FORMAL BACKGROUND FOR THE INCOMPLETENESS AND UNDEFINABILITY THEOREMS

RICHARD G. HECK, JR.

CONTENTS

The 1930s saw enormous advances in logic. One of the most important of these came in 1931, when Kurt Gödel proved his famous 'incompleteness theorems'. Not only did these results transform mathematical logic, but they have had a significant influence on philosophy, and even on popular culture. Unfortunately, however, much of the discussion of 'Gödel's theorem' is marred by a poor understanding of his results.

The only real way to acquire an understanding of Gödel's result is to work one's way through a proof of it. But it would, it seems to me, be a shame if students who are, for whatever reason, unwilling or unable to take a course on Gödel's theorem were, for that same reason, barred from any real understanding of this important result. And one can acquire such an understanding without wading through all the details that—as conceptually interesting as they may be in their own right—throw no real light on the incompleteness theorem itself. Hence this note, whose purpose is to make it possible for those have a firm grasp of basic logic to acquire a solid understanding of exactly what Gödel proved and, at least roughly, how he proved it.[1]

As will be familiar to insiders, the techniques needed for the proof of Gödel's theorem also allow us to prove Tarski's theorem on the indefinability of truth. We'll prove that here, too. A separate note discusses the positive side of Tarski's work on truth: his provision of a method that, applied to any of a wide class of languages, yields a "materially adequate" theory of truth for that language, the 'object language'.[2]

1. Languages

Logic, as it is done in an introductory course, is generally concerned with 'schemata', such as: $\forall x(Fx \rightarrow \exists y(Rxy))$. Here, F and R are 'predicate letters', whose interpretation varies from case to case. In more advanced logic, we are often concerned instead with formulas whose meaning we treat as more or less fixed, for example: $\forall x(0 < x \rightarrow x < x + x)$. And the set of formulas in which we are interested is often drawn from a fixed 'language', whose 'vocabulary' is determined in advance.

A *language*, in our technical sense, is just a collection of symbols. Any collection of symbols may be regarded as a language, but we shall only be interested here in so-called 'first-order' languages. Such languages are generated, in the obvious way, from the basic logical symbols, such as $\neg$, $\vee$, $\exists$,[3] and the variables, plus as many 'non-logical' symbols as one likes. These may include constants, such as '0', function symbols, such

[1] What follows draws heavily upon the presentation by Boolos and Jeffrey (1989), which is where I learned this stuff. See especially Ch. 15. More generally, my own understanding of the 'limitative theorems' was heavily influenced by discussions with Boolos and by his general logico-philosophical orientation.

[2] See my "Tarski's Theory of Truth", available on my web site.

[3] Familiarly, exactly which symbols we include is to some extent a matter of choice and convenience.

as '$+$' or 'g', and predicate symbols, such as '$<$' or 'F', of various numbers of arguments.[4] We assume that these languages have the usual syntax of first-order logic, i.e., that formulas are built up from the primitives in the usual way. And we regard languages as *fixed*—a language never gains or loses vocabulary—and we identify a language with its stock of non-logical expressions.[5]

One important language is the *language of arithmetic*, whose (basic) non-logical symbols are the constant '0', the one-place function-symbol 'S', a pair of two-place function-symbols, '$+$' and '$\times$', and a two-place relation symbol '$<$'.[6] Another important language is the language of set theory, whose only non-logical symbol is the two-place relation symbol '$\in$'. So '$0 + Sx$' is an (open) term of the language of arithmetic and '$\forall x \exists y(x = y + Sy)$' is a sentence of that language. On the other hand, neither '$\exists z(+x)$' nor '$\forall x(Fx \rightarrow Gx)$' is a sentence of the language of arithmetic. The former is not well-formed and so it is not a sentence of any (first-order) language. The latter is a sentence of some languages, indeed, of any language that contains both 'F' and 'G' as one place predicate-symbols. But it is not a sentence of the language of arithmetic, which does not contain either.

Strictly speaking, as syntax is usually developed, much of the previous paragraph is false. Strictly speaking, there are always supposed to be parentheses around the various terms and formulas that comprise a larger term or formula in order to guarantee 'unique readability', i.e., to prevent ambiguity. Thus, officially, we should write something like: '$(Sy + y)$', rather than: '$Sy + y$', since the latter could also mean '$S(y + y)$'. In practice, when nothing depends upon their visibility, we write parentheses with invisible ink. But it is worth recalling, from time to time, that, strictly speaking, the string '$\forall x \exists y(x = y+Sy)$' is not a sentence of the language of arithmetic, though the string '$\forall x\,(\exists y\,(x = (y + Sy)))$' is. That said, however, we shall henceforth promptly forget about this point.[7]

It simplifies things considerably if one defines the language of arithmetic in such a way that its 'alphabet' is finite, i.e., if every term, sentence,

---

[4] Sentence-letters are not often useful, but if one wants them, a sentence-letter may be regarded as a zero-place predicate-letter.

[5] All of these restrictions can be relaxed. For example, one might regard some languages as containing the means for defining new expressions.

[6] The last is often omitted and instead treated as a defined symbol, but it is convenient, especially in the study of weak fragments of arithmetic, to take it as primitive.

[7] One has some choice about where to put parentheses, and in some developments other sorts of notation are used, e.g., so called 'Polish notation', which does not need parentheses. Note, too, that, since the conventions that allow the invisibility of parentheses define recursive functions between 'strict' and 'sloppy' formulae, we can, if we like, allow the conventions to be part of the formal syntax, though this will make the syntax complicated.

etc., is constructed from some finite number of primitive symbols. Of course, the logical symbols, and special arithmetical symbols, are finite in number, but there are infinitely many variables. We can, however, construct our infinitely many variables from a finite alphabet, as follows: x, x', x'', etc.; a variable is thus an 'x' followed by a (possibly empty) string of prime-symbols. We shall, however, frequently abbreviate 'x'' as: y; 'x''' as: z, and so forth, to enhance readability. Where convenient, we may also use 'x_2', e.g., as an abbreviation for 'x''''.

Many of the proofs we will give below are proofs about the syntactic features of languages. And one of the most common methods used in such proofs is what is known as *induction on the complexity of expressions*. This method is similar to 'mathematical' induction. In arithmetic, one can show that every number has a certain property P by showing that:

(i) 0 has P
(ii) If n has P, then $n + 1$ must also have P

This works because every number is either zero or is the result of adding one to zero some finite number of times. But similarly: Every term in the language of arithmetic is either a basic term—either '0' or a variable—or is the result of applying certain syntatic operations to the basic terms. So we can show that all terms have P by showing that:

(i) '0' has P, and every variable has P
(ii) Whenever terms t and u have P, so do the terms $\ulcorner St \urcorner$, $\ulcorner t + u \urcorner$, and $\ulcorner t \times u \urcorner$

Similarly, we can show that all formulae have some property P by showing that:

(i) All atomic formulae (i.e., all formulae of the form $\ulcorner t = u \urcorner$ and $\ulcorner t < u \urcorner$) have P
(ii) Whenever two formulae A and B have P, so do $\ulcorner \neg A \urcorner$, $\ulcorner A \vee B \urcorner$, $\ulcorner A \wedge B \urcorner$, $\ulcorner A \rightarrow B \urcorner$, $\ulcorner \exists v A \urcorner$ and $\ulcorner \forall v A \urcorner$, for any variable v.

Indeed, *any* time one has a set whose elements are generated from some 'basic' elements by certain 'operations' on those elements, one can always show that all members of the set have some property P by showing (i) that all the basic elements have P and (ii) that the contructing operations 'preserve' P, i.e.: If you have some things that have P and construct a new thing using one of the operations, then you get something that also has P.

The language of arithmetic is an *interpreted* language, a language whose formulae and sentences mean something. What they mean is determined by what is called the 'standard interpretation' of the language. The standard interpretation is an interpretation, in the usual sense: It has a domain, and it assigns values to constants, functions to

function-symbols, and relations to relation-symbols. In this case, the interpretation in which we are interested is this one:[8]

- The domain is the set of all natural numbers
- '0' denotes zero
- 'S' denotes successor, or plus one
- '$+$' denotes addition
- '$\times$' denotes multiplication
- '$<$' denotes the less-than relation

So we may now regard a given sentence, say, '$\forall x \exists y(x = y+y)$', as meaning something, in this case that every number is the sum of some number and itself (roughly, that every number is even).

Although the language of arithmetic has a standard interpretation, this does not stop us from considering other interpretations, too. We need to do so for the usual reason, e.g., to show that '$\forall y \exists x(y < x)$', though true in the standard interpretation, is not logically valid.

Expressions like '$SS0$'—that is, strings of 'S's, followed by a '0'—are called *numerals*. Note that, in the standard interpretation, a string of n 'S's followed by '0' denotes the number n. (See Exercise 15.1.) Thus, '$SS0$' denotes the number 2 in the standard interpretation, and so what '$SS0 = S0$' *says*, as it were, is that $2 = 1$. We write $\bar{n}$ to mean: the numeral that denotes n, that is:

$$\underbrace{S \ldots S}_{n} 0$$

So we may now write things like $\ulcorner \overline{2} \neq \overline{1} \urcorner$, to mean: $SS0 \neq S0$, i.e, that 2 is not 1.

Note the use of the funny quotes, which are called corners and which are strictly speaking necessary. Strictly speaking, '$\overline{2} \neq \overline{1}$' is not a formula of the language of arithmetic, for the simple reason that '$\overline{2}$' is not a symbol of the language of arithmetic but is, rather, a *name* of such a symbol, namely, of '$SS0$'. (Strictly speaking, '$\neq$' is not in the language of arithmetic either but is an abbreviation. But let us not get *too* pedantic.) Corners are a solution to this problem, introduced by Quine. Exactly how they work is not easily explained, but one does get the hang of it fairly quickly. In much technical writing nowadays, the corners are just written as ordinary quotes and context is left to distinguish them. On occasion, quotes themselves are even written with invisible ink. We may indulge in this practice ourselves from time to time. But we shall try not to overdo it.

[8] If we want to use the standard reduction of functions to sets, then the extension of 'S' is the set: $\{< x, y >: y = x + 1\}$; that of '+ ' is: $\{< x, y, z >: x + y = z\}$; etc. But it is easier to speak, as Frege does, in terms of functions. Similarly for relations.

2. THEORIES

In basic logic, we concern ourselves with such questions as whether a certain schema is valid, or whether some schemata imply some other schema. In more advanced logic, we are also interested in such questions, but attention tends to be focused on certain specific sets of schemata, and what they do or do not imply. Consider, for example, these three:

$$\forall x \forall y \forall z (x \cdot (y \cdot z) = (x \cdot y) \cdot z)$$
$$\forall x (x \cdot 1 = x \wedge 1 \cdot x = x)$$
$$\forall x \exists y (x \cdot y = 1 \wedge y \cdot x = 1)$$

The language here contains the constant symbol '1' and a two-place function symbol '$\cdot$'. These three formula together constitute the axioms of what is known in abstract algebra as *group theory*, and we may regard them as a formulation of the *theory of groups*. Then we can bring the tools for formal logic to bear. It is easy to provide an interpretation in which all of these axioms hold but in which, say, '$\forall x \forall y (x \cdot y = y \cdot x)$' does not hold, thus showing that the axioms of group theory do not imply that the group operation $\cdot$ is commutative (i.e., that it is logically possible for there to be so-called non-Abelian groups).

We now generalize this idea.

Definition 2.1. A *theory* is a set of formulae of some fixed language, called the language of the theory. A theory is *stated in* or *formulated in* its language.

We shall typically use capital Greek letters, such as Σ, to range over theories. Note that any set of formulae constitutes a theory, although some theories will be more interesting than others.

Here are some more examples of theories.

(1) The set

$$\{\forall x \forall y (Sx = Sy \rightarrow x = y), \neg \exists x (0 = Sx)\}$$

is a theory stated in the language of arithmetic. It is also a theory in the language $\{0, S\}$, where '0' is a constant, and 'S' is a one-place function-symbol. These are two theories, since a theory is a pair $< \mathcal{T}, \mathcal{L} >$ of a set of formulae and a language.

(2) The theory known as *weak* (or *adjunctive*) *set theory* is the set:

$$\{\exists x \forall y (y \notin x), \forall u \forall v \exists x \forall y (y \in x \equiv y \in u \vee y = v)\}$$

Its language is the language of set theory: $\{\in\}$.

The empty set is always a theory, though we do need to specify, in each case, what the language of the theory is.

