmetic" be the theory $\{\varphi : \mathbb{N} \vDash \varphi\}$, that is, the set of sentences in the language of arithmetic that are true in the standard interpretation.

**Corollary 4.6.** *True arithmetic is not decidable.*

## 4.5 Axiomatizable Theories

inc:tcp:cax:sec

A theory $\mathbf{T}$ is said to be *axiomatizable* if it has a computable set of axioms $A$. (Saying that $A$ is a set of axioms for $\mathbf{T}$ means $T = \{\varphi : A \vdash \varphi\}$.) Any "reasonable" axiomatization of the natural numbers will have this property. In particular, any theory with a finite set of axioms is axiomatizable.

**Lemma 4.7.** *Suppose* $\mathbf{T}$ *is axiomatizable. Then* $\mathbf{T}$ *is computably enumerable.*

*Proof.* Suppose $A$ is a computable set of axioms for $\mathbf{T}$. To determine if $\varphi \in T$, just search for a proof of $\varphi$ from the axioms.

Put slightly differently, $\varphi$ is in $\mathbf{T}$ if and only if there is a finite list of axioms $\psi_1, \ldots, \psi_k$ in $A$ and a proof of $(\psi_1 \wedge \cdots \wedge \psi_k) \to \varphi$ in first-order logic. But we already know that any set with a definition of the form "there exists ... such that ..." is c.e., provided the second "..." is computable. □

## 4.6 Axiomatizable Complete Theories are Decidable

inc:tcp:cdc:sec

A theory is said to be *complete* if for every sentence $\varphi$, either $\varphi$ or $\neg\varphi$ is provable.

**Lemma 4.8.** *Suppose a theory* $\mathbf{T}$ *is complete and axiomatizable. Then* $\mathbf{T}$ *is decidable.*

*Proof.* Suppose $\mathbf{T}$ is complete and $A$ is a computable set of axioms. If $\mathbf{T}$ is inconsistent, it is clearly computable. (Algorithm: "just say yes.") So we can assume that $\mathbf{T}$ is also consistent.

To decide whether or not a sentence $\varphi$ is in $\mathbf{T}$, simultaneously search for a proof of $\varphi$ from $A$ and a proof of $\neg\varphi$. Since $\mathbf{T}$ is complete, you are bound to find one or another; and since $\mathbf{T}$ is consistent, if you find a proof of $\neg\varphi$, there is no proof of $\varphi$.

Put in different terms, we already know that $\mathbf{T}$ is c.e.; so by a theorem we proved before, it suffices to show that the complement of $\mathbf{T}$ is c.e. also. But a formula $\varphi$ is in $\bar{\mathbf{T}}$ if and only if $\neg\varphi$ is in $\mathbf{T}$; so $\bar{\mathbf{T}} \leq_m \mathbf{T}$. □

## 4.7 Q has no Complete, Consistent, Axiomatizable Extensions

inc:tcp:inc:sec
inc:tcp:inc:
thm:first-incompleteness

**Theorem 4.9.** *There is no complete, consistent, axiomatizable extension of* $\mathbf{Q}$.

*Proof.* We already know that there is no consistent, decidable extension of $\mathbf{Q}$. But if $\mathbf{T}$ is complete and axiomatized, then it is decidable. □

explanation

This theorems is not that far from Gödel's original 1931 formulation of the First Incompleteness Theorem. Aside from the more modern terminology, the key differences are this: Gödel has "$\omega$-consistent" instead of "consistent"; and he could not say "axiomatizable" in full generality, since the formal notion of computability was not in place yet. (The formal models of computability were developed over the following decade, including by Gödel, and in large part to be able to characterize the kinds of theories that are susceptible to the Gödel phenomenon.)

The theorem says you can't have it all, namely, completeness, consistency, and axiomatizability. If you give up any one of these, though, you can have the other two: $\mathbf{Q}$ is consistent and computably axiomatized, but not complete; the inconsistent theory is complete, and computably axiomatized (say, by $\{0 \neq 0\}$), but not consistent; and the set of true sentence of arithmetic is complete and consistent, but it is not computably axiomatized.

## 4.8 Sentences Provable and Refutable in Q are Computably Inseparable

inc:tcp:ins: sec

Let $\bar{\mathbf{Q}}$ be the set of sentences whose *negations* are provable in $\mathbf{Q}$, i.e., $\bar{\mathbf{Q}} = \{\varphi : \mathbf{Q} \vdash \neg\varphi\}$. Remember that disjoint sets $A$ and $B$ are said to be computably inseparable if there is no computable set $C$ such that $A \subseteq C$ and $B \subseteq \overline{C}$.

**Lemma 4.10.** $\mathbf{Q}$ *and* $\bar{\mathbf{Q}}$ *are computably inseparable.*

*Proof.* Suppose $C$ is a computable set such that $\mathbf{Q} \subseteq C$ and $\bar{\mathbf{Q}} \subseteq \overline{C}$. Let $R(x, y)$ be the relation

> $x$ codes a formula $\theta(u)$ and $\theta(\overline{y})$ is in $C$.

We will show that $R(x, y)$ is a universal computable relation, yielding a contradiction.

Suppose $S(y)$ is computable, represented by $\theta_S(u)$ in $\mathbf{Q}$. Then

$$\begin{aligned} S(\overline{n}) &\rightarrow \mathbf{Q} \vdash \theta_S(\overline{n}) \\ &\rightarrow \theta_S(\overline{n}) \in C \end{aligned}$$

and

$$\begin{aligned} \neg S(\overline{n}) &\rightarrow \mathbf{Q} \vdash \neg\theta_S(\overline{n}) \\ &\rightarrow \theta_S(\overline{n}) \in \bar{\mathbf{Q}} \\ &\rightarrow \theta_S(\overline{n}) \notin C \end{aligned}$$

So $S(y)$ is equivalent to $R(\#(\theta_S(\overline{u})), y)$. □

## 4.9 Theories Consistent with Q are Undecidable

inc:tcp:con:sec

The following theorem says that not only is **Q** undecidable, but, in fact, any theory that does not disagree with **Q** is undecidable.

**Theorem 4.11.** *Let* **T** *be any theory in the language of arithmetic that is consistent with* **Q** *(i.e.,* $\mathbf{T} \cup \mathbf{Q}$ *is consistent). Then* **T** *is undecidable.*

*Proof.* Remember that **Q** has a finite set of axioms, $Q_1, \ldots, Q_8$. We can even replace these by a single axiom, $\alpha = Q_1 \wedge \cdots \wedge Q_8$.

Suppose **T** is a decidable theory consistent with **Q**. Let

$$C = \{\varphi : \mathbf{T} \vdash \alpha \to \varphi\}.$$

We show that $C$ would be a computable separation of **Q** and $\bar{\mathbf{Q}}$, a contradiction. First, if $\varphi$ is in **Q**, then $\varphi$ is provable from the axioms of **Q**; by the deduction theorem, there is a proof of $\alpha \to \varphi$ in first-order logic. So $\varphi$ is in $C$.

On the other hand, if $\varphi$ is in $\bar{\mathbf{Q}}$, then there is a proof of $\alpha \to \neg\varphi$ in first-order logic. If **T** also proves $\alpha \to \varphi$, then **T** proves $\neg\alpha$, in which case $\mathbf{T} \cup \mathbf{Q}$ is inconsistent. But we are assuming $\mathbf{T} \cup \mathbf{Q}$ is consistent, so **T** does not prove $\alpha \to \varphi$, and so $\varphi$ is not in $C$.