We assume here a notion of *derivation* provided by ordinary first-order logic. We say that A is derivable from the set of formulae Γ if there is a formal derivation of A whose premises are all in Γ. Note that not all the

sentences in Γ need be used as premises—and of course they cannot be if Γ is infinite, which is perfectly possible.

Definition 2.2. A formula A is a *theorem of* the theory Σ if, and only if, A is derivable from Σ.

We write: $\Sigma \vdash A$, to mean that A is a theorem of Σ.

One also sees such things as: $A \vdash B$ and $\Sigma, A \vdash B$—meaning that B is derivable from A, or from Σ together with A—where strictly we should write: $\{A\} \vdash B$, or: $\Sigma \cup \{A\} \vdash B$. We also read '$\Sigma \vdash B$' as '$\Sigma$ proves B'. One also sees the notation: $\vdash_\Sigma B$, meaning: $\Sigma \vdash B$. This is more common when Σ is some important theory about which we are proving a number of facts.

The notion of derivability is not the same as the notion of implication. Whether a theory Σ *implies* a given formula A is a matter of whether A is true under every interpretation that makes every formula in Σ true. Whether Σ *proves* A is a matter of whether a formal derivation of a certain sort exists. The completeness theorem for first-order logic (first proved by Gödel, in 1929) tells us that, *in the case of first-order logic*, it is possible to specify a formal system (indeed, many formal systems) that have the property that Σ proves A (in that system) if, and only if, Σ implies A. But this is a deep theorem about first-order logic, not a trivial fact of terminology. It is, indeed, a *very* deep theorem, one whose analogue for other sorts of logics does not always hold.[9]

As noted, Σ implies A if, and only if, A is true in every interpretation in which all formulae in Σ are true. Interpretations in which all formulae in Σ are true are thus of special interest. We call such an interpretation a *model* of Σ. So Σ implies A if, and only if, A is true in every model of Σ. If there is a model of Σ in which A is not true, then Σ does not imply A, so A is not derivable from Σ (by the soundness theorem), so A is not a theorem of Σ. Such a model is called a *counter-model* for A.

What we earlier called the 'standard interpretation' of the language of arithmetic is thus also known as the *standard model* of the sorts of arithmetical theories we shall consider in Section 5.

All we need to know for present purposes is that *there is* a sound and complete system of axioms and rules for first-order logic available for use in derivations. To say that such a system is *sound* is to say that, whenever Γ proves A, also Γ implies A. To say that it is *complete* is to say that the converse holds: If Γ implies A, then Γ proves A. So, if a formal system is sound and complete, then Σ implies A if, and only if, Σ proves A; and this will hold for any system of axioms and rules that is both sound and complete. We will appeal to this fact frequently, without explicit mention, and without bothering to say how exactly the logic is being formalized. See Appendix A for one way of formalizing it, however.

[9] In particular, the completeness theorem fails for second-order logic (with the so-called 'standard semantics').

3. Some Important Properties of Theories

Definition 3.1. A theory Σ is *consistent* if for no formula A is $\ulcorner A \wedge \neg A \urcorner$ derivable from Σ. If Σ is not consistent, we say that it is *inconsistent*.

Note that we cannot state this definition as: ... iff '$p \wedge \neg p$' is derivable from Σ. There is no guarantee that '$p \wedge \neg p$' is in the language of Σ. It is not, for example, in the language of arithmetic. Note also that inconsistency, as it has been defined here (and as the notion is usually used in logic), is a *syntactic* property (which has to do with derivability) not a *semantic* one (which has to do with models). We can say that a theory if *satisfiable* if it has a model, but we will not really have use for this notion.[10]

Proposition 3.2. *Σ is consistent if, and only if, not every formula is derivable from Σ.*

Proof. See Exercise 15.2. □

Proposition 3.3. *$\Sigma \vdash A$ iff $\Sigma \cup \{\neg A\}$ is inconsistent.*

Proof. Suppose $\Sigma \vdash A$. Then certainly $\Sigma \cup \{\neg A\} \vdash A$. But $\Sigma \cup \{\neg A\} \vdash \neg A$, as well. So $\Sigma \cup \{\neg A\} \vdash A \wedge \neg A$.

Suppose, then, that $\Sigma \cup \{\neg A\}$ is inconsistent. Then $\Sigma \cup \{\neg A\} \vdash B \wedge \neg B$, for some B. So $\Sigma \vdash \neg A \rightarrow B \wedge \neg B$, so $\Sigma \vdash A$. □

Since we have not said what formal system of deduction we are using, the argument just given tacitly appeals several times to soundness and completeness, and to some basic facts about implication. Exercise 15.3 asks you to re-write the argument to make those appeals explicit. Note, however, that, if we *do* specify a system of deduction, then Proposition 3.3 can be proven purely syntactically. That is, we can show directly that, whenever there is a derivation of A from Σ, that derivation can be 'converted' into a proof of a contradiction from $\Sigma \cup \{\neg A\}$.

Definition 3.4. A theory Σ is *complete* if, and only if, for every sentence A in the language of the theory, either $\Sigma \vdash A$ or $\Sigma \vdash \neg A$. If Σ is not complete, we say that it is *incomplete*.

If Σ is incomplete, there is a formula A such that neither A nor $\neg A$ is derivable from Σ (and so there are infinitely many such formulae: $A \wedge A$, $A \wedge A \wedge A$, etc).

Note that every inconsistent theory is complete, since *both* A and $\neg A$ will be derivable from it, for any formula A. Inconsistent theories are boring,[11] so let us mean by a complete theory a complete *consistent*

[10] Of course, the soundness and completeness of our logic guarantees that a theory is consistent just in case it is satisfiable.

[11] Assuming, that is, that we are not working in a so-called *paraconsistent* logic. In such logics, contradictions do not imply everything and so Proposition 3.2 does not

theory. Note also that the completeness of *theories* is a very different matter from the completeness of a formal system of logic.

Definition 3.5. A *closed theory* is one that is 'closed under derivability'. That is: Σ is closed iff every formula (in the language of Σ) that is deducible from Σ is already in Σ. More formally: Σ is closed iff, if $\Sigma \vdash A$, then $A \in \Sigma$.

Some authors use the term 'theory' to mean what I am here calling a 'closed' theory. So, on this usage, a 'theory' is a set of *theorems* of a theory in our sense. For some purposes, it is sufficient to distinguish theories in this way: by what they prove. But for our purposes, it is important to individuate theories in a finer-grained way: in terms of their *axioms*. The terminology, of course, is not what matters here. It is what one takes to be the fundamental notion.

Note that closed theories need not be complete: The set of all logically valid formulas of, say, the language of arithmetic is closed but not complete. Nor need a complete theory be closed. To give a fairly trivial example, it can be shown that the theory $\{\forall x \forall y(x = y)\}$, in the empty language—sometimes called the theory of a one-element model—is complete, but it is clearly not closed.

4. Comparing the Strength of Theories

An important subject of logical investigation concerns the relative strength of two theories. If the theories are formulated in the same language, then one way of addressing this kind of question is obvious: A theory $\mathcal{T}$ will be stronger than another theory $\mathcal{T}'$ if every theorem of $\mathcal{T}'$ is a theorem of $\mathcal{T}$.

Definition 4.1. A theory Θ *extends* a theory Σ if all theorems of Σ are theorems of Θ, i.e,. if, whenever $\Sigma \vdash A$, also $\Theta \vdash A$. We also say that Θ *contains* Σ.

But what if the languages are not the same? Is there some way of comparing strength, then? We'll here address a fairly simple case, in which the language of the one theory is an expansion (superset) of the language of the other.[12]

For the rest of this section, let $\mathcal{T}$ and $\mathcal{T}^*$ be theories in the languages $\mathcal{L}$ and $\mathcal{L}^*$, respectively, and suppose $\mathcal{L} \subseteq \mathcal{L}^*$. If $\mathcal{L}$ is a proper subset of $\mathcal{L}^*$, then *of course* $\mathcal{T}^*$ will have theorems that are not theorems of $\mathcal{T}$, simply because there are sentences in the language of $\mathcal{T}^*$ that are not

hold. In that setting, we need to distinguish inconsistent theories, which prove some contradictions, from *trivial* theories, which prove everything. Inconsistent theories need not be boring in such contexts, though trivial theories are.

[12]The usual method of comparing the strength of theories when the languages are not so related involves the notion of (syntactic) interpretation, which we shall not discuss here.

sentences in the language of $\mathcal{T}$. For example, if $A \in \mathcal{L}^*$ but $A \notin \mathcal{L}$, then $\mathcal{T}^* \vdash A \vee \neg A$, since that is a logical truth of $\mathcal{L}^*$, but of course $\mathcal{T} \nvdash A \vee \neg A$, simply because $A \vee \neg A$ is not in the language of $\mathcal{T}$. So the most for which one could hope is that the *only* 'new' theorems of $\mathcal{T}^*$ are ones not in $\mathcal{L}$, that is, that every sentence that even *might* be a theorem of $\mathcal{T}$—that is, every sentence in the language of $\mathcal{T}$—that is a theorem of $\mathcal{T}^*$ is already a theorem of $\mathcal{T}$. This motivates the following definition.[13]

Definition 4.2. $\mathcal{T}^*$is a *conservative extension* of $\mathcal{T}$ if, for every formula $A \in \mathcal{L}$, if $\mathcal{T}^* \vdash A$, then already $\mathcal{T} \vdash A$.

The easiest way to show that one theory is a conservative extension of another is via a simple model-theoretic argument.

Definition 4.3. Let $\mathcal{M}$ be an interpretation of $\mathcal{L}$. An interpretation $\mathcal{M}^*$ of $\mathcal{L}^*$ is an *expansion* of $\mathcal{M}$ if $\mathcal{M}$ and $\mathcal{M}^*$ differ only in what they assign to non-logical symbols in $\mathcal{L}^*$ that are not in $\mathcal{L}$. That is: $\mathcal{M}$ and $\mathcal{M}^*$ have the same domain, and they assign the same values to the non-logical symbols of $\mathcal{L}$.

Fact 4.4. *Let $\mathcal{M}$ be an interpretation of $\mathcal{L}$; $\mathcal{M}^*$, an expansion of $\mathcal{M}$. Suppose that t is an expression of $\mathcal{L}$. Then the value of t in $\mathcal{M}$ is the same as the value of t in $\mathcal{M}^*$. In particular, if A is a sentence in $\mathcal{L}$, then A is true in $\mathcal{M}^*$ iff A is true in $\mathcal{M}$.*

Intuitively, the point is simply that the value of an expression in an interpretation depends only upon the domain of the interpretation and upon what it assigns to the non-logical symbols occurring in that expression.

The proof is by induction on the complexity of expressions. We first need to prove the corresponding fact for terms, and we need to take account, as well, of assignments to variables.

Fact 4.5. *Let $\mathcal{M}$ be an interpretation of $\mathcal{L}$; $\mathcal{M}^*$, an expansion of $\mathcal{M}$. Suppose that t is a term of $\mathcal{L}$. Then the value of t in $\mathcal{M}$, under any assignment of values to varaibles in t, is the same as the value of t in $\mathcal{M}^*$, under the same assignment of values to variables.*

Proof. For the basis of the induction, we need to consider the constants of $\mathcal{L}$, if there are any, and the variables. Now the variables obviously have the same value, since the assignment is the same; and the constants too have the same value, since they are in $\mathcal{L}$, and $\mathcal{M}$ and $\mathcal{M}^*$ assign the same values to elements of $\mathcal{L}$.

For the induction step, let f_n be an n-place function symbol in $\mathcal{L}$ (if such there are). We need to show that, if $t_1, \ldots, t_n$ satisfy the theorem, then so does $f_n(t_1, \ldots, t_n)$. Let α be an assignment. By the induction hypothesis, the t_i all have the same value, under α, in both $\mathcal{M}$ and $\mathcal{M}^*$.

[13]The notion defined here is sometimes known as *syntactic* conservativity. There are also semantic notions in the same general vicinity.

And since $f_n \in \mathcal{L}$, $\mathcal{M}$ and $\mathcal{M}^*$ assign it the same function as its value. So, in both cases, the value of $f_n(t_1, \ldots, t_n)$ is the result of applying this function to the values of the t_i, which are the same in both cases; so the result is the same in both cases. □

Proof of 4.4. Very similar to the proof just given, though more complicated, due to the need to take account of the quantifers. □

Theorem 4.6. *If every model of $\mathcal{T}$ can be expanded to a model of $\mathcal{T}^*$, then $\mathcal{T}^*$is a conservative extension of $\mathcal{T}$.*

Proof. Suppose A is a sentence of $\mathcal{L}$ that is not a theorem of $\mathcal{T}$. By the completeness theorem, there is a model $\mathcal{M}$ of $\mathcal{T}$ in which A is not true. So, by assumption, $\mathcal{M}$ can be expanded to a model $\mathcal{M}^*$ of $\mathcal{T}^*$. Since $A \in \mathcal{L}$, however, A must still not be true in $\mathcal{M}^*$, by Fact 4.4. So A is not a theorem of $\mathcal{T}^*$. □

5. Arithmetical Theories

We shall be especially interested here in theories formulated in the language of arithmetic.

Robinson Arithmetic, or Q, is the theory with the following eight axioms:[14]

Q1	$\forall x(Sx \neq 0)$
Q2	$\forall x \forall y(Sx = Sy \rightarrow x = y)$
Q3	$\forall x(x + 0 = x)$
Q4	$\forall x \forall y(x + Sy = S(x + y))$[15]
Q5	$\forall x(x \times 0 = 0)$
Q6	$\forall x \forall y(x \times Sy = (x \times y) + x)$[16]
Q7	$\forall x(x \neq 0 \rightarrow \exists y(x = Sy))$
Q8	$x < y \equiv \exists z(y = Sz + x)$

It can be shown that these axioms are independent, i.e., that none of them is provable from the rest.

Note that all axioms of Q are true in the standard interpretation of the language of arithmetic. Moreover, as its namesake, Raphael Robinson, noted, Q is distinguished by the simplicity and clear mathematical content of its axioms. It's hard to imagine disputing Q, unless one wanted to dispute the very coherence of arithmetical notions.

In some ways, Q is a very weak theory. There are all kinds of very elementary facts about the natural numbers that Q does not prove. For example:

Fact 5.1. Q *does not prove* $\forall x(x \neq Sx)$.

[14] When '$<$' is not included in the language, Q8 is taken to define it.

[15] I.e., $x + (y + 1) = (x + y) + 1$.

[16] I.e., $x \times (y + 1) = (x \times y) + x$.

Proof. We show this by constructing a counter-model for $\forall x(x \neq Sx)$, i.e., one in which it is false, but in which the axioms of Q are true. To do that, we need to say what the domain of the model is, and what the interpretations of '0', 'S', '$+$', '$\times$', and '$<$' are.

Let the domain of the model consist of the natural numbers, $\mathbb{N}$,[17] and one additional element, say, Augustus Caesar, which we shall call A. The expressions '0', 'S', '$+$', '$\times$', and '$<$' are to behave as usual on the numerical part of the model. So we just need to say how they behave when A is involved.

The extension of 'S' is to be:

$$\{\textbf{<}x, y\textbf{>} : (x \in \mathbb{N} \wedge y = n + 1) \vee (x = A \wedge y = A)\}$$

That is, roughly speaking, the successor of x is what it usually is if $x \in \mathbb{N}$; it is A if x is A. This will guarantee that $\forall x(x \neq Sx)$ is false in the model, since '$x \neq Sx$' will be false when 'x' has A as value. But we cannot stop here, since we have not completed the definition of the model. We need also to specify the interpretations of '$+$', '$\times$', and '$<$'.

The extension of '$+$' is to be:

$$\{\textbf{<}x, y, z\textbf{>} : (x, y, z \in \mathbb{N} \wedge x + y = z) \vee ((x = A \vee y = A) \wedge z = A)\}$$

That is, roughly: $A + x = x + A = A$, no matter what x may be.

- Let $A \times 0 = 0 \times A = 0$; if $n \in N$ and $n \neq 0$, let $A \times n = A \times n = A$; let $A \times A = A$.

This more compressed specification is, strictly speaking, inaccurate, but it is undeniably convenient, and it is easy enough to convert it to a more formal specification if one wishes to do so. It is fine to use it so long as one realizes what one is actually doing, namely, specifying an interpretation of the symbol '$\times$'.

- Let $x < y$ iff $\exists z(x+Sz = y)$, where '$+$' is defined as just specified.[18]

We must now verify that the axioms of Q all hold in this model.

For Q1, we need to show that '$Sx \neq 0$' is true, no matter what the value of 'x' may be, i.e., that the value of the term 'Sx' is never 0. Now, either the value of 'x' is some $n \in \mathbb{N}$ or else it is A. If the former, then the value of 'Sx' is $n + 1 \neq 0$; if the latter, it is $A \neq 0$.

For Q2, we need to show that '$Sx = Sy \rightarrow x = y$' is true, for all assignments to 'x' and 'y', i.e., that if '$Sx = Sy$' is true, so is '$x = y$'; i.e., that if 'Sx' and 'Sy' have the same value, then so do 'x' and 'y'. Suppose the value of 'x' is $n \in \mathbb{N}$. Then the value of 'Sx' is $n + 1$. So, if 'Sy' has the same value, i.e., $n + 1$, we the value of 'y' must also be n, in which case 'x'

[17] It can be shown that every model of Q will have to contain, at least as a part, a copy of the natural numbers. (More strictly: Every model of Q contains a part that is isomorphic to the standard model.)

[18] Thus, $A < A$ since $A + S0 = A$; moreover, for any $n \in N$, $n < A$, since $n + SA = A$. On the other hand, $\neg A < n$, since for no x do we have $A + Sx = n$.

and 'y' have the same value. And if the value of 'x' is A, then the value of 'Sx' is A; and 'Sy' cannot have the same value unless also the value of 'y' is A.

For Q3, we need to show that '$x + 0 = x$' is true, no matter what the value of 'x' may be. Let the value of 'x' be n. If $n \in \mathbb{N}$, then the value of the term '$x + 0$' is the sum of the value of 'x' and 0, i.e., $n + 0$, i.e., n; and of course the value of 'x' is also n; hence '$x + 0 = x$' is true. If, on the other hand, $n = A$, then the value of '$x + 0$' is A, which is also the value of 'x'; hence, again, '$x + 0 = x$' is true. □

It is natural to want to compress the foregoing as follows.

> For Q1: If $n \in \mathbb{N}$, then $Sn = n + 1 \neq 0$; and $SA = A \neq 0$.
> For Q2: If $n \in \mathbb{N}$, then $Sn = n + 1$, so $Sn = Sm$ iff $Sm = n + 1$ iff $m = n$. Moreover, $SA = Sy$ iff $y = A$.
> For Q3: If $n \in \mathbb{N}$, then $n + 0 = n$; and also $A + 0 = A$.

This is all fine, in a way, but it is important to see that it really is an abuse of language. The symbol '+' is not being used uniformly in this argument. Sometimes, it is being used to mean ordinary addition (as when we assert that $n + 1 \neq 0$) and sometimes it is not (as when we speak of $A + 0$). More generally, the 'shorter proof' obscures the character of what is being proved, which concerns a *semantic* fact about certain *symbols*, namely, that those symbols—the formulae expressing the axioms of Q—are true in a certain interpretation.

In practice, however, this sort of compression is, again, undeniably convenient, and so we shall now indulge ourselves.

Proof of 5.1, completed. For Q4: If $n, m \in \mathbb{N}$, then $n + Sm = n + (m + 1) = (n + m) + 1 = S(n + m)$, and also $A + Sm = A = A + m = S(A + m)$, whatever m is.

For Q5: $x \times 0 = x$, whether $x \in \mathbb{N}$ or not.

For Q6, then, we need to show that $x \times Sy = x \times y + x$. If $x, y \in \mathbb{N}$, then this is obvious, since everything behaves as usual on the numerical part of the model. I.e., we really only need to be checking what happens when either $x = A$ or $y = A$. But $A \times Sy = A$, no matter what y is; and also $A \times y + A = A$, no matter what y is (indeed, no matter what $A \times y$ is).

For Q7, we need to show that $x \neq 0 \to \exists y(x = Sy)$. The only case we need check is when $x = A$, but $A = SA$.

Finally, Q8 holds because we just defined $<$ according to it. (This is usually a good idea.) □

One can show in a similar way that Q does not prove that addition is commutative: That is, '$\forall x \forall y(x + y = y + x)$' is not a theorem of Q. Even '$\forall x(0 + x = x)$' turns out not to be a theorem of Q. (This is Exercise 15.7.) In another sense, however, Q is quite a powerful theory, as we shall see.

Q has a nice property we shall need later: It proves all basic arithmetical equalities and inequalities. By this we mean, for example, that

whenever n is not m, Q proves $\ulcorner \overline{n} \neq \overline{m} \urcorner$. So, for example, since $2 \neq 1$, Q proves $\ulcorner \overline{2} \neq \overline{1} \urcorner$, or: $SS0 \neq S0$. Indeed, much more is true.

Proposition 5.2. Q *proves all basic arithmetical equalities and inequalities. That is:*

(1) If $n = m$, Q $\vdash \overline{n} = \overline{m}$
(2) If $n \neq m$, Q $\vdash \overline{n} \neq \overline{m}$
(3) If $n + m = k$ *, then* Q $\vdash \overline{n} + \overline{m} = \overline{k}$
(4) If $n + m \neq k$, *then* Q $\vdash \overline{n} + \overline{m} \neq \overline{k}$
(5) If $n \times m = k$ *, then* Q $\vdash \overline{n} \times \overline{m} = \overline{k}$
(6) If $n \times m \neq k$, *then* Q $\vdash \overline{n} \times \overline{m} \neq \overline{k}$
(7) If $m < n$, *then* Q $\vdash \overline{m} < \overline{n}$
(8) If $\neg m < n$, *then* Q $\vdash \neg \overline{m} < \overline{n}$

Proof. For part (1), let $n = m$. Then the claim is that Q proves $\overline{n} = \overline{m}$. But this is just $\overline{n} = \overline{n}$, which is a logical truth.

For part (2), the proof is by induction on m. Before we actually proceed with the proof, however, let me make a few remarks about it. This is a proof by what is sometimes called 'external' induction. It is an induction *we* are carrying out in the language *we* are speaking. It is not an argument we are carrying out in Q, or even one that can be 'formalized' in Q. We shall see later, in Section 11, that Q is perfectly capable of talking about what can be proven in Q. Nonetheless, however, the argument we are giving, to show establish (2), is not an argument that can be given in Q, since Q doesn't have any axioms corresponding to mathematical induction.

Second, the proof involves a number of syntactic claims. For example, we shall shortly need to know that, if $n \neq 0$, then the numeral for n, $\overline{n}$, is the result of adding a successor symbol to the numeral for $n - 1$, i.e., it is $S(\overline{n-1})$. Note, as said, that this is a *syntactic* claim, not an arithmetical one: The claim is one about what expression $\overline{n}$ is, namely, that it is the result of adding a successor symbol to another numeral.

To return to the proof, then, for the basis, we must show that, if $n \neq 0$, then Q proves $\overline{n} \neq 0$. Now, as said, since $n \neq 0$, $\overline{n}$ is $S(\overline{n-1})$. What must be shown, then, is just that Q proves $S(\overline{n-1}) \neq 0$. But that is an instance of axiom Q1.

For the induction step, we suppose that Q proves $\overline{n} \neq \overline{m}$ when $n \neq m$ and need to show that Q proves $\overline{n} \neq \overline{m+1}$ when $n \neq m + 1$. Note first that this certainly holds if $n = 0$, by the preceding. So suppose $n \neq 0$. Then $\overline{n}$ is $S(\overline{n-1})$, and $\overline{m+1}$ is just $S(\overline{m})$, so what we need to show is that Q proves $S(\overline{n-1}) \neq S(\overline{m})$, whenever, in fact, $n \neq m + 1$. But then $n - 1 \neq m$, so by the induction hypothesis, Q proves $\overline{n-1} \neq \overline{m}$. By Q2, however, Q proves $S(\overline{n-1}) = S(\overline{m}) \rightarrow \overline{k} = \overline{m}$. But then it proves $S(\overline{n-1}) \neq S(\overline{m})$, by *modus tollens*.

The proof of part (3) is by induction on m. For the basis, we must show that, if $n + 0 = k$, then Q proves $\overline{n} + 0 = \overline{k}$. I.e., we must show that Q proves $\overline{n} + 0 = \overline{n}$. But this is immediate by Q3.

For the induction step, we assume that Q proves $\overline{n} + \overline{m} = \overline{k}$ whenever $n + m = k$ and need to show that Q proves $\overline{n} + \overline{m+1} = \overline{k}$ whenever $n + (m + 1) = k$, i.e., that Q proves $\overline{n} + S(\overline{m}) = \overline{k}$. Now $k = n + m + 1 \neq 0$, so $\overline{k}$ is $S(\overline{n+m})$, and our goal is to prove $\overline{n} + S(\overline{m}) = S(\overline{n+m})$. But $\overline{n} + S(\overline{m}) = S(\overline{n} + \overline{m})$, by Q4. And by the induction hypothesis, Q proves $\overline{n} + \overline{m} = \overline{n+m}$. Hence, by logic, Q proves $S(\overline{n} + \overline{m}) = S(\overline{n+m})$, and we are done.