We've shown that if $\varphi$ is in **Q**, then it is in $C$, and if $\varphi$ is in $\bar{\mathbf{Q}}$, then it is in $\overline{C}$. So $C$ is a computable separation, which is the contradiction we were looking for. □

This theorem is very powerful. For example, it implies:

**Corollary 4.12.** *First-order logic for the language of arithmetic (that is, the set* $\{\varphi : \varphi$ *is provable in first-order logic*$\}$*) is undecidable.*

*Proof.* First-order logic is the set of consequences of $\emptyset$, which is consistent with **Q**. □

## 4.10 Theories in which Q is Intepretable are Undecidable

inc:tcp:itp:sec

We can strengthen these results even more. Informally, an interpretation of a language $\mathcal{L}_1$ in another language $\mathcal{L}_2$ involves defining the universe, relation symbols, and function symbols of $\mathcal{L}_1$ with formulas in $\mathcal{L}_2$. Though we won't take the time to do this, one can make this definition precise.

**Theorem 4.13.** *Suppose* **T** *is a theory in a language in which one can interpret the language of arithmetic, in such a way that* **T** *is consistent with the interpretation of* **Q**. *Then* **T** *is undecidable. If* **T** *proves the interpretation of the axioms of* **Q**, *then no consistent extension of* **T** *is decidable.*

The proof is just a small modification of the proof of the last theorem; one could use a counterexample to get a separation of **Q** and $\bar{\mathbf{Q}}$. One can take **ZFC**, Zermelo-Fraenkel set theory with the axiom of choice, to be an axiomatic foundation that is powerful enough to carry out a good deal of ordinary mathematics. In **ZFC** one can define the natural numbers, and via this interpretation, the axioms of **Q** are true. So we have

**Corollary 4.14.** *There is no decidable extension of* **ZFC***.*

**Corollary 4.15.** *There is no complete, consistent, computably axiomatizable extension of* **ZFC***.*

The language of **ZFC** has only a single binary relation, $\in$. (In fact, you don't even need equality.) So we have

**Corollary 4.16.** *First-order logic for any language with a binary relation symbol is undecidable.*

This result extends to any language with two unary function symbols, since one can use these to simulate a binary relation symbol. The results just cited are tight: it turns out that first-order logic for a language with only *unary* relation symbols and at most one *unary* function symbol is decidable.

One more bit of trivia. We know that the set of sentences in the language $\mathsf{0}$, $'$, $+$, $\times$, $<$ true in the standard model is undecidable. In fact, one can define $<$ in terms of the other symbols, and then one can define $+$ in terms of $\times$ and $'$. So the set of true sentences in the language $\mathsf{0}$, $'$, $\times$ is undecidable. On the other hand, Presburger has shown that the set of sentences in the language $\mathsf{0}$, $'$, $+$ true in the language of arithmetic is decidable. The procedure is computationally infeasible, however.

# Chapter 5

# Incompleteness and Provability

## 5.1 Introduction

inc:inp:int: sec

Hilbert thought that a system of axioms for a mathematical structure, such as the natural numbers, is inadequate unless it allows one to derive all true statements about the structure. Combined with his later interest in formal systems of deduction, this suggests that he thought that we should guarantee that, say, the formal systems we are using to reason about the natural numbers is not only consistent, but also *complete*, i.e., every statement in its language is either provable or its negation is. Gödel's first incompleteness theorem shows that no such system of axioms exists: there is no complete, consistent, axiomatizable formal system for arithmetic. In fact, no "sufficiently strong," consistent, axiomatizable mathematical theory is complete.

A more important goal of Hilbert's, the centerpiece of his program for the justification of modern ("classical") mathematics, was to find finitary consistency proofs for formal systems representing classical reasoning. With regard to Hilbert's program, then, Gödel's second incompleteness theorem was a much bigger blow. The second incompleteness theorem can be stated in vague terms, like the first incompleteness theorem. Roughly speaking, it says that no sufficiently strong theory of arithmetic can prove its own consistency. We will have to take "sufficiently strong" to include a little bit more than **Q**.

The idea behind Gödel's original proof of the incompleteness theorem can be found in the Epimenides paradox. Epimenides, a Cretan, asserted that all Cretans are liars; a more direct form of the paradox is the assertion "this sentence is false." Essentially, by replacing truth with provability, Gödel was able to formalize a sentence which, in a roundabout way, asserts that it itself is not provable. If that sentence were provable, the theory would then be inconsistent. Assuming $\omega$-consistency—a property stronger than consistency—Gödel was able to show that this sentence is also not refutable from the system of axioms he was considering.

The first challenge is to understand how one can construct a sentence that refers to itself. For every formula $\varphi$ in the language of **Q**, let $\ulcorner\varphi\urcorner$ denote the

numeral corresponding to $^{\#}\varphi^{\#}$. Think about what this means: $\varphi$ is a formula in the language of $\mathbf{Q}$, $^{\#}\varphi^{\#}$ is a natural number, and $\ulcorner\varphi\urcorner$ is a *term* in the language of $\mathbf{Q}$. So every formula $\varphi$ in the language of $\mathbf{Q}$ has a *name*, $\ulcorner\varphi\urcorner$, which is a term in the language of $\mathbf{Q}$; this provides us with a conceptual framework in which formulas in the language of $\mathbf{Q}$ can "say" things about other formulas. The following lemma is known as the fixed-point lemma.

**Lemma 5.1.** *Let* $\mathbf{T}$ *be any theory extending* $\mathbf{Q}$*, and let* $\psi(x)$ *be any formula with only the variable* $x$ *free. Then there is a sentence* $\varphi$ *such that* $\mathbf{T}$ *proves* $\varphi \leftrightarrow \psi(\ulcorner\varphi\urcorner)$*.*

The lemma asserts that given any property $\psi(x)$, there is a sentence $\varphi$ that asserts "$\psi(x)$ is true of me."

How can we construct such a sentence? Consider the following version of the Epimenides paradox, due to Quine:

> "Yields falsehood when preceded by its quotation" yields falsehood when preceded by its quotation.

This sentence is not directly self-referential. It simply makes an assertion about the syntactic objects between quotes, and, in doing so, it is on par with sentences like

1. "Robert" is a nice name.
2. "I ran." is a short sentence.
3. "Has three words" has three words.

But what happens when one takes the phrase "yields falsehood when preceded by its quotation," and precedes it with a quoted version of itself? Then one has the original sentence! In short, the sentence asserts that it is false.

## 5.2 The Fixed-Point Lemma

explanation

The fixed-point lemma says that for any formula $\psi(x)$, there is a sentence $\varphi$ such that $\mathbf{T} \vdash \varphi \leftrightarrow \psi(\ulcorner\varphi\urcorner)$, provided $\mathbf{T}$ extends $\mathbf{Q}$. In the case of the liar sentence, we'd want $\varphi$ to be equivalent (provably in $\mathbf{T}$) to "$\ulcorner\varphi\urcorner$ is false," i.e., the statement that $^{\#}\varphi^{\#}$ is the Gödel number of a false sentence. To understand the idea of the proof, it will be useful to compare it with Quine's informal gloss of $\varphi$ as, "'yields a falsehood when preceded by its own quotation' yields a falsehood when preceded by its own quotation." The operation of taking an expression, and then forming a sentence by preceding this expression by its own quotation may be called *diagonalizing* the expression, and the result its diagonalization. So, the diagonalization of 'yields a falsehood when preceded by its own quotation' is "'yields a falsehood when preceded by its own quotation' yields a falsehood when preceded by its own quotation." Now note that Quine's liar sentence is not the diagonalization of 'yields a falsehood' but of

inc:inp:fix:sec

'yields a falsehood when preceded by its own quotation.' So the property being diagonalized to yield the liar sentence itself involves diagonalization!