The proof of part (5) is similar. Part (4) follows from parts (2) and (3); part (6) follows from parts (2) and (5). Part (7) follows from part (2), making crucial use of Q8. Showing as much is the content of Exercise 15.6.

The most difficult of these is part (8), for which we again use induction on n. For the basis, then, we need to show that Q proves $\neg\overline{m} < 0$ whenever $\neg m < 0$, which of course always holds. In fact, we can show something stronger: that Q proves $\neg x < 0$, where now x is a *variable* and not just a numeral.[19] For, by Q8, $\neg x < 0$ iff $\neg\exists z(Sz + x = 0)$, i.e., iff $\forall z(Sz + x \neq 0)$. So it is enough to show that Q proves $Sz + x \neq 0$. Now either $x = 0$ or not. If so, then $Sz + 0 = Sz \neq 0$, by Q1. And if not, then, by Q7, $x = Sy$, for some y, and $Sz + x = Sz + Sy = S(Sz + y) \neq 0$, by Q4 and Q1. That is, Q proves $x = 0 \rightarrow Sz + x \neq 0$ and $x \neq 0 \rightarrow Sz + x \neq 0$, so Q proves $Sz + x \neq 0$, by logic.

For the induction step, then, suppose that Q proves $\neg\overline{m} < \overline{n}$ whenever $\neg m < n$. We need to show that Q proves $\neg\overline{m} < \overline{n+1}$ whenever $\neg m < n+1$. Now $m \neq 0$, so $m - 1 < n$, $\overline{m}$ is $S(\overline{m-1})$, and $\overline{n+1}$ is $S(\overline{n})$, so what we need to show is that Q proves $\neg S(\overline{m-1}) < S(\overline{n})$. Now, by the induction hypothesis, Q proves $\neg\overline{m-1} < \overline{n}$. So it will be enough to show that Q proves $\overline{m-1} < \overline{n} \rightarrow S(\overline{m-1}) < S(\overline{n})$. In fact, we can again establish something stronger: Q proves the generalization $x < y \rightarrow Sx < Sy$ and even the stronger $x < y \equiv Sx < Sy$. For $Sz + x = y$ iff $S(Sz + x = Sy)$, by logic, iff $Sz + Sy = Sx$, by Q4. So $x < y$ iff $\exists z(Sz + y = x)$, by Q8, iff $\exists z(Sz + Sy = Sx)$, by what we just did, iff $Sy < Sx$, by Q8 again. □

Proposition 5.3. *Q proves all true equalities and inequalities. That is, let t and u be arbitrary closed terms of the language of arithmetic (i.e., terms without variables). Then Q proves '$t = u$' if it is true, and similarly for '$t \neq u$' and '$t < u$'.*

Proof. See Exercise 15.8. □

[19] I.e., Q does not just prove each *case* of $\neg x < 0$ but actually proves the generalization. As we'll see shortly, there are plenty of cases where Q proves each case of a generalization but does not prove the generalization itself.

Note carefully here what Q is being said to prove. It follows from what has been said, for example, that, for each n and m, Q proves $\overline{n}+\overline{m} = \overline{m}+\overline{n}$. So Q is being said to prove each of infinitely many theorems. But, as already said, Q does *not* prove that addition is commutative: It does not prove $\forall x \forall y(x + y = y + x)$. It proves each case, but it does not prove the generalization. This difference, between 'each' and 'every', arises constantly in this area, and has to be kept firmly in mind.

This difference between 'each' and 'every' gives rise to a phenomenon known as 'ω-incompleteness'.

Definition 5.4. Let $\mathcal{T}$ be a theory in the language of arithmetic (or, in fact, in any theory containing '0' and 'S'). Then $\mathcal{T}$ is said to be ω-incomplete if there is a formula $A(x)$, containing just the variable 'x' free, such that, for each n, $\mathcal{T} \vdash A(\overline{n})$ but $\mathcal{T} \nvdash \forall x A(x)$.

Now, if $\mathcal{T} \nvdash \forall x A(x)$, then the theory $\mathcal{T} \cup \{\neg \forall x A(x)\}$ is consistent. Call this theory $\mathcal{U}$. Then, for each n, $\mathcal{U} \vdash A(\overline{n})$; but also $\mathcal{U} \vdash \exists x \neg A(x)$. So, even though $\mathcal{U}$ is not inconsistent, one might think there is something strange about it. We call that strangeness ω-*inconsistency*.

Definition 5.5. Let $\mathcal{T}$ be a theory in the language of arithmetic (or, in fact, in any theory containing '0' and 'S'). Then $\mathcal{T}$ is said to be ω-inconsistent if there is a formula $A(x)$, containing just the variable 'x' free, such that, for each n, $\mathcal{T} \vdash \neg A(\overline{n})$ but also $\mathcal{T} \vdash \exists x A(x)$. Otherwise, $\mathcal{T}$ is said to be ω-consistent.

Note that any theory that is ω-consistent is also consistent.

We will also want to talk about two other arithmetical theories.

Peano Arithmetic, or PA, is the theory containing (1)–(6) and (8) from Q and all instances of the induction scheme:

$$A(0) \wedge \forall x(A(x) \rightarrow A(Sx)) \rightarrow \forall x A(x)$$

where $A(x)$ is a formula of the language of arithmetic containing x free (and not bound). Obviously, there are infinitely many such axioms. And the standard interpretation of the language of arithmetic is a model of PA: All of the axioms of PA are true in it.

Induction allows PA to prove many things that Q cannot prove.

Example 5.6. PA proves $\forall x(x \neq Sx)$.

Proof. Let $A(x)$ be the formula $x \neq Sx$. So the following is an induction axiom:

$$0 \neq S0 \wedge \forall x(x \neq Sx \rightarrow Sx \neq SSx) \rightarrow \forall x(x \neq Sx)$$

But $0 \neq S0$ follows immediately from **Q1**, and $\forall x(x \neq Sx \rightarrow Sx \neq SSx)$ follows just as immediately from **Q2**. □

Example 5.7. PA proves $0 + x = x$.

Proof. This time the relevant induction axiom is:

$$0 + 0 = 0 \wedge \forall x(0 + x = x \rightarrow 0 + Sx = Sx) \rightarrow \forall x(0 + x = x)$$

The first conjunct follows from Q3. For the second, suppose $0 + x = x$. Then $S(0 + x) = Sx$, by logic; and by Q4, $S(0 + x) = 0 + Sx$. So $0 + Sx = Sx$. □

Note that it now follows, by Q1, that PA proves $0 + x = x + 0$. This can then be used as the basis for an inductive proof, in PA, that $y + x = x + y$, i.e., a proof of the commutativity of addition. We'll leave that proof to the adventurous reader. It can further be shown in this way that PA proves all the basic arithmetical facts, e.g., the associativity of multiplication and the distribution laws.

One might wonder why we did not include Q7 among the axioms of PA. The answer is that it would have been redundant.

Proposition 5.8. *Q7 is provable in* PA.

Proof. This is Exercise 15.9. □

We shall also have occasion to discuss a theory known just as *Arithmetic* (or, sometimes, 'True Arithmetic'). This is the theory containing all truths of the language arithmetic—that is, all formulae in the language of arithmetic that are true in the standard interpretation. Arithmetic is a complete, closed theory. It is complete, because every formula in its language is either true or false in the standard interpretation; so for any such formula A, either A is in Arithmetic or its negation is. It is closed because, if A is derivable from Arithmetic, it is true in every model of Arithmetic. But since the standard interpretation *is* a model of Arithmetic, it follows that A must be true in the standard interpretation, and so it is in Arithmetic.

6. Representability

PA is sufficient to prove many facts about numbers. Consider, for example, the claim that $\forall x(1 < x \rightarrow x < x^2)$. Can that be proven in PA? One might worry that the symbol for *squared* is not part of the language of arithmetic, and one would be right to raise this issue. But it is no real obstacle, for there is an obvious way to define 'x^2'—as '$x \times x$'—whence we can raise the question whether PA proves '$\forall x(1 < x \rightarrow x < x^2)$'. By the obvious induction, it does.

Consider now the question whether PA can prove that $\forall x(x < 2^x)$. The natural idea would again be to define exponentiation in the language of arithmetic, that is, to find some term $\varphi(x)$ with just 'x' free that defines 2^x—defines it, in the sense that the value of $\varphi(x)$, when 'x' is assigned the number n, is 2^n. That, unfortunately, turns out to be impossible, as the following two results show.

Proposition 6.1. *Let $\varphi(x)$ be a term in the language of arithmetic with at most 'x' free. Then $\varphi(x)$ is equivalent to a polynomial in 'x', that is, to something of the form: $a_0x^n + a_1x^{n-1} + \cdots + a_n$.*

Proof. Induction on the complexity of expressions. Obviously, the basic terms '0' and 'x' are equivalent to polynomials. And if t and u are, then clearly $\ulcorner St \urcorner$, $\ulcorner t + u \urcorner$, and $\ulcorner t \times u \urcorner$ are as well. □

Proposition 6.2. *Let $\varphi(x)$ be a polynomial. Then we can find a y such that, if $z > y$, then $2^z > \varphi(z)$.*

We can put this by saying that exponentiation grows faster than any polynomial. A careful proof actually shows how to calculate such a y, given a specific polynomial $\varphi(x)$.[20]

It follows from the last two propositions that, if $\varphi(x)$ is a term in the language of arithmetic containing just 'x' free, then, for some n, the value of '$\varphi(x)$' when x is assigned n is less than, and so not equal to, 2^n. So no term of the language of arithmetic defines exponentiation.

There is, however, a way around this problem, for there is a *formula* $\textsc{Exp}(x, y)$ that defines exponentiation in a closely related sense: $\textsc{Exp}(x, y)$ holds in the standard interpretation when 'x' is assigned n and 'y' is assigned m if, and only if, $m = 2^n$. So $\textsc{Exp}(x, y)$ defines not the exponentiation *function* but the *graph* of that function, which is the *relation* $y = 2^x$, and that is almost as good. Moreover, $\textsc{Exp}(x, y)$ is, as Frege liked to put it, provably a 'many-one' relation: It can be proven in PA to be 'function-like', in the sense that

$$\forall x \exists y(\textsc{Exp}(x, y) \wedge \forall z(\textsc{Exp}(x, z) \rightarrow y = z))$$

can be proven in PA.[21] So we can think of PA as proving that $x < 2^x$ if it proves

$$\forall x \exists y(\textsc{Exp}(x, y) \wedge x < y)$$

for what that says, in effect, is that, for each x, there is a y, such that $y = 2^x$ and $x < y$. And given an appropriate definition of $\textsc{Exp}(x, y)$, PA does prove the mentioned formula.

These reflections motivate the following definition.

Definition 6.3. A formula $\Phi(x_1, \ldots, x_n, y)$ *defines* the n-place function $\varphi(x_1, \ldots, x_n)$ in the language of arithmetic if, and only if,

(i) $\Phi(\overline{k}_1, \ldots, \overline{k}_n, \overline{m})$ is true[22] iff $\varphi(k_1, \ldots, k_n) = m$

[20] In an earlier version of this note, I called this an 'easy exercise in number theory'. It turns out it is not so easy, though it is not terribly difficult, either.

[21] Strictly speaking, of course, PA does not prove the mentioned formula, since 'EXP' is not in the language of arithmetic. What PA proves, strictly speaking, is $\ulcorner \forall x \exists y(\textsc{Exp}(x, y) \wedge \forall z(\textsc{Exp}(x, z) \rightarrow y = z)) \urcorner$, the result of substituting for the various occurrences of '$\textsc{Exp}(\xi, \zeta)$' in the mentioned formula the formula of the language of arithmetic that it abbreviates.

[22] Truth here is, of course, truth in the standard interpretation.

(ii) $\forall x \forall y \forall z[\Phi(x_1, \ldots, x_n, y) \wedge \Phi(x_1, \ldots, x_n, z) \rightarrow y = z]$ is true

We say that a function is *definable* in the language of arithmetic if there is a formula that defines it.

Note that the notion of definability is *semantic*: It is given in terms of the notion of *truth*.[23] What we need here is the corresponding *syntactic* notion: one that stands to definability much as the notion of derivation stands to that of implication.

Definition 6.4. A formula $\Phi(x_1, \ldots, x_n, y)$ *represents* the n-place function $\varphi(x_1, \ldots, x_n)$ in the theory Σ if, and only if, whenever $\varphi(k_1, \ldots, k_n) = m$:

(i) $\Sigma \vdash \Phi(\overline{k}_1, \ldots, \overline{k}_n, \overline{m})$

(ii) $\Sigma \vdash \forall y(\Phi(\overline{k}_1, \ldots, \overline{k}_n, y) \rightarrow y = \overline{m})$

or, equivalently:

$$\Sigma \vdash \forall y(\Phi(\overline{k}_1, \ldots, \overline{k}_n, y) \equiv y = \overline{m})$$

We say that a function is *representable* in Σ if there is a formula that represents it in Σ.

Note that the definition of representation requires only that Σ prove that φ has the value it does *in each specific case* and also that it prove that φ has only one value *in each specific case*. It is *not* required that Σ prove the generalization:

$$\forall x_1 \ldots \forall x_n[\exists y(\Phi(x_1, \ldots, x_n, y) \wedge \forall z(\Phi(x_1, \ldots, x_n, z) \rightarrow y = z)]$$

It is, in fact, important that we omit this stronger condition. As we shall see below, it is true that exponentiation is representable in Q, that is, that there is a formula $\textsc{Exp}(x, y)$ such that, whenever $n = 2^m$, Q proves:

(i) $\textsc{Exp}(\overline{m}, \overline{n})$

(ii) $\forall y(\textsc{Exp}(\overline{m}, y) \rightarrow y = \overline{n})$

But it is an important fact about Q—one that is, again, a reflection of its weakness—that it does *not* prove that exponentiation always has a unique value, and it does not prove that exponentiation is total, either:

$$\mathsf{Q} \nvdash \forall x \forall y \forall z(\textsc{Exp}(x, y) \wedge \textsc{Exp}(x, z) \rightarrow y = z)$$

$$\mathsf{Q} \nvdash \forall x \exists y(\textsc{Exp}(x, y))$$

PA, however, proves both of these facts, by the obvious sort of induction.

Proposition 6.5. *Suppose* $\Phi(x_1, \ldots, x_n, y)$ *represents a total function* $\varphi(x_1, \ldots, x_n)$ *in* Σ *and, whenever* $m \neq n$, Σ *proves* $\ulcorner \overline{m} \neq \overline{n} \urcorner$. *Then whenever* $\phi(k_1, \ldots, k_n, y) \neq m$, $\Sigma \vdash \neg\Phi(\overline{k}_1, \ldots, \overline{k}_n, \overline{m})$.

[23] The contrast between representability and definability here is thus not the same as the contrast between these notions in Boolos and Jeffrey (1989). There, definability is just the analogue of representability for sets and relations. Here, on the other hand, definability is a semantic notion; representability, a syntactic one. (My terminology is thus closer to Tarski's.) The reason for this change will become clear below.

Proof. Fix the k_i and suppose that $\varphi(k_1, \ldots, k_n) \neq m$. Since ϕ is total, for some $n \neq m$, $\varphi(k_1, \ldots, k_n) = n$ and so, since $\Phi(x_1, \ldots, x_n, y)$ represents $\varphi(x_1, \ldots, x_n)$ in Σ, Σ proves $\forall y(\Phi(\overline{k}_1, \ldots, \overline{k}_n, y) \rightarrow y = \overline{n})$. So, by logic, Σ proves $\Phi(\overline{k}_1, \ldots, \overline{k}_n, \overline{m}) \rightarrow \overline{m} = \overline{n}$. But now, by hypothesis, Σ proves $\ulcorner \overline{m} \neq \overline{n} \urcorner$, so it also proves $\neg\Phi(\overline{k}_1, \ldots, \overline{k}_n, \overline{m})$. □

Corollary 6.6. *Suppose that $\Phi(x_1, \ldots, x_n, y)$ represents a total function $\varphi(x_1, \ldots, x_n)$ in* Q. *Then, if $\varphi(k_1, \ldots, k_n) \neq m$,* Q $\vdash \neg\Phi(\overline{k}_1, \ldots, \overline{k}_n, \overline{m})$.

Proof. Immediate from 5.2 and the preceding proposition. □

Similar notions can be defined for sets—or, more generally, relations—rather than functions.

Definition. We say that a formula $\Phi(x_1, \ldots, x_n)$ *defines* the n-place relation $F(x_1, \ldots, x_n)$ if $\Phi(\overline{k}_1, \ldots, \overline{k}_n)$ is true (in the standard interpretation) iff $F(k_1, \ldots, k_n)$. We say that a relation is *definable* in the language of arithmetic if there is a formula that defines it.

Equivalently: $\Phi(x_1, \ldots, x_n)$ defines the relation $F(x_1, \ldots, x_n)$ if the extension of $\Phi(x_1, \ldots, x_n)$ in the standard interpretation is the set $\{< k_1, \ldots, k_n >: F(k_1, \ldots, k_n)\}$.

Definition. We say that a formula $\Phi(x_1, \ldots, x_n)$ *represents* the n-place relation $F(x_1, \ldots, x_n)$ in the theory Σ iff:

(i) whenever $F(k_1, \ldots, k_n)$, $\Sigma \vdash \Phi(\overline{k}_1, \ldots, \overline{k}_n)$

(ii) whenever $\neg F(k_1, \ldots, k_n)$, $\Sigma \vdash \neg\Phi(\overline{k}_1, \ldots, \overline{k}_n)$

We say that a relation is *representable* in Σ if there is a formula that represents it in Σ.

These definitions are reconciled with those for functions through the notion of a *characteristic function*.

Definition. Let s be a set. Its *characteristic function* is the function $\varphi_s(x)$ whose value is 1 if $x \in s$ and 0 if $x \notin s$. Similarly, if r is an n-place relation, its characteristic function is the n-place function $\varphi_r(x_1, \ldots, x_n)$ whose value, for arguments $k_1, \ldots, k_n$ is 1 if $r(k_1, \ldots, k_n)$ and 0 otherwise.

Proposition 6.7.

(1) A set or relation is definable in the language of arithmetic iff its characteristic function is definable in the language of arithmetic.

(2) So long as $\Sigma \vdash \overline{0} \neq \overline{1}$, a set or relation is representable in Σ iff its characteristic function is representable in Σ.

Proof. This is Exercise 15.12. □

As noted above, we aim to use the notion of representability to overcome the fact that the language of arithmetic is "term poor". As also noted, however, a theory can represent a function without proving that the function so represented really is a function. For example, Q does

not prove that exponentiation is either many-one or total. If a theory Σ *does* prove that exponentiation is both many-one and total, however—that is, if Σ proves both $\forall x[\exists y(\textsc{Exp}(x,y) \wedge \forall z(\textsc{Exp}(x,z) \to y = z)]$ and $\forall x \exists y(\textsc{Exp}(x,y))$—then it is very much as if a function-symbol for exponentiation is present in the language. The following result helps make it clear in what sense that is so.

Theorem 6.8. *Suppose that the function $\phi(x)$ is represented in Σ by the formula $F(x,y)$. Suppose further that Σ proves:*

$$\forall x \forall y \forall z(F(x,y) \wedge F(x,z) \to y = z)$$
$$\forall x \exists y(F(x,y))$$

Expand the language of Σ by adding a new function symbol $f(x)$ and let Σ_f be Σ plus the new axiom:

$$f(x) = y \equiv F(x,y)$$

Then Σ_f is a conservative extension of Σ.

Proof. This is exercise 15.13. □

7. Recursiveness and Representability in Q

We are going to need to know, below, that certain functions and relations are indeed representable in Q. One way to prove that they are is, obviously, to exhibit formulae that represent them: That, in fact, is how Gödel proceeds in his proof of the incompleteness theorem. We, however, shall take a shortcut.

Certain functions are, in an intuitive sense, *computable* or *calculable*, in the sense that there is a method, algorithm, or procedure by means of which one can in principle determine, given some arguments for a function, what its value is for those arguments. For example, addition and multiplication are computable, in this sense: We all learned the relevant algorithms in elementary school. Properties of natural numbers, and relations among them, may, in the same sense, be computable. The relation: x divides y (without remainder) is intuitively computable. Long division is an algorithm by means of which one can decide whether one number divides another. It is, by the way, also an algorithm by means of which one can calculate the remainder upon dividing y by x, and so remainder is also computable.

There is a developed mathematical theory of computability, *recursion theory*, which was founded by Alonzo Church, Alan Turing, Kurt Gödel, and others in the 1930s.[24] This theory provides a definition of the notion of a 'recursive function' that makes the intuitive notion of a computable function rigorous. Indeed, there are many such definitions of the set of

[24] See Martin Davis, *The Undecidable*, for some of the early papers in the subject.

recursive functions, all of which are provably equivalent. That gives us reason to suppose that recursive functions form a 'natural kind'.

Church famously conjectured that the recursive functions are exactly the computable ones, a claim known as *Church's Thesis*. It is widely, but not universally, held that one cannot prove Church's Thesis rigorously: To do so, one might think, one would need a mathematical definition of the set of computable functions, and that is precisely what the notion of recursive function is supposed to be.[25] No one, however, has ever exhibited a function that is computable in the intuitive sense but that is not recursive. Turing, for his part, offered a philosophical argument for Church's Thesis, namely, that his definition of the class of recursive functions constitutes a *philosophical analysis* of the intuitive notion of a computable function.

In any event, Church's Thesis is now very widely accepted. We shall assume it here. For our purposes, then, one can show a function to be recursive by exhibiting (or, more often, alluding to) an algorithm that computes it. So x divides y, for example, is recursive, since you can compute it by long division. And so forth. Functions whose arguments and values are not natural numbers can be computable, or recursive, too, in much the same sense. We shall accept Church's Thesis for them, too. So, for example, the syntactic function: the negation of the expression x, is recursive: To compute it, once just writes down '$\neg$' followed by a left parenthesis, the expression x, and a right parenthesis, and then one stares at the result.

As we saw above, Q is, in many respects, a very weak theory. We are now ready to see that, in another respect, Q is quite powerful.

Theorem 7.1. *Every recursive function is representable in* Q.

This was originally proven by Tarski, Mostowski, and Robinson (1953). For a modern proof, see *Computability and Logic* (Boolos et al., 2007).

Corollary 7.2. *Every computable function is representable in* Q.

Proof. By Church's Thesis. □

Corollary 7.3. *Every computable set and relation is representable in* Q.

Proof. By Proposition 6.7, a set s is representable in Σ iff its characteristic function φ_s is representable in Σ. But, if s is computable, then so is φ_s, for one can compute $\varphi_s(n)$ simply by computing whether $n \in S$ and then returning the answer: 1, if it is, and 0, if it is not. □

So it suffices, for our purposes, to show that a function or set or relation is representable in Q to show that it is computable, that is, to exhibit (or allude to) an algorithm that computes it.

[25] It has been argued by Peter Smith, by contrast, that one *can* prove Church's thesis, using what Georg Kreisel called a 'squeezing' argument.

Now, if a function is representable in Q, then it is representable in any theory that extends Q, since representability only requires that certain things should be theorems of the theory in question. So we have:

Corollary 7.4. *Every recursive function is representable in every theory* Σ *that extends* Q*, in particular, in* PA*. Hence, so is every computable relation.*

8. Gödel Numbering

Computers crunch numbers. Yet they can also be used as word processors. How does that work? Well, very simply. Letters and words and sentences get represented, in the computer, by numbers. Perhaps the most common representation uses something called the American Standard Code for Information Interchange, or ASCII, which defines a correspondence between Roman letters (and some other symbols) and numbers.[26] The capital 'A', for example, is represented as 65; 'B', as 66; and so forth; the small 'a' is represented as 97; 'b', as 98, and so forth. The word 'Abba' might then be represented as: 65989897, just stringing the digits together.

This (very important!) idea is due to Gödel, and it plays a central role in his proof of the incompleteness theorem.