In the language of arithmetic, we form quotations of a formula with one free variable by computing its Gödel numbers and then substituting the standard numeral for that Gödel number into the free variable. The diagonalization of $\alpha(x)$ is $\alpha(\overline{n})$, where $n = {}^{\#}\alpha(x)^{\#}$. (From now on, let's abbreviate $\overline{{}^{\#}\alpha(x)^{\#}}$ as $\ulcorner\alpha(x)\urcorner$.) So if $\psi(x)$ is "is a falsehood," then "yields a falsehood if preceded by its own quotation," would be "yields a falsehood when applied to the Gödel number of its diagonalization." If we had a symbol $\mathit{diag}$ for the function $\mathrm{diag}(n)$ which computes the Gödel number of the diagonalization of the formula with Gödel number $n$, we could write $\alpha(x)$ as $\psi(\mathit{diag}(x))$. And Quine's version of the liar sentence would then be the diagonalization of it, i.e., $\alpha(\ulcorner\alpha\urcorner)$ or $\psi(\mathit{diag}(\ulcorner\psi(\mathit{diag}(x))\urcorner))$. Of course, $\psi(x)$ could now be any other property, and the same construction would work. For the incompleteness theorem, we'll take $\psi(x)$ to be "$x$ is unprovable in $\mathbf{T}$." Then $\alpha(x)$ would be "yields a sentence unprovable in $\mathbf{T}$ when applied to the Gödel number of its diagonalization."

To formalize this in $\mathbf{T}$, we have to find a way to formalize diag. The function $\mathrm{diag}(n)$ is computable, in fact, it is primitive recursive: if $n$ is the Gödel number of a formula $\alpha(x)$, $\mathrm{diag}(n)$ returns the Gödel number of $\alpha(\ulcorner\alpha(x)\urcorner)$. (Recall, $\ulcorner\alpha(x)\urcorner$ is the standard numeral of the Gödel number of $\alpha(x)$, i.e., $\overline{{}^{\#}\alpha(x)^{\#}}$). If $\mathit{diag}$ were a function symbol in $\mathbf{T}$ repqresenting the function diag, we could take $\varphi$ to be the formula $\psi(\mathit{diag}(\ulcorner\psi(\mathit{diag}(x))\urcorner))$. Notice that

$$\begin{aligned}\mathrm{diag}({}^{\#}\psi(\mathit{diag}(x))^{\#}) &= {}^{\#}\psi(\mathit{diag}(\ulcorner\psi(\mathit{diag}(x))\urcorner)^{\#}\\ &= {}^{\#}\varphi^{\#}.\end{aligned}$$

Assuming $\mathbf{T}$ can prove

$$\mathit{diag}(\ulcorner\psi(\mathit{diag}(x))\urcorner) = \ulcorner\varphi\urcorner,$$

it can prove $\psi(\mathit{diag}(\ulcorner\psi(\mathit{diag}(x))\urcorner)) \leftrightarrow \psi(\ulcorner\varphi\urcorner)$. But the left hand side is, by definition, $\varphi$.

Of course, $\mathit{diag}$ will in general not be a function symbol of $\mathbf{T}$, and certainly is not one of $\mathbf{Q}$. But, since diag is computable, it is *representable* in $\mathbf{Q}$ by some formula $\theta_{\mathrm{diag}}(x, y)$. So instead of writing $\psi(\mathit{diag}(x))$ we can write $\exists y\,(\theta_{\mathrm{diag}}(x, y) \wedge \psi(y))$. Otherwise, the proof sketched above goes through, and in fact, it goes through already in $\mathbf{Q}$.

inc:inp:fix: lem:fixed-point

**Lemma 5.2.** *Let $\psi(x)$ be any formula with one free variable $x$. Then there is a sentence $\varphi$ such that $\mathbf{Q} \vdash \varphi \leftrightarrow \psi(\ulcorner\varphi\urcorner)$.*

*Proof.* Given $\psi(x)$, let $\alpha(x)$ be the formula $\exists y\,(\theta_{\mathrm{diag}}(x, y) \wedge \psi(y))$ and let $\varphi$ be its diagonalization, i.e., the formula $\alpha(\ulcorner\alpha(x)\urcorner)$.

Since $\theta_{\mathrm{diag}}$ represents diag, and $\mathrm{diag}({}^{\#}\alpha(x)^{\#}) = {}^{\#}\varphi^{\#}$, $\mathbf{Q}$ can prove

inc:inp:fix: repdiag1
$$\theta_{\mathrm{diag}}(\ulcorner\alpha(x)\urcorner, \ulcorner\varphi\urcorner) \tag{5.1}$$

inc:inp:fix: repdiag2
$$\forall y\,(\theta_{\mathrm{diag}}(\ulcorner\alpha(x)\urcorner, y) \to y = \ulcorner\varphi\urcorner). \tag{5.2}$$

Now we show that $\mathbf{Q} \vdash \varphi \leftrightarrow \psi(\ulcorner\varphi\urcorner)$. We argue informally, using just logic and facts provable in $\mathbf{Q}$.

First, suppose $\varphi$, i.e., $\alpha(\ulcorner\alpha(x)\urcorner)$. Going back to the definition of $\alpha(x)$, we see that $\alpha(\ulcorner\alpha(x)\urcorner)$ just is

$$\exists y\,(\theta_{\text{diag}}(\ulcorner\alpha(x)\urcorner, y) \wedge \psi(y)).$$

Consider such a $y$. Since $\theta_{\text{diag}}(\ulcorner\alpha(x)\urcorner, y)$, by eq. (5.2), $y = \ulcorner\varphi\urcorner$. So, from $\psi(y)$ we have $\psi(\ulcorner\varphi\urcorner)$.

Now suppose $\psi(\ulcorner\varphi\urcorner)$. By eq. (5.1), we have $\theta_{\text{diag}}(\ulcorner\alpha(x)\urcorner, \ulcorner\varphi\urcorner) \wedge \psi(\ulcorner\varphi\urcorner)$. It follows that $\exists y\,(\theta_{\text{diag}}(\ulcorner\alpha(x)\urcorner, y) \wedge \psi(y))$. But that's just $\alpha(\ulcorner\alpha\urcorner)$, i.e., $\varphi$. □

digression You should compare this to the proof of the fixed-point lemma in computability theory. The difference is that here we want to define a *statement* in terms of itself, whereas there we wanted to define a *function* in terms of itself; this difference aside, it is really the same idea.

## 5.3 The First Incompleteness Theorem

inc:inp:1in: sec We can now describe Gödel's original proof of the first incompleteness theorem. Let $\mathbf{T}$ be any computably axiomatized theory in a language extending the language of arithmetic, such that $\mathbf{T}$ includes the axioms of $\mathbf{Q}$. This means that, in particular, $\mathbf{T}$ represents computable functions and relations.

We have argued that, given a reasonable coding of formulas and proofs as numbers, the relation $\text{Prf}_T(x, y)$ is computable, where $\text{Prf}_T(x, y)$ holds if and only if $x$ is the Gödel number of a derivation of the formula with Gödel number $y$ in $\mathbf{T}$. In fact, for the particular theory that Gödel had in mind, Gödel was able to show that this relation is primitive recursive, using the list of 45 functions and relations in his paper. The 45th relation, $xBy$, is just $\text{Prf}_T(x, y)$ for his particular choice of $\mathbf{T}$. Remember that where Gödel uses the word "recursive" in his paper, we would now use the phrase "primitive recursive."

Since $\text{Prf}_T(x, y)$ is computable, it is representable in $\mathbf{T}$. We will use $\mathsf{Prf}_T(x, y)$ to refer to the formula that represents it. Let $\mathsf{Prov}_T(y)$ be the formula $\exists x\, \mathsf{Prf}_T(x, y)$. This describes the 46th relation, $\text{Bew}(y)$, on Gödel's list. As Gödel notes, this is the only relation that "cannot be asserted to be recursive." What he probably meant is this: from the definition, it is not clear that it is computable; and later developments, in fact, show that it isn't.