To apply this idea to the language of arithmetic, we first establish a correspondence between the primitive symbols of the language of arithmetic and the hexadecimal (base-16) digits:

(	)	$\exists$	$\forall$	$\vee$	$\wedge$	$\rightarrow$	$\neg$	x	$'$	0	S	$+$	$\times$	$=$
1	2	3	4	5	6	7	8	9	a	b	c	d	e	f

We extend the correspondence to one between strings of symbols of the language of arithmetic and hexadecimal numerals in the obvious way, basically, by treating these symbols as if they just *were* the corresponding hexadecimal digits. Thus, for example, we read the string '$) + 0\exists\vee$' as if it just were the hexadecimal numeral '$2db35$'. This induces a correlation between strings of symbols and natural numbers. The one-element string '0' is thus correlated with the number b_{16}, or 11; the string '$) + 0\exists\vee$' is correlated with the number $2db35_{16}$, or $2 \times 16^4 + 13 \times 16^3 + 11 \times 16^2 + 3 \times 16^1 + 5 \times 16^0$, or $187,189$. Every string of symbols is thus correlated with its own unique natural number, called its *Gödel number*.

For most purposes, the specific Gödel numbering we employ does not matter very much. What certainly does matter is that everything should be computable: It should be computable what the Gödel number of a given string is; conversely, it needs to be computable whether a given number is the Gödel number of a string and, if so, of which one. Note that our Gödel numbering is computable, in this sense. To calculate the Gödel

[26] A more modern, and multi-cultural, standard is Unicode, of which the ASCII codes are all part.

number of a string, one just reads it as if it were a hexadecimal numeral. To decide whether a number is the Gödel number of a string and, if so, which one, first calculate its hexadecimal representation. The number is the Gödel number of a string if, and only if, that representation contains no (non-leading) zeros. If it does not, then just reverse the correspondence to figure out which string the number represents.

Gödel numbering allows us to talk about syntax—about properties of strings, or sequences thereof, and operations on them—within the language of arithmetic. Consider, for example, the function whose value, for a given string, is the result of enclosing that string in parentheses. Thus, $\mathsf{paren}[\text{`}) + 0 \exists \vee \text{'}]$ is just '$() + 0 \exists \vee$)'. Given our Gödel numbering, there is a corresponding numerical function that takes the Gödel number of a string and returns the Gödel number of the result of enclosing that string in parentheses. Call this function $\textsc{paren}[x]$. Then $\textsc{paren}[2db35_{16}] = 12db352_{16}$, since Gödel number of ')+0$\exists\vee$' is $2db35_{16}$ and the Gödel number of '()+0$\exists \vee$)' is $12db352_{16}$. More generally:

$$\begin{aligned}\textsc{paren}[x] = 0&, \text{ if } x \text{ has (non-leading) zeros in its}\\ &\quad\text{hexadecimal representation;}\\ = 1 \times 16^{k+1} + 16 \times x + 2&\text{ otherwise, where } k \text{ is}\\ &\quad\text{the least } n \text{ such that } 16^n > x\end{aligned}$$

Here 0 is a 'throwaway' value, provided so that $\textsc{paren}[x]$ is a total function. Since $\textsc{paren}[x]$ is clearly computable, it is representable in Q. So there is a formula $\textsc{Paren}[x, y]$ such that, whenever $n = \textsc{paren}[m]$, Q proves: $\textsc{Paren}[\overline{m}, \overline{n}]$. Thus, Q proves $\textsc{Paren}[\overline{2db35_{16}}, \overline{12db352_{16}}]$. Or, as we may also put it: Q proves that the result of enclosing ')+0$\exists\vee$' in parentheses is '()+0$\exists \vee$)'.

Generalizing the above, we have the following.

Lemma 8.1. *Let $\varphi(s_1, \ldots, s_n)$ be a computable n-place function from strings to strings. Then there is a formula $\Phi(x_1, \ldots, x_n, y)$ of the language of arithmetic that represents it in* Q*, in the sense that, whenever $\varphi(s_1, \ldots, s_n) = t$ and $\sigma_1, \ldots, \sigma_n$ and τ are the Gödel numbers of $s_1, \ldots, s_n$ and t, respectively:*

(i) $\quad \mathsf{Q} \vdash \Phi(\overline{\sigma_1}, \ldots, \overline{\sigma_n}, \overline{\tau})$
(ii) $\quad \mathsf{Q} \vdash \forall x(\Phi(\overline{\sigma_1}, \ldots, \overline{\sigma_n}, x) \to x = \overline{\tau})$

Moreover, the same holds if $\varphi(s_1, \ldots, s_n)$ is a computable n-place function from strings to numbers. I.e, whenever $\varphi(s_1, \ldots, s_n) = m$ and $\sigma_1, \ldots, \sigma_n$ are the Gödel numbers of $s_1, \ldots, s_n$:

(i) $\quad \mathsf{Q} \vdash \Phi(\overline{\sigma_1}, \ldots, \overline{\sigma_n}, \overline{m})$
(ii) $\quad \mathsf{Q} \vdash \forall x(\Phi(\overline{\sigma_1}, \ldots, \overline{\sigma_n}, x) \to x = \overline{m})$

Note that the Lemma has been stated, and will be proved, in such a way that any it holds for all computable Gödel numberings.

Proof. We prove just the first part. (The second is similar.) Let φ be as in the statement of the Lemma. We define a numerical function φ^* as follows:

$$\begin{aligned}\varphi^*(x_1, \ldots, x_n) &= \text{the Gödel number of } \varphi(s_1, \ldots, s_n), \text{ if each} \\ &\qquad x_i \text{ is the Gödel number of a string } s_i \\ &= 0, \text{ otherwise}\end{aligned}$$

The definition of φ^* makes it computable: To calculate $\varphi^*(x_1, \ldots, x_n)$, one need only follow these steps: First, determine which string s_i each of the x_i is the Gödel number of, if any (there is a method for doing that, since the Gödel numbering is computable); return 0 if one of the x_i isn't the Gödel number of a string; otherwise, calculate $\varphi(s_1, \ldots, s_n)$ and then determine its Gödel number (there are methods for doing each of those things, since φ is computable and so is the Gödel numbering); hence, φ^* is computable and so is representable in Q. So there is a formula $\Phi(x_1, \ldots, x_n, y)$ such that, whenever $\varphi^*(\sigma_1, \ldots, \sigma_n) = \tau$:

(i) $\mathsf{Q} \vdash \Phi(\overline{\sigma_1}, \ldots, \overline{\sigma_n}, \overline{\tau})$

(ii) $\mathsf{Q} \vdash \forall x(\Phi(\overline{\sigma_1}, \ldots, \overline{\sigma_n}, , x) \rightarrow x = \overline{\tau})$

But if, as in the statement of the lemma, $\varphi(s_1, \ldots, s_n) = t$ and $\sigma_1, \ldots, \sigma_n$ and τ are the Gödel numbers of $s_1, \ldots, s_n$ and t, respectively, then we do indeed have that $\varphi^*(\sigma_1, \ldots, \sigma_n) = \tau$, by the definition of φ^*. □

Corollary 8.2. *Let $\varphi(s_1, \ldots, s_n)$ be a computable n-place relation on strings. Then there is a formula $\Phi(x_1, \ldots, x_n)$ of the language of arithmetic that represents it in* Q*, in the sense that, where $\sigma_1, \ldots, \sigma_n$ are the Gödel numbers of $s_1, \ldots, s_n$:*

(i) whenever $\varphi(s_1, \ldots, s_n)$, $\mathsf{Q} \vdash \Phi(\overline{\sigma_1}, \ldots, \overline{\sigma_n})$

(ii) whenever $\neg\varphi(s_1, \ldots, s_n)$, $\mathsf{Q} \vdash \neg\Phi(\overline{\sigma_1}, \ldots, \overline{\sigma_n})$

Proof. This is exercise 15.14. □

Many syntactic notions are now easily shown to be representable in Q by appeal to the preceding results. Thus, it is obviously computable whether a string of symbols of the language of arithmetic is a well-formed formula. Hence, there is a formula $\textsc{Wff}(x)$ of the language of arithmetic that represents 'x is a well-formed formula'. Similarly, the relation 'z is a conditional whose antecedent is x and consequent is y' is obviously computable and hence representable in Q. And so on and so forth.

9. Formal Theories

Definition 9.1. A *formal theory* is a theory—that is, a set of formulae—that is recursive. The formulae in a formal theory are called its *axioms*.

Every finite theory is obviously a formal theory. PA is, too: There is an obvious algorithm by means of which one can decide, or compute, whether an arbitrary formula is an axiom of PA. As we shall see below, however,

it follows from Gödel's first incompleteness theorem that Arithmetic is *not* a formal theory.

Definition 9.2. A theory Σ is said to be *axiomatizable* if there is a formal theory Θ in the same language such that, for every formula A in their common language: $\Sigma \vdash A$ iff $\Theta \vdash A$.

Σ is *finitely* axiomatizable if there is a finite theory Θ meeting the same condition.

There is an equivalent statement of this definition: Given a theory Σ, define its *closure* as follows:

Definition 9.3. The closure of a theory Σ—written: $\mathsf{Cl}(\Sigma)$—is the set of its theorems, that is, of the formulae derivable from it. I.e.: $\mathsf{Cl}(\Sigma) = \{A : \Sigma \vdash A\}$.

We can easily prove that $\mathsf{Cl}(\Sigma)$ is always closed (see Exercise 15.15), thus justifying the terminology, and, moreover, that it is the smallest closed theory containing Σ itself. The equivalent definition of axiomatizability is then:

Definition 9.4. Σ is axiomatizable if there is a formal theory Θ such that $\mathsf{Cl}(\Sigma) = \mathsf{Cl}(\Theta)$. Similarly for finite axiomatizability.

10. The Diagonal Lemma

We now prove the so-called diagonal lemma, which is the last piece we need before we can prove the first incompleteness theorem.

First, a piece of notation. We shall use: $\ulcorner G \urcorner$, to mean: the numeral for the Gödel number of the expression *G*. We shall also use it to mean: the Gödel number of G, leaving it to context to disambiguate.

Lemma 10.1 (Diagonal Lemma)**.** *Let Σ be a theory that represents every recursive function, and let $A(x)$ be a formula in the language of Σ with just 'x' free. Then there is a sentence G such that $\Sigma \vdash G \equiv A(\ulcorner G \urcorner)$.*

Thus, G (for Gödel) is a sentence that 'says of itself' that it has the property that $A(x)$ expresses. In this sense, the existence of self-referential sentences is mathematically provable: It is, more precisely, guaranteed by the axioms of Q, since Q satisfies the hypothesis of the diagonal lemma.

We shall prove the lemma two ways. We begin, however, with some motivational remarks.

Consider the following open term (i.e., functional expression):

> the result of substituting the quotation name of x for the sole free variable contained in it

If we apply this function to 'frogs eat x', then we get:

> frogs eat 'frogs eat *x*'

which is false. (Frogs don't eat formulas!) If we apply it to 'x contains 16 symbols, including blanks', then the result is:

> 'x contains 16 symbols, including blanks' contains 16 symbols, including blanks

which is also false. And if we apply it to 'x contains the word "the"', we get:

> 'x contains the word "the"' contains the word "the"

which is obviously true.

Consider now what happens if we apply the function to 'the result of substituting the quotation name of x for the sole free variable contained in it is very strange'. The result is:

> the result of substituting the quotation name of 'the result of substituting the quotation name of x for the sole free variable contained in it is very strange' for the sole free variable contained in it is very strange

Call this sentence Buster. Then what we have just seen—proven, really, by calculation—is that Buster is the result of substituting the quotation name of 'the result of substituting the quotation name of x for the free variable contained in it is very strange' for the sole free variable contained in it. I.e., Buster is the sentence named by the subject phrase of Buster itself (everything except its last three words). So, by the laws of identity, Buster is equivalent to the sentence: Buster is very strange. In that sense, then, Buster 'says of itself' that it is very strange. Which, indeed, it is.

The formal proof we shall give in a bit is just a formalization of this informal argument. It is, however, in some ways more complicated than it really ought to be. The complications are the result of the fact that, as we saw above, the language of arithmetic is rather lacking in terms. If we ignore that and pretend there are a lot of terms, we can give a much simpler proof that makes the *point* of the argument a lot clearer.

Proof. (*Somewhat dishonest variety*)[27] Consider the following syntactic operation: Given a formula $A(x)$, with just 'x' free, its 'pseusodiagonalization' is the result of substituting the numeral for the Gödel number of $A(x)$ for 'x' in $A(x)$. That is, the pseudo-diagonalization of $A(x)$ is: $A(\ulcorner A(x) \urcorner)$.[28] This operation is obviously recursive, so (we will

[27] There are ways one can make this argument legitimate, if Σ is strong enough. PA is 'strong enough'. So is $I\Sigma_1$, which is Q with induction for Σ_1 sentences. See *The Logic of Provability* (Boolos, 1993), from which this proof is adapted, for the details. Moreover, there are systems, such as primitive recursive arithmetic, in which we really do have a function symbol such as $\textsc{Pdiag}(x)$, and in such systems the proof can stand as is.