**Definition 5.3.** A theory $\mathbf{T}$ is $\omega$-consistent if the following holds: if $\exists x\, \varphi(x)$ is any sentence and $\mathbf{T}$ proves $\neg\varphi(\overline{0})$, $\neg\varphi(\overline{1})$, $\neg\varphi(\overline{2})$, ... then $\mathbf{T}$ does not prove $\exists x\, \varphi(x)$. inc:inp:1in: thm:oconsis-q

We can now prove the following.

**Theorem 5.4.** *Let $\mathbf{T}$ be any $\omega$-consistent, axiomatizable theory extending $\mathbf{Q}$. Then $\mathbf{T}$ is not complete.* inc:inp:1in: thm:first-incompleteness

*Proof.* Let $\mathbf{T}$ be an axiomatizable theory containing $\mathbf{Q}$ Then $\mathrm{Prf}_T(x, y)$ is decidable, hence representable in $\mathbf{Q}$ by a formula $\mathsf{Prf}_T(x, y)$. Let $\mathsf{Prov}_T(y)$ be the formula we described above. By the fixed-point lemma, there is a formula $\gamma_{\mathbf{T}}$ such that $\mathbf{Q}$ (and hence $\mathbf{T}$) proves

inc:inp:1in: eqn:qpf

$$\gamma_{\mathbf{T}} \leftrightarrow \neg\mathsf{Prov}_T(\ulcorner\gamma_{\mathbf{T}}\urcorner). \tag{5.3}$$

Note that $\varphi$ says, in essence, "$\varphi$ is not provable."

We claim that

1. If $\mathbf{T}$ is consistent, $\mathbf{T}$ doesn't prove $\gamma_{\mathbf{T}}$
2. If $\mathbf{T}$ is $\omega$-consistent, $\mathbf{T}$ doesn't prove $\neg\gamma_{\mathbf{T}}$.

This means that if $\mathbf{T}$ is $\omega$-consistent, it is incomplete, since it proves neither $\gamma_{\mathbf{T}}$ nor $\neg\gamma_{\mathbf{T}}$. Let us take each claim in turn.

Suppose $\mathbf{T}$ proves $\gamma_{\mathbf{T}}$. Then there *is* a derivation, and so, for some number $m$, the relation $\mathrm{Prf}_T(m, {}^{\#}\gamma_{\mathbf{T}}{}^{\#})$ holds. But then $\mathbf{Q}$ proves the sentence $\mathsf{Prf}_T(\overline{m}, \ulcorner\gamma_{\mathbf{T}}\urcorner)$. So $\mathbf{Q}$ proves $\exists x\, \mathsf{Prf}_T(x, \ulcorner\gamma_{\mathbf{T}}\urcorner)$, which is, by definition, $\mathsf{Prov}_T(\ulcorner\gamma_{\mathbf{T}}\urcorner)$. By eq. (5.3), $\mathbf{Q}$ proves $\neg\gamma_{\mathbf{T}}$, and since $\mathbf{T}$ extends $\mathbf{Q}$, so does $\mathbf{T}$. We have shown that if $\mathbf{T}$ proves $\gamma_{\mathbf{T}}$, then it also proves $\neg\gamma_{\mathbf{T}}$, and hence it would be inconsistent.

For the second claim, let us show that if $\mathbf{T}$ proves $\neg\gamma_{\mathbf{T}}$, then it is $\omega$-inconsistent. Suppose $\mathbf{T}$ proves $\neg\gamma_{\mathbf{T}}$. If $\mathbf{T}$ is inconsistent, it is $\omega$-inconsistent, and we are done. Otherwise, $\mathbf{T}$ is consistent, so it does not prove $\gamma_{\mathbf{T}}$. Since there is no proof of $\gamma_{\mathbf{T}}$ in $\mathbf{T}$, $\mathbf{Q}$ proves

$$\neg\mathsf{Prf}_T(\overline{0}, \ulcorner\gamma_{\mathbf{T}}\urcorner), \neg\mathsf{Prf}_T(\overline{1}, \ulcorner\gamma_{\mathbf{T}}\urcorner), \neg\mathsf{Prf}_T(\overline{2}, \ulcorner\gamma_{\mathbf{T}}\urcorner), \ldots$$

and so does $\mathbf{T}$. On the other hand, by eq. (5.3), $\neg\gamma_{\mathbf{T}}$ is equivalent to $\exists x\, \mathsf{Prf}_T(x, \ulcorner\gamma_{\mathbf{T}}\urcorner)$. So $\mathbf{T}$ is $\omega$-inconsistent. □

## 5.4 Rosser's Theorem

inc:inp:ros: sec

Can we modify Gödel's proof to get a stronger result, replacing "$\omega$-consistent" with simply "consistent"? The answer is "yes," using a trick discovered by Rosser. Rosser's trick is to use a "modified" provability predicate $\mathsf{RProv}_T(y)$ instead of $\mathsf{Prov}_T(y)$.

inc:inp:ros: thm:rosser

**Theorem 5.5.** *Let $\mathbf{T}$ be any consistent, axiomatizable theory extending $\mathbf{Q}$. Then $\mathbf{T}$ is not complete.*

*Proof.* Recall that $\mathsf{Prov}_T(y)$ is defined as $\exists x\, \mathsf{Prf}_T(x, y)$, where $\mathsf{Prf}_T(x, y)$ represents the decidable relation which holds iff $x$ is the Gödel number of a derivation of the sentence with Gödel number $y$. The relation that holds between $x$ and $y$ if $x$ is the Gödel number of a *refutation* of the sentence with Gödel number $y$ is also decidable. Let $\mathrm{not}(x)$ be the primitive recursive function which does the following: if $x$ is the code of a formula $\varphi$, $\mathrm{not}(x)$ is a code of $\neg\varphi$. Then $\mathrm{Ref}_T(x, y)$ holds iff $\mathrm{Prf}_T(x, \mathrm{not}(y))$. Let $\mathsf{Ref}_T(x, y)$ represent it. Then,

if $\mathbf{T} \vdash \neg\varphi$ and $\delta$ is a corresponding derivation, $\mathbf{Q} \vdash \mathsf{Ref}_T(\ulcorner\delta\urcorner, \ulcorner\varphi\urcorner)$. We define $\mathsf{RProv}_T(y)$ as

$$\exists x\,(\mathsf{Prf}_T(x, y) \wedge \forall z\,(z < x \to \neg\mathsf{Ref}_T(z, y))).$$

Roughly, $\mathsf{RProv}_T(y)$ says "there is a proof of $y$ in $\mathbf{T}$, and there is no shorter refutation of $y$." (You might find it convenient to read $\mathsf{RProv}_T(y)$ as "$y$ is shmovable.") Assuming $\mathbf{T}$ is consistent, $\mathsf{RProv}_T(y)$ is true of the same numbers as $\mathsf{Prov}_T(y)$; but from the point of view of *provability* in $\mathbf{T}$ (and we now know that there is a difference between truth and provability!) the two have different properties. (If $\mathbf{T}$ is *in*consistent, then the two do *not* hold of the same numbers!)

By the fixed-point lemma, there is a formula $\rho_{\mathbf{T}}$ such that

$$\mathbf{Q} \vdash \rho_{\mathbf{T}} \leftrightarrow \neg\mathsf{RProv}_T(\ulcorner\rho_{\mathbf{T}}\urcorner). \tag{5.4}$$

inc:inp:ros:RT

In contrast to the proof of Theorem 5.4, here we claim that if $\mathbf{T}$ is consistent, $\mathbf{T}$ doesn't prove $\rho_{\mathbf{T}}$, and $\mathbf{T}$ also doesn't prove $\neg\rho_{\mathbf{T}}$. (In other words, we don't need the assumption of $\omega$-consistency.)

First, let's show that $\mathbf{T} \nvdash \rho_T$. Suppose it did, so there is a derivation of $\rho_T$ from $T$; let $n$ be its Gödel number. Then $\mathbf{Q} \vdash \mathsf{Prf}_T(\overline{n}, \ulcorner\rho_T\urcorner)$, since $\mathsf{Prf}_T$ represents $\mathrm{Prf}_T$ in $\mathbf{Q}$. Also, for each $k < n$, $k$ is not the Gödel number of $\neg\rho_T$, since $\mathbf{T}$ is consistent. So for each $k < n$, $\mathbf{Q} \vdash \neg\mathsf{Ref}_T(\overline{k}, \ulcorner\rho_T\urcorner)$. By Lemma 3.22(2), $\mathbf{Q} \vdash \forall z\,(z < \overline{n} \to \neg\mathsf{Ref}_T(z, \ulcorner\rho_T\urcorner))$. Thus,

$$\mathbf{Q} \vdash \exists x\,(\mathsf{Prf}_T(x, \ulcorner\rho_T\urcorner) \wedge \forall z\,(z < x \to \neg\mathsf{Ref}_T(z, \ulcorner\rho_T\urcorner))),$$

but that's just $\mathsf{RProv}_T(\ulcorner\rho_T\urcorner)$. By eq. (5.4), $\mathbf{Q} \vdash \neg\rho_T$. Since $\mathbf{T}$ extends $\mathbf{Q}$, also $\mathbf{T} \vdash \neg\rho_T$. We've assumed that $\mathbf{T} \vdash \rho_T$, so $\mathbf{T}$ would be inconsistent, contrary to the assumption of the theorem.

Now, let's show that $\mathbf{T} \nvdash \neg\rho_T$. Again, suppose it did, and suppose $n$ is the Gödel number of a derivation of $\neg\rho_T$. Then $\mathrm{Ref}_T(n, {}^{\ulcorner}\rho_T{}^{\#})$ holds, and since $\mathsf{Ref}_T$ represents $\mathrm{Ref}_T$ in $\mathbf{Q}$, $\mathbf{Q} \vdash \mathsf{Ref}_T(\overline{n}, \ulcorner\rho_T\urcorner)$. We'll again show that $\mathbf{T}$ would then be inconsistent because it would also prove $\rho_T$. Since $\mathbf{Q} \vdash \rho_T \leftrightarrow \neg\mathsf{RProv}_T(\ulcorner\rho_T\urcorner)$, and since $\mathbf{T}$ extends $\mathbf{Q}$, it suffices to show that $\mathbf{Q} \vdash \neg\mathsf{RProv}_T(\ulcorner\rho_T\urcorner)$. The sentence $\neg\mathsf{RProv}_T(\ulcorner\rho_T\urcorner)$, i.e.,

$$\neg\exists x\,(\mathsf{Prf}_T(x, \ulcorner\rho_T\urcorner) \wedge \forall z\,(z < x \to \neg\mathsf{Ref}_T(z, \ulcorner\rho_T\urcorner)))$$

is logically equivalent to

$$\forall x\,(\mathsf{Prf}_T(x, \ulcorner\rho_T\urcorner) \to \exists z\,(z < x \wedge \mathsf{Ref}_T(z, \ulcorner\rho_T\urcorner)))$$

We argue informally using logic, making use of facts about what $\mathbf{Q}$ proves. Suppose $x$ is arbitrary and $\mathsf{Prf}_T(x, \ulcorner\rho_T\urcorner)$. We already know that $\mathbf{T} \nvdash \rho_T$, and so for every $k$, $\mathbf{Q} \vdash \neg\mathsf{Prf}_T(\overline{k}, \ulcorner\rho_T\urcorner)$. Thus, for every $k$ it follows that $x \neq \overline{k}$. In particular, we have (a) that $x \neq \overline{n}$. We also have $\neg(x = \overline{0} \vee x =$

$\overline{1} \vee \cdots \vee x = \overline{n-1}$) and so by Lemma 3.22(2), (b) $\neg(x < \overline{n})$. By Lemma 3.23, $\overline{n} < x$. Since $\mathbf{Q} \vdash \mathsf{Ref}_T(\overline{n}, \ulcorner \rho_T \urcorner)$, we have $\overline{n} < x \wedge \mathsf{Ref}_T(\overline{n}, \ulcorner \rho_T \urcorner)$, and from that $\exists z\,(z < x \wedge \mathsf{Ref}_T(z, \ulcorner \rho_T \urcorner))$. Since $x$ was arbitrary we get

$$\forall x\,(\mathsf{Prf}_T(x, \ulcorner \rho_T \urcorner) \to \exists z\,(z < x \wedge \mathsf{Ref}_T(z, \ulcorner \rho_T \urcorner)))$$

as required. □

## 5.5 Comparison with Gödel's Original Paper

inc:inp:gop: sec

It is worthwhile to spend some time with Gödel's 1931 paper. The introduction sketches the ideas we have just discussed. Even if you just skim through the paper, it is easy to see what is going on at each stage: first Gödel describes the formal system $P$ (syntax, axioms, proof rules); then he defines the primitive recursive functions and relations; then he shows that $xBy$ is primitive recursive, and argues that the primitive recursive functions and relations are represented in **P**. He then goes on to prove the incompleteness theorem, as above. In section 3, he shows that one can take the unprovable assertion to be a sentence in the language of arithmetic. This is the origin of the $\beta$-lemma, which is what we also used to handle sequences in showing that the recursive functions are representable in **Q**. Gödel doesn't go so far to isolate a minimal set of axioms that suffice, but we now know that **Q** will do the trick. Finally, in Section 4, he sketches a proof of the second incompleteness theorem.

## 5.6 The Provability Conditions for PA

inc:inp:prc: sec

Peano arithmetic, or **PA**, is the theory extending **Q** with induction axioms for all formulas. In other words, one adds to **Q** axioms of the form

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

for every formula $\varphi$. Notice that this is really a *schema*, which is to say, infinitely many axioms (and it turns out that **PA** is *not* finitely axiomatizable). But since one can effectively determine whether or not a string of symbols is an instance of an induction axiom, the set of axioms for **PA** is computable. **PA** is a much more robust theory than **Q**. For example, one can easily prove that addition and multiplication are commutative, using induction in the usual way. In fact, most finitary number-theoretic and combinatorial arguments can be carried out in **PA**.