[28] If the argument isn't a formula with just 'x' free, then it doesn't matter what the value is. For definiteness, just take it to be the same as the argument.

pretend that) there is an open term, $\text{PDIAG}(x)$, that 'represents' pseudo-diagonalization in Σ, in the sense that, whenever n is indeed the Gödel number of the pseudo-diagonalization of the expression whose Gödel number is m, Σ proves $\text{PDIAG}(\overline{m}) = \overline{n}$.[29] Now consider the expression:[30]

$$(P) \qquad A(\text{PDIAG}(x))$$

which has only the variable 'x' free. Its pseudo-diagonalization is then:

$$(G) \qquad A(\text{PDIAG}(\ulcorner A(\text{PDIAG}(x))\urcorner))$$

which contains no free variables[31] and so is a sentence. So $\ulcorner G\urcorner$ is the Gödel number of the formula which is the pseudo-diagonalization of the formula whose Gödel number is $\ulcorner A(\text{PDIAG}(x)\urcorner$. Hence:

$$\Sigma \vdash \text{PDIAG}(\ulcorner A(\text{PDIAG}(x))\urcorner) = \ulcorner G\urcorner$$

and so, by the laws of identity:

$$\Sigma \vdash A(\text{PDIAG}(\ulcorner A(\text{PDIAG}(x))\urcorner)) \equiv A(\ulcorner G\urcorner)$$

But the sentence on the left-hand side here just is G! So

$$\Sigma \vdash G \equiv A(\ulcorner G\urcorner)$$

and we are done. □

The 'real' proof of the diagonal lemma we shall now give just recapitulates this argument, but it makes use of a *formula* '$\text{DIAG}(x, y)$' instead of the pretend term '$\text{PDIAG}(x)$'. As said earlier, this sort of complication is the price we pay for the language of arithmetic's poverty of terms. We first need a definition.

Definition. Let A be an expression. Its *diagonalization* is the expression: $\exists x(x = \ulcorner A\urcorner \wedge A)$, where $\ulcorner A\urcorner$ is, as before, the numeral for the Gödel number of A.

The way we have defined diagonalization, the expression A need not be a formula. If it isn't, then its diagonalization will not be a formula either. But if A *is* a formula, then its diagonalization will also be a formula, and, if A has only the variable 'x' free, then its diagonalization will be a sentence, that is, it will not have any variables free, since the free occurrences of 'x' will all be bound by the initial existential quantifier.

All we *really* need to know for the proof of the diagonal lemma is that diagonalization is representable in Σ. So what we will actually prove is

Lemma 10.2 (Diagonal Lemma)**.** *Let Σ be a theory that represents diagonalization. Then, for each formula $A(x)$ with just 'x' free, there is a sentence G such that $\Sigma \vdash G \equiv A(\ulcorner G\urcorner)$.*

[29] Note that this incorporates both of the usual conditions on representability.

[30] Note that '$A(\text{PDIAG}(x))$' is the result of substituting $\text{PDIAG}(x)$ for all free occurrences of 'x' in $A(x)$.

[31] Note that 'x' does not occur in (G). It only appears within quotation marks.

Proof. Since diagonalization is representable in Σ, there is a formula $\textsc{Diag}(x, y)$ of the language of Σ such that, whenever n is the Gödel number of the diagonalization of the expression whose Gödel number is m, $\Sigma \vdash \forall x(\textsc{Diag}(\overline{m}, x) \equiv x = \overline{n})$.[32] Now, consider the expression:

$$(P) \qquad \exists y(\textsc{Diag}(x, y) \wedge A(y))$$

which has only the variable 'x' free. Its diagonalization is then:

$$(G) \qquad \exists x[x = \ulcorner P \urcorner \wedge \exists y(\textsc{Diag}(x, y) \wedge A(y))]$$

Note that G is a sentence.

Since G is logically equivalent to:

$$\exists y[\textsc{Diag}(\ulcorner P \urcorner, y) \wedge A(y)]$$

Σ proves that it is:

$$(1) \qquad \exists x[x = \ulcorner P \urcorner \wedge \exists y(\textsc{Diag}(x, y) \wedge A(y))] \equiv \exists y[\textsc{Diag}(\ulcorner P \urcorner, y) \wedge A(y)]$$

Further, G, as we said, is the diagonalization of P. So, obviously, $\ulcorner G \urcorner$ is the Gödel number of the diagonalization of the formula whose Gödel number is $\ulcorner P \urcorner$. Hence Σ proves:

$$(2) \qquad \forall x[\textsc{Diag}(\ulcorner P \urcorner, x) \equiv x = \ulcorner G \urcorner]$$

From (2), Σ will prove, by logic:

$$\exists y[\textsc{Diag}(\ulcorner P \urcorner, y) \wedge A(y)] \equiv \exists y[y = \ulcorner G \urcorner \wedge A(y)]$$

but the right-hand side is obviously equivalent to '$A(\ulcorner G \urcorner)$', so Σ also proves:

$$(3) \qquad \exists y[\textsc{Diag}(\ulcorner P \urcorner, y) \wedge A(y)] \equiv A(\ulcorner G \urcorner)$$

And now, from (1) and (3), Σ proves:

$$(4) \qquad \exists x[x = \ulcorner P \urcorner \wedge \exists y(\textsc{Diag}(x, y) \wedge A(y))] \equiv A(\ulcorner G \urcorner)$$

But the left-hand side here just is G! So (4) just is:

$$G \equiv A(\ulcorner G \urcorner)$$

as wanted. □

The diagonal lemma, as we originally stated it in Lemma 10.1, is now easily provable. Since diagonalization is clearly computable, it is representable in any theory that represents all computable functions. Any such theory therefore satisfies the hypothesis of Lemma 10.2.

The diagonal lemma can be generalized in two ways: We can allow other free variables to occur in A, which just get carried along; and we can apply the construction simultaneously to a number of formulae. The fully generalized version is thus:

[32] It is worth emphasizing that $\textsc{Diag}(x, y)$ is a formula that can actually be written down. George Boolos told me that he once had a graduate student write it out in primitive notation. It was about six pages long.

Lemma 10.3 (Generalized Diagonal Lemma)**.** *Suppose* Σ *represents every recursive function. For each* $i = 1, \ldots n$*, let* $A_i(x_1, \ldots x_n, y_1, \ldots, y_m)$ *be a formula in the language of* Σ *with at most the displayed variables free and in which* x_i *is free in* A_i*. Then there are formulae* $G_1, \ldots, G_n$ *such that, for each i,*

$$\Sigma \vdash G_i(y_1, \ldots, y_m) \equiv A_i(\ulcorner G_1 \urcorner, \ldots, \ulcorner G_n \urcorner, y_1, \ldots, y_m)$$

The proof is a generalization of that of the diagonal lemma and is left as 15.16.

11. Gödel's First Incompleteness Theorem

We can now prove Gödel's first incompleteness theorem.

Before we do so, we need to talk a little bit about proofs. It is easiest here to assume that our system of logic is a so-called 'axiomatic' system, in which formal proofs are simply sequences of formulas satisfying certain conditions. So we think of proofs as precisely such objects: sequences of formulas, satisfying certain conditions.

Now, just as we are able to talk about formulas in the language of arithmetic, through Gödel numbering, so we can talk about proofs. We need only extend our Gödel numbering so that it encompasses finite sequences of formulas. This is easily done. Since we did not use 0 as the code of any of our primitives, we can use it as a separator, the way one might use a comma. So, e.g., the sequence $\forall x(x = x), 0 = 0$ can be given the Gödel number: $4919f920bfb_{16}$. What comes before the '0' in this numeral is the Gödel number of '$\forall x(x = x)$'; what comes after it is the Gödel number of '$0 = 0$'. Note that, in some systems, this would therefore be the Gödel number of a (correct) proof of the sentence '$0 = 0$'.

Terminology: By a Σ-proof, we mean a proof that uses, as well as any 'logical' axioms there may be, also any of the axioms in the formal theory Σ. And the crucial point now is that the property 'x is the Gödel number of a (correct) Σ-proof' is intuitively computable, if Σ is a formal theory: Given a putative proof, one can check it, mechanicially, for correctness. It is essentially definitive of a 'formal system of logic' that one can check purely logical proofs in this way; and if Σ is a formal theory, then we can tell what its axioms are, so we can check whether the (finitely many) 'non-logical' assumptions used in the proof are all axioms of Σ.

Moreover, the relation 'x is the Gödel number of a (correct) Σ-proof of y' is also computable: One simply checks whether x codes a proof and, if so, checks what is on the last line of that proof.

Hence, in Q, and in any other system that represents all recursive functions, there will be a formula $\text{BEW}_\Sigma(x, y)$[33] that represents the relation: x is the Gödel number of a Σ-proof of the formula with Gödel number y.

Note: Henceforth, we shall not bother with the distinction between formulas and their Gödel numbers. So we shall allow ourselves to say such things as: x is a Σ-proof of y, instead of what we just said.

Theorem 11.1. *Let Σ be a theory in which every recursive function is representable, and suppose that Σ is ω-consistent. Then there is a formula A that is undeciable by Σ, i.e., for which $\Sigma \nvdash A$ and also $\Sigma \nvdash \neg A$.*

Proof. Since every recursive function is representable in Σ, there is a formula $\text{BEW}_\Sigma(x, y)$ that represents, in Σ, the relation: x is a Σ-proof of y. Consider the following formula:

$$\neg\exists y(\text{BEW}_\Sigma(y, x))$$

which says that there is no Σ-proof of x, i.e., that x is not Σ-provable. By the diagonal lemma, we have a formula G such that

$$\Sigma \vdash G \equiv \neg\exists y(\text{BEW}_\Sigma(y, \ulcorner G \urcorner)) \tag{1}$$

Thus, G 'says of itself' that it is not Σ-provable.

I claim, first, that, if Σ is consistent, then $\Sigma \nvdash G$. For suppose Σ does prove G. Then there really is a Σ-proof of G. Let p be the Gödel number of this proof. Then p is a Σ-proof of G, so, since $\text{BEW}_\Sigma(x, y)$ represents Σ-proof in Σ:

$$\Sigma \vdash \text{BEW}_\Sigma(p, \ulcorner G \urcorner)$$

and so

$$\Sigma \vdash \exists y(\text{BEW}_\Sigma(y, \ulcorner G \urcorner))$$

But then by (1):

$$\Sigma \vdash \neg G$$

and so Σ is inconsistent.

I claim, second, that if Σ is ω-consistent, then $\Sigma \nvdash \neg G$. For suppose Σ does prove $\neg G$. Then by (1),

$$\Sigma \vdash \exists y(\text{BEW}_\Sigma(y, \ulcorner G \urcorner)) \tag{2}$$

Now, if Σ is ω-consistent, it is also consistent. So no natural number is actually the Gödel number of a Σ-proof of G. Otherwise, there actually would be a Σ-proof of G, i.e., Σ would prove G, and then Σ would be inconsistent, since we are supposing it proves $\neg G$. So, since $\text{BEW}_\Sigma(x, y)$ represents Σ-proof in Σ:

$$\Sigma \vdash \neg\text{BEW}_\Sigma(\overline{n}, \ulcorner G \urcorner) \tag{3}$$

for each n. But then (2) and (3) show Σ to be ω-inconsistent. □

[33] It is traditional to use 'Bew', which is taken from the German word for 'proof': *Beweis*. It is also what Gödel uses.

Note, again, that (3) says that Σ proves *each* case of $\neg\textsc{Bew}_\Sigma(x, \ulcorner G \urcorner)$, *not* that it proves the generalization $\forall x \neg\textsc{Bew}_\Sigma(x, \ulcorner G \urcorner)$. If it did that, then it would be inconsistent, since that contradicts (2). Since it only proves *each* case, however, it is just ω-inconsistent.

The hypothesis of ω-inconsistency can be weakened to simple inconsistency, as was first shown by Rosser. The proof uses a slightly more complicated diagonal construction.

Theorem 11.2 (Rosser's Theorem)**.** *Let Σ be a consistent theory containing* Q*. Then there is a formula A that is undeciable by Σ, i.e., for which $\Sigma \nvdash A$ and also $\Sigma \nvdash \neg A$.*

The proof requires the following fact about Q.

Proposition 11.3. *For each n,*

(i) $\quad \mathsf{Q} \vdash \forall x(x < \overline{n} \equiv x = 0 \vee x = \overline{1} \vee \cdots \vee x = \overline{n-1})$

(ii) $\quad \mathsf{Q} \vdash \forall x(x < \overline{n} \vee x = n \vee \overline{n} < x)$

When $n = 0$, (i) should be read as: $\mathsf{Q} \vdash \forall x(\neg x < 0)$.[34]

Proof. We prove the first part and leave the second as Exercise 15.17.