Since **PA** is computably axiomatized, the provability predicate $\mathrm{Prf}_{\mathbf{PA}}(x, y)$ is computable and hence represented in **Q** (and so, in **PA**). As before, I will take $\mathsf{Prf}_{\mathbf{PA}}(x, y)$ to denote the formula representing the relation. Let $\mathsf{Prov}_{\mathbf{PA}}(y)$ be the formula $\exists x\, \mathrm{Prf}_{\mathbf{PA}}(x, y)$, which, intuitively says, "$y$ is provable from the axioms of **PA**." The reason we need a little bit more than the axioms of **Q** is we need to know that the theory we are using is strong enough to prove a few basic facts about this provability predicate. In fact, what we need are the following facts:

P1. If $\mathbf{PA} \vdash \varphi$, then $\mathbf{PA} \vdash \mathsf{Prov}_{\mathbf{PA}}(\ulcorner\varphi\urcorner)$

P2. For all formulas $\varphi$ and $\psi$,

$$\mathbf{PA} \vdash \mathsf{Prov}_{\mathbf{PA}}(\ulcorner\varphi \to \psi\urcorner) \to (\mathsf{Prov}_{\mathbf{PA}}(\ulcorner\varphi\urcorner) \to \mathsf{Prov}_{\mathbf{PA}}(\ulcorner\psi\urcorner))$$

P3. For every formula $\varphi$,

$$\mathbf{PA} \vdash \mathsf{Prov}_{\mathbf{PA}}(\ulcorner\varphi\urcorner) \to \mathsf{Prov}_{\mathbf{PA}}(\ulcorner\mathsf{Prov}_{\mathbf{PA}}(\ulcorner\varphi\urcorner)\urcorner).$$

The only way to verify that these three properties hold is to describe the formula $\mathsf{Prov}_{\mathbf{PA}}(y)$ carefully and use the axioms of **PA** to describe the relevant formal proofs. Conditions (1) and (2) are easy; it is really condition (3) that requires work. (Think about what kind of work it entails. . . ) Carrying out the details would be tedious and uninteresting, so here we will ask you to take it on faith that **PA** has the three properties listed above. A reasonable choice of $\mathsf{Prov}_{\mathbf{PA}}(y)$ will also satisfy

P4. If $\mathbf{PA} \vdash \mathsf{Prov}_{\mathbf{PA}}(\ulcorner\varphi\urcorner)$, then $\mathbf{PA} \vdash \varphi$.

But we will not need this fact.

digression Incidentally, Gödel was lazy in the same way we are being now. At the end of the 1931 paper, he sketches the proof of the second incompleteness theorem, and promises the details in a later paper. He never got around to it; since everyone who understood the argument believed that it could be carried out (he did not need to fill in the details.)

## 5.7 The Second Incompleteness Theorem

How can we express the assertion that **PA** doesn't prove its own consistency? Saying **PA** is inconsistent amounts to saying that **PA** proves $0 = 1$. So we can take $\mathsf{Con}_{\mathbf{PA}}$ to be the formula $\neg\mathsf{Prov}_{\mathbf{PA}}(\ulcorner 0 = 1\urcorner)$, and then the following theorem does the job: inc:inp:2in:sec

**Theorem 5.6.** *Assuming* **PA** *is consistent, then* **PA** *does not prove* $\mathsf{Con}_{\mathbf{PA}}$. inc:inp:2in:thm:second-incompleteness

It is important to note that the theorem depends on the particular representation of $\mathsf{Con}_{\mathbf{PA}}$ (i.e., the particular representation of $\mathsf{Prov}_{\mathbf{PA}}(y)$). All we will use is that the representation of $\mathsf{Prov}_{\mathbf{PA}}(y)$ has the three properties above, so the theorem generalizes to any theory with a provability predicate having these properties.

It is informative to read Gödel's sketch of an argument, since the theorem follows like a good punch line. It goes like this. Let $\gamma_{\mathbf{PA}}$ be the Gödel sentence that we constructed in the proof of Theorem 5.4. We have shown "If **PA** is consistent, then **PA** does not prove $\gamma_{\mathbf{PA}}$." If we formalize this *in* **PA**, we have a proof of

$$\mathsf{Con}_{\mathbf{PA}} \to \neg\mathsf{Prov}_{\mathbf{PA}}(\ulcorner\gamma_{\mathbf{PA}}\urcorner).$$

Now suppose **PA** proves $\mathsf{Con}_{\mathbf{PA}}$. Then it proves $\neg\mathsf{Prov}_{\mathbf{PA}}(\ulcorner\gamma_{\mathbf{PA}}\urcorner)$. But since $\gamma_{\mathbf{PA}}$ is a Gödel sentence, this is equivalent to $\gamma_{\mathbf{PA}}$. So **PA** proves $\gamma_{\mathbf{PA}}$.

But: we know that if **PA** is consistent, it doesn't prove $\gamma_{\mathbf{PA}}$! So if **PA** is consistent, it can't prove $\mathsf{Con}_{\mathbf{PA}}$.

To make the argument more precise, we will let $\gamma_{\mathbf{PA}}$ be the Gödel sentence for **PA** and use the provability conditions (1)–(3) above to show that **PA** proves $\mathsf{Con}_{\mathbf{PA}} \to \gamma_{\mathbf{PA}}$. This will show that **PA** doesn't prove $\mathsf{Con}_{\mathbf{PA}}$. Here is a sketch of the proof, in **PA**. (For simplicity, we drop the **PA** subscripts.)

inc:inp:2in: G2-1
$$\gamma \leftrightarrow \neg\mathsf{Prov}(\ulcorner\gamma\urcorner) \tag{5.5}$$
$\gamma$ is a Gödel sentence

inc:inp:2in: G2-2
$$\gamma \to \neg\mathsf{Prov}(\ulcorner\gamma\urcorner) \tag{5.6}$$
from eq. (5.5)

inc:inp:2in: G2-3
$$\gamma \to (\mathsf{Prov}(\ulcorner\gamma\urcorner) \to \bot) \tag{5.7}$$
from eq. (5.6) by logic

inc:inp:2in: G2-4
$$\mathsf{Prov}(\ulcorner\gamma \to (\mathsf{Prov}(\ulcorner\gamma\urcorner) \to \bot)\urcorner) \tag{5.8}$$
by from eq. (5.7) by condition P1

inc:inp:2in: G2-5
$$\mathsf{Prov}(\ulcorner\gamma\urcorner) \to \mathsf{Prov}(\ulcorner(\mathsf{Prov}(\ulcorner\gamma\urcorner) \to \bot)\urcorner) \tag{5.9}$$
from eq. (5.8) by condition P2

inc:inp:2in: G2-6
$$\mathsf{Prov}(\ulcorner\gamma\urcorner) \to (\mathsf{Prov}(\ulcorner\mathsf{Prov}(\ulcorner\gamma\urcorner)\urcorner) \to \mathsf{Prov}(\ulcorner\bot\urcorner)) \tag{5.10}$$
from eq. (5.9) by condition P2 and logic

inc:inp:2in: G2-7
$$\mathsf{Prov}(\ulcorner\gamma\urcorner) \to \mathsf{Prov}(\ulcorner\mathsf{Prov}(\ulcorner\gamma\urcorner)\urcorner) \tag{5.11}$$
by P3

inc:inp:2in: G2-8
$$\mathsf{Prov}(\ulcorner\gamma\urcorner) \to \mathsf{Prov}(\ulcorner\bot\urcorner) \tag{5.12}$$
from eq. (5.10) and eq. (5.11) by logic

inc:inp:2in: G2-9
$$\mathsf{Con} \to \neg\mathsf{Prov}(\ulcorner\gamma\urcorner) \tag{5.13}$$
contraposition of eq. (5.12) and $\mathsf{Con} \equiv \neg\mathsf{Prov}(\ulcorner\bot\urcorner)$

$\mathsf{Con} \to \gamma$

from eq. (5.5) and eq. (5.13) by logic

The use of logic in the above just elementary facts from propositional logic, e.g., eq. (5.7) uses $\vdash \neg\varphi \leftrightarrow (\varphi \to \bot)$ and eq. (5.12) uses $\varphi \to (\psi \to \chi), \varphi \to \psi \vdash \varphi \to \chi$. The use of condition P2 in eq. (5.9) and eq. (5.10) relies on instances of P2, $\mathsf{Prov}(\ulcorner\varphi \to \psi\urcorner) \to (\mathsf{Prov}(\ulcorner\varphi\urcorner) \to \mathsf{Prov}(\ulcorner\psi\urcorner))$. In the first one, $\varphi \equiv \gamma$ and $\psi \equiv \mathsf{Prov}(\ulcorner\gamma\urcorner) \to \bot$; in the second, $\varphi \equiv \mathsf{Prov}(\ulcorner G\urcorner)$ and $\psi \equiv \bot$.

The more abstract version of the incompleteness theorem is as follows:

inc:inp:2in: thm:second-incompleteness-gen
**Theorem 5.7.** *Let* **T** *be any axiomatized theory extending* **Q** *and let* $\mathsf{Prov}_T(y)$ *be any formula satisfying provability conditions P1–P3 for* **T**. *Then if* **T** *is consistent, then* **T** *does not prove* $\mathsf{Con}_T$.

**Problem 5.1.** Show that **PA** proves $\gamma_{\mathbf{PA}} \to \mathsf{Con}_{\mathbf{PA}}$.

digression

The moral of the story is that no "reasonable" consistent theory for mathematics can prove its own consistency. Suppose $\mathbf{T}$ is a theory of mathematics that includes $\mathbf{Q}$ and Hilbert's "finitary" reasoning (whatever that may be). Then, the whole of $\mathbf{T}$ cannot prove the consistency of $\mathbf{T}$, and so, a fortiori, the finitary fragment can't prove the consistency of $\mathbf{T}$ either. In that sense, there cannot be a finitary consistency proof for "all of mathematics."

There is some leeway in interpreting the term "finitary," and Gödel, in the 1931 paper, grants the possibility that something we may consider "finitary" may lie outside the kinds of mathematics Hilbert wanted to formalize. But Gödel was being charitable; today, it is hard to see how we might find something that can reasonably be called finitary but is not formalizable in, say, **ZFC**.

## 5.8 Löb's Theorem

The Gödel sentence for a theory $\mathbf{T}$ is a fixed point of $\neg\mathsf{Prov}_T(x)$, i.e., a sentence $\gamma$ such that inc:inp:lob:sec

$$\mathbf{T} \vdash \neg\mathsf{Prov}_T(\ulcorner\gamma\urcorner) \leftrightarrow \gamma.$$

It is not provable, because if $\mathbf{T} \vdash \gamma$, (a) by provability condition (1), $\mathbf{T} \vdash \mathsf{Prov}_T(\ulcorner\gamma\urcorner)$, and (b) $\mathbf{T} \vdash \gamma$ together with $\mathbf{T} \vdash \neg\mathsf{Prov}_T(\ulcorner\gamma\urcorner) \leftrightarrow \gamma$ gives $\mathbf{T} \vdash \neg\mathsf{Prov}_T(\ulcorner\gamma\urcorner)$, and so $\mathbf{T}$ would be inconsistent. Now it is natural to ask about the status of a fixed point of $\mathsf{Prov}_T(x)$, i.e., a sentence $\delta$ such that

$$\mathbf{T} \vdash \mathsf{Prov}_T(\ulcorner\delta\urcorner) \leftrightarrow \delta.$$

If it were provable, $\mathbf{T} \vdash \mathsf{Prov}_T(\ulcorner\delta\urcorner)$ by condition (1), but the same conclusion follows if we apply modus ponens to the equivalence above. Hence, we don't get that $\mathbf{T}$ is inconsistent, at least not by the same argument as in the case of the Gödel sentence. This of course does not show that $\mathbf{T}$ *does* prove $\delta$.

We can make headway on this question if we generalize it a bit. The left-to-right direction of the fixed point equivalence, $\mathsf{Prov}_T(\ulcorner\delta\urcorner) \to \delta$, is an instance of a general schema called a *reflection principle*: $\mathsf{Prov}_T(\ulcorner\varphi\urcorner) \to \varphi$. It is called that because it expresses, in a sense, that $\mathbf{T}$ can "reflect" about what it can prove; basically it says, "If $\mathbf{T}$ can prove $\varphi$, then $\varphi$ is true," for any $\varphi$. This is true for sound theories only, of course, and this suggests that theories will in general not prove every instance of it. So which instances can a theory (strong enough, and satisfying the provability conditions) prove? Certainly all those where $\varphi$ itself is provable. And that's it, as the next result shows.

**Theorem 5.8.** *Let $\mathbf{T}$ be an axiomatizable theory extending $\mathbf{Q}$, and suppose $\mathsf{Prov}_T(y)$ is a formula satisfying conditions P1–P3 from section 5.7. If $\mathbf{T}$ proves $\mathsf{Prov}_T(\ulcorner\varphi\urcorner) \to \varphi$, then in fact $\mathbf{T}$ proves $\varphi$.*

Put differently, if $\mathbf{T} \nvdash \varphi$, then $\mathbf{T} \nvdash \mathsf{Prov}_T(\ulcorner\varphi\urcorner) \to \varphi$. This result is known as Löb's theorem.

explanation

The heuristic for the proof of Löb's theorem is a clever proof that Santa Claus exists. (If you don't like that conclusion, you are free to substitute any other conclusion you would like.) Here it is:

1. Let $X$ be the sentence, "If $X$ is true, then Santa Claus exists."

2. Suppose $X$ is true.

3. Then what it says holds; i.e., we have: if $X$ is true, then Santa Claus exists.

4. Since we are assuming $X$ is true, we can conclude that Santa Claus exists, by modus ponens from (2) and (3).

5. We have succeeded in deriving (4), "Santa Claus exists," from the assumption (2), "$X$ is true." By conditional proof, we have shown: "If $X$ is true, then Santa Claus exists."

6. But this is just the sentence $X$. So we have shown that $X$ is true.

7. But then, by the argument (2)–(4) above, Santa Claus exists.

A formalization of this idea, replacing "is true" with "is provable," and "Santa Claus exists" with $\varphi$, yields the proof of Löb's theorem. The trick is to apply the fixed-point lemma to the formula $\mathsf{Prov}_T(y) \to \varphi$. The fixed point of that corresponds to the sentence $X$ in the preceding sketch.

*Proof.* Suppose $\varphi$ is a sentence such that $\mathbf{T}$ proves $\mathsf{Prov}_T(\ulcorner\varphi\urcorner) \to \varphi$. Let $\psi(y)$ be the formula $\mathsf{Prov}_T(y) \to \varphi$, and use the fixed-point lemma to find a sentence $\theta$

such that $\mathbf{T}$ proves $\theta \leftrightarrow \psi(\ulcorner\theta\urcorner)$. Then each of the following is provable in $\mathbf{T}$:

$$\theta \leftrightarrow (\mathsf{Prov}_T(\ulcorner\theta\urcorner) \to \varphi) \tag{5.14}$$ inc:inp:lob:L-1

$\theta$ is a fixed point of $\psi(y)$

$$\theta \to (\mathsf{Prov}_T(\ulcorner\theta\urcorner) \to \varphi) \tag{5.15}$$ inc:inp:lob:L-2

from eq. (5.14)

$$\mathsf{Prov}_T(\ulcorner\theta \to (\mathsf{Prov}_T(\ulcorner\theta\urcorner) \to \varphi)\urcorner) \tag{5.16}$$ inc:inp:lob:L-3

from eq. (5.15) by condition P1

$$\mathsf{Prov}_T(\ulcorner\theta\urcorner) \to \mathsf{Prov}_T(\ulcorner\mathsf{Prov}_T(\ulcorner\theta\urcorner) \to \varphi\urcorner) \tag{5.17}$$ inc:inp:lob:L-4

from eq. (5.16) using condition P2

$$\mathsf{Prov}_T(\ulcorner\theta\urcorner) \to (\mathsf{Prov}_T(\ulcorner\mathsf{Prov}_T(\ulcorner\theta\urcorner)\urcorner) \to \mathsf{Prov}_T(\ulcorner\varphi\urcorner)) \tag{5.18}$$ inc:inp:lob:L-5

from eq. (5.17) using P2 again

$$\mathsf{Prov}_T(\ulcorner\theta\urcorner) \to \mathsf{Prov}_T(\ulcorner\mathsf{Prov}_T(\ulcorner\theta\urcorner)\urcorner) \tag{5.19}$$ inc:inp:lob:L-6

by provability condition P3

$$\mathsf{Prov}_T(\ulcorner\theta\urcorner) \to \mathsf{Prov}_T(\ulcorner\varphi\urcorner) \tag{5.20}$$ inc:inp:lob:L-7

from eq. (5.18) and eq. (5.19)

$$\mathsf{Prov}_T(\ulcorner\varphi\urcorner) \to \varphi \tag{5.21}$$ inc:inp:lob:L-8

by assumption of the theorem

$$\mathsf{Prov}_T(\ulcorner\theta\urcorner) \to \varphi \tag{5.22}$$ inc:inp:lob:L-9

from eq. (5.20) and eq. (5.21)

$$(\mathsf{Prov}_T(\ulcorner\theta\urcorner) \to \varphi) \to \theta \tag{5.23}$$ inc:inp:lob:L-10

from eq. (5.14)

$$\theta \tag{5.24}$$ inc:inp:lob:L-11

from eq. (5.22) and eq. (5.23)

$$\mathsf{Prov}_T(\ulcorner\theta\urcorner) \tag{5.25}$$ inc:inp:lob:L-12

from eq. (5.24) by condition P1

$\varphi$ from eq. (5.21) and eq. (5.25)

□

With Löb's theorem in hand, there is a short proof of the first incompleteness theorem (for theories having a provability predicate satisfying conditions P1–P3: if $\mathbf{T} \vdash \mathsf{Prov}_T(\ulcorner\bot\urcorner) \to \bot$, then $\mathbf{T} \vdash \bot$. If $\mathbf{T}$ is consistent, $\mathbf{T} \nvdash \bot$. So, $\mathbf{T} \nvdash \mathsf{Prov}_T(\ulcorner\bot\urcorner) \to \bot$, i.e., $\mathbf{T} \nvdash \mathsf{Con}_{\mathbf{T}}$. We can also apply it to show that $\delta$, the fixed point of $\mathsf{Prov}_T(x)$, is provable. For since

$$\mathbf{T} \vdash \mathsf{Prov}_T(\ulcorner\delta\urcorner) \leftrightarrow \delta$$

in particular

$$\mathbf{T} \vdash \mathsf{Prov}_T(\ulcorner\delta\urcorner) \to \delta$$

and so by Löb's theorem, $\mathbf{T} \vdash \delta$.

**Problem 5.2.** Let $\mathbf{T}$ be a computably axiomatized theory, and let $\mathsf{Prov}_T$ be a provability predicate for $\mathbf{T}$. Consider the following four statements:

1. If $T \vdash \varphi$, then $T \vdash \mathsf{Prov}_T(\ulcorner\varphi\urcorner)$.
2. $T \vdash \varphi \to \mathsf{Prov}_T(\ulcorner\varphi\urcorner)$.
3. If $T \vdash \mathsf{Prov}_T(\ulcorner\varphi\urcorner)$, then $T \vdash \varphi$.
4. $T \vdash \mathsf{Prov}_T(\ulcorner\varphi\urcorner) \to \varphi$

Under what conditions are each of these statements true?

## 5.9 The Undefinability of Truth

inc:inp:tar: sec

The notion of *definability* depends on having a formal semantics for the language of arithmetic. We have described a set of formulas and sentences in the language of arithmetic. The "intended interpretation" is to read such sentences as making assertions about the natural numbers, and such an assertion can be true or false. Let $\mathfrak{N}$ be the structure with domain $\mathbb{N}$ and the standard interpretation for the symbols in the language of arithmetic. Then $\mathfrak{N} \vDash \varphi$ means "$\varphi$ is true in the standard interpretation."

**Definition 5.9.** A relation $R(x_1, \ldots, x_k)$ of natural numbers is *definable* in $\mathfrak{N}$ if and only if there is a formula $\varphi(x_1, \ldots, x_k)$ in the language of arithmetic such that for every $n_1, \ldots, n_k$, $R(n_1, \ldots, n_k)$ if and only if $\mathfrak{N} \vDash \varphi(\overline{n}_1, \ldots, \overline{n}_k)$.

Put differently, a relation is definable in in $\mathfrak{N}$ if and only if it is representable in the theory $\mathbf{TA}$, where $\mathbf{TA} = \{\varphi : \mathfrak{N} \vDash \varphi\}$ is the set of true sentences of arithmetic. (If this is not immediately clear to you, you should go back and check the definitions and convince yourself that this is the case.)

**Lemma 5.10.** *Every computable relation is definable in* $\mathfrak{N}$.

*Proof.* It is easy to check that the formula representing a relation in $\mathbf{Q}$ defines the same relation in $\mathfrak{N}$. □

Now one can ask, is the converse also true? That is, is every relation definable in $\mathfrak{N}$ computable? The answer is no. For example:

**Lemma 5.11.** *The halting relation is definable in* $\mathfrak{N}$.

*Proof.* Let $H$ be the halting relation, i.e.,

$$H = \{\langle e, x\rangle : \exists s\, T(e, x, s)\}.$$

Let $\theta_T$ define $T$ in $\mathfrak{N}$. Then

$$H = \{\langle e, x\rangle : \mathfrak{N} \vDash \exists s\, \theta_T(\overline{e}, \overline{x}, s)\},$$

so $\exists s\, \theta_T(z, x, s)$ defines $H$ in $\mathfrak{N}$. □

**Problem 5.3.** Show that $Q(n) \Leftrightarrow n \in \{^{\#}\varphi^{\#} : \mathbf{Q} \vdash \varphi\}$ is definable in arithmetic.

What about **TA** itself? Is it definable in arithmetic? That is: is the set $\{^{\#}\varphi^{\#} : \mathfrak{N} \vDash \varphi\}$ definable in arithmetic? Tarski's theorem answers this in the negative.

**Theorem 5.12.** *The set of true statements of arithmetic is not definable in arithmetic.* *inc:inp:tar: thm:tarski*

*Proof.* Suppose $\theta(x)$ defined it. By the fixed-point lemma, there is a formula $\varphi$ such that $\mathbf{Q}$ proves $\varphi \leftrightarrow \neg\theta(\ulcorner\varphi\urcorner)$, and hence $\mathfrak{N} \vDash \varphi \leftrightarrow \neg\theta(\ulcorner\varphi\urcorner)$. But then $\mathfrak{N} \vDash \varphi$ if and only if $\mathfrak{N} \vDash \neg\theta(\ulcorner\varphi\urcorner)$, which contradicts the fact that $\theta(y)$ is supposed to define the set of true statements of arithmetic. □

Tarski applied this analysis to a more general philosophical notion of truth. Given any language $L$, Tarski argued that an adequate notion of truth for $L$ would have to satisfy, for each sentence $X$,

> '$X$' is true if and only if $X$.

Tarski's oft-quoted example, for English, is the sentence

> 'Snow is white' is true if and only if snow is white.

However, for any language strong enough to represent the diagonal function, and any linguistic predicate $T(x)$, we can construct a sentence $X$ satisfying "$X$ if and only if not $T(\text{'}X\text{'})$." Given that we do not want a truth predicate to declare some sentences to be both true and false, Tarski concluded that one cannot specify a truth predicate for all sentences in a language without, somehow, stepping outside the bounds of the language. In other words, a the truth predicate for a language cannot be defined in the language itself.

# Photo Credits

# Bibliography