The proof is by induction on n. For the basis, then, we must show that $\mathsf{Q} \vdash \forall x(\neg x < 0)$. We have already done this, however, in the course of proving Proposition 5.2.

For the induction step, then, suppose

$$\mathsf{Q} \vdash \forall x(x < \overline{n} \equiv x = 0 \vee x = \overline{1} \vee \cdots \vee x = \overline{n-1})$$

We want to show that

$$\mathsf{Q} \vdash \forall x(x < \overline{n+1} \equiv x = 0 \vee x = \overline{1} \vee \cdots \vee x = \overline{n})$$

Right to left is easy: We know from Proposition 5.2 that $\mathsf{Q} \vdash \overline{m} < \overline{n+1}$ whenever $m < n+1$. But $0 < \overline{n+1}$, so $\mathsf{Q} \vdash 0 < \overline{n+1}$ and similarly for the other cases. So $\mathsf{Q} \vdash 0 < \overline{n+1} \wedge \overline{1} < \overline{n+1} \wedge \cdots \wedge \overline{n} < \overline{n+1}$, which logically implies $x = 0 \vee x = \overline{1} \vee \cdots \vee x = \overline{n-1} \to x < \overline{n+1}$.

Left to right is harder. And, before we begin, let me make a comment about how this proof will be presented. Rather than talk about what Q proves, I am simply going to present a proof in Q. Of course, it will not really be a formal proof, but it will be clear—if I am doing my job correctly—that the proof is one that could be 'formalized' in Q; that is, I will strive to make it clear that the proof uses only resources that are available in Q. This is sometimes called 'reasoning in Q', and it is a common method of presentation, as well as a useful way to think.

So, I invite you to reason with me in Q. Suppose that $x < \overline{n+1}$. We want to show that $x = 0$ or $x = \overline{1}$ or… $x = \overline{n}$. So we shall suppose further that $x \neq 0$, and we shall show that $x = \overline{1}$ or… $x = \overline{n}$. Now, by

[34] That is, we treat the empty disjunction as being logically false (since none of its disjuncts are true).

Q8, $\exists z(Sz + x = \overline{n+1})$, so fix some such z. As before, $\overline{n+1}$ is $S(\overline{n})$, so $Sz + x = S(\overline{n})$. But if $x \neq 0$, then by Q7, $x = Sy$, for some y; hence $Sz + Sy = S(\overline{n})$, and so by Q4, $S(Sz + y) = S(\overline{n})$ and so by Q2, $Sz + y = \overline{n}$. Thus, $\exists z(Sz + y = \overline{n})$, and so $y < \overline{n}$, by Q8. By the induction hypothesis, then, $y = 0$ or... or $y = \overline{n-1}$. So $Sy = S0$ or... or $Sy = S(\overline{n-1})$. That is: $x = \overline{1}$ or... or $x = \overline{n}$, as promised. □

Note carefully, again, what Proposition 11.3 says, namely, that Q proves *each* case of the displayed formula. Thus, for example,

$$\mathsf{Q} \vdash \forall x(x < SSS0 \equiv x = 0 \vee x = S0 \vee x = SS0)$$
$$\mathsf{Q} \vdash \forall x(x < SSS0 \vee x = SSS0 \vee SSS0 < x)$$

and so forth for the other cases. This is *not* to say that Q proves the generalization:

$$\forall y \forall x(x < y \vee x = y \vee y < x)$$

Indeed, Q does *not* prove the 'law of trichotomy'.

We can now prove Rosser's Theorem.

Proof of 11.2. Let Σ be as in the statement of the theorem, and consider the recursive relation: x is the negation of y. Since Σ contains Q, there is a formula $\textsc{Neg}(x, y)$ that represents this relation in Σ. Now consider the formula:

$$\exists y[\textsc{Bew}_\Sigma(y, x) \wedge \neg\exists z(z < y \wedge \exists w(\textsc{Neg}(x, w) \wedge \neg\textsc{Bew}_\Sigma(z, w)))]$$

This says that there is a proof of x such that there is no 'earlier' proof of x's negation (i.e., no such proof with a smaller Gödel number). By the diagonal lemma, there is a formula R such that Σ proves:

(1) $R \equiv \neg\exists y[\textsc{Bew}_\Sigma(y, \ulcorner R \urcorner) \wedge$
$\neg\exists z(z < y \wedge \exists w(\textsc{Neg}(\ulcorner R \urcorner, w) \wedge \neg\textsc{Bew}_\Sigma(z, w)))]$

Since $\ulcorner \neg R \urcorner$ is the negation of $\ulcorner R \urcorner$, Σ also proves:

$$\forall y(\textsc{Neg}(\ulcorner R \urcorner, y) \equiv y = \ulcorner \neg R \urcorner)$$

So, by logic, it proves:

(2) $R \equiv \neg\exists y[\textsc{Bew}_\Sigma(y, \ulcorner R \urcorner) \wedge \neg\exists z(z < y \wedge \neg\textsc{Bew}_\Sigma(z, \ulcorner \neg R \urcorner))]$

Thus, R 'says of itself' that there is no proof of it, unless there is an 'earlier' proof of its negation.

I claim first that $\Sigma \nvdash R$. The basic idea here is that, if Σ does prove R, then, since Σ is consistent, there will not be *any* Σ-proof of $\neg R$. In particular, none of the 'earlier' proofs will be proofs of $\neg R$, and this is something that Σ itself will be able to prove: There are only finitely many such 'earlier' proofs. But then that contradicts the right-hand side of (2). Let's fill in the details.

Suppose that there is a Σ-proof of R. Let p be the Gödel number of this proof. Then p is a Σ-proof of R, so, since $\textsc{Bew}_\Sigma(x, y)$ represents Σ-proof in Σ:

$$\Sigma \vdash \textsc{Bew}_\Sigma(\overline{p}, \ulcorner R \urcorner) \tag{3}$$

Moreover, since Σ is consistent, there is no proof of $\ulcorner \neg R \urcorner$. So:

$$\begin{gathered}\Sigma \vdash \neg\textsc{Bew}_\Sigma(0, \ulcorner \neg R \urcorner)\\ \Sigma \vdash \neg\textsc{Bew}_\Sigma(\overline{1}, \ulcorner \neg R \urcorner)\\ \vdots\\ \Sigma \vdash \neg\textsc{Bew}_\Sigma(\overline{p-1}, \ulcorner \neg R \urcorner)\end{gathered}$$

By 11.3, then:

$$\Sigma \vdash \neg\exists z < \overline{p}(\textsc{Bew}_\Sigma(z, \ulcorner \neg R \urcorner))$$

Putting this together with (3):

$$\Sigma \vdash \textsc{Bew}_\Sigma(\overline{p}, \ulcorner R \urcorner) \wedge \neg\exists z < \overline{p}(\textsc{Bew}_\Sigma(z, \ulcorner \neg R \urcorner))$$

and so:

$$\Sigma \vdash \exists y[\textsc{Bew}_\Sigma(y, \ulcorner R \urcorner) \wedge \neg\exists z < y(\textsc{Bew}_\Sigma(z, \ulcorner \neg R \urcorner))]$$

But this is just the negation of the right-hand side of (2), so $\Sigma \vdash \neg R$, and Σ is inconsistent.

I claim, second, that $\Sigma \nvdash \neg R$. For suppose Σ does prove $\neg R$. Then, as we are about to see, Σ proves the right-hand side of (2), in which case Σ proves R as well and is thus inconsistent. That is, in this case, Σ will be able to prove that there is no Σ-proof of R unless there is an earlier proof of $\neg R$. In particular, it can do so by trilemma: None of the (finitely many) proofs 'earlier' than the known proof of $\neg R$ can be proofs of R at all, since Σ is consistent; certainly the proof of $\neg R$ itself is not a proof of R; and any proof of R that might come 'after' the proof of $\neg R$ certainly isn't a proof of R that is not preceded by a proof of $\neg R$. To the details, then.

The right-hand side of (2) is logically equivalent to:

$$\forall y[\textsc{Bew}_\Sigma(y, \ulcorner R \urcorner) \rightarrow \exists z(z < y \wedge \neg\textsc{Bew}_\Sigma(z, \ulcorner \neg R \urcorner))] \tag{4}$$

so it is enough to show that Σ proves (4). We will divide this up into cases. Let the smallest Gödel number of a Σ-proof of $\neg R$ be p. Then, since $\mathsf{Q} \vdash \forall y(y < \overline{p} \vee y = \overline{p} \vee \overline{p} < y)$, by Proposition 5.2, Σ proves (3) iff it proves:

$$\forall y[y < \overline{p} \wedge \textsc{Bew}_\Sigma(y, \ulcorner R \urcorner) \rightarrow \exists z(z < y \wedge \textsc{Bew}_\Sigma(z, \ulcorner \neg R \urcorner))] \tag{4a}$$

$$\forall y[y = \overline{p} \wedge \textsc{Bew}_\Sigma(y, \ulcorner R \urcorner) \rightarrow \exists z(z < y \wedge \textsc{Bew}_\Sigma(z, \ulcorner \neg R \urcorner))] \tag{4b}$$

$$\forall y[\overline{p} < y \wedge \textsc{Bew}_\Sigma(y, \ulcorner R \urcorner) \rightarrow \exists z(z < y \wedge \textsc{Bew}_\Sigma(z, \ulcorner \neg R \urcorner))] \tag{4c}$$

But Σ clearly proves (4c), since it proves:

$$\textsc{Bew}_\Sigma(\overline{p}, \ulcorner \neg R \urcorner)$$

and so proves

$$\overline{p} < y \to (\overline{p} < y \wedge \textsc{Bew}_\Sigma(\overline{p}, \ulcorner \neg R \urcorner))$$

and therefore proves

$$\overline{p} < y \to \exists z(z < y \wedge \textsc{Bew}_\Sigma(z, \ulcorner \neg R \urcorner))$$

which logically implies (4c).

Moreover, since p is not (the code of) a proof of R (it's the code of a proof of $\neg R$), Σ proves that it isn't:

$$\neg \textsc{Bew}_\Sigma(\overline{p}, \ulcorner R \urcorner)$$

and so proves

$$\forall y(y = \overline{p} \to \neg \textsc{Bew}_\Sigma(y, \ulcorner R \urcorner))$$

which logically implies (4b), since it implies that the antecedent of (4b) is never true.

Finally, since

$$\mathsf{Q} \vdash \forall y(y < \overline{p} \equiv y = 0 \vee y = \overline{1} \vee \cdots \vee y = \overline{p-1})$$

Σ proves (4a) iff it proves each of:

$$\begin{gathered}
\textsc{Bew}_\Sigma(0, \ulcorner R \urcorner) \to \exists z(z < 0 \wedge \textsc{Bew}_\Sigma(z, \ulcorner \neg R \urcorner)) \\
\textsc{Bew}_\Sigma(\overline{1}, \ulcorner R \urcorner) \to \exists z(z < \overline{1} \wedge \textsc{Bew}_\Sigma(z, \ulcorner \neg R \urcorner)) \\
\vdots \qquad (5) \\
\textsc{Bew}_\Sigma(\overline{p-1}, \ulcorner R \urcorner) \to \exists z(z < \overline{p-1} \wedge \textsc{Bew}_\Sigma(z, \ulcorner \neg R \urcorner))
\end{gathered}$$

But Σ is consistent, so there is no Σ-proof of R. In particular, neither 0, nor 1, ..., nor $p-1$ codes a Σ-proof of R. So:

$$\begin{gathered}
\Sigma \vdash \neg \textsc{Bew}_\Sigma(0, \ulcorner R \urcorner) \\
\Sigma \vdash \neg \textsc{Bew}_\Sigma(\overline{1}, \ulcorner R \urcorner) \\
\vdots \qquad (6) \\
\Sigma \vdash \neg \textsc{Bew}_\Sigma(\overline{p-1}, \ulcorner R \urcorner)
\end{gathered}$$

and each of the formulae in group (5) follows logically from the corresponding formula in group (6). So Σ does prove (4a), and we are done. $\square$

The following related fact may seem like a curiosity, but it turns out to be important in connection with Gödel's second incompleteness theorem.

Proposition 11.4. *Let Σ be a consistent theory containing Q, and suppose $\textsc{Bew}_\Sigma(x, y)$ represents Σ-proof in Σ (or in Q). Then the formula*

$$\textsc{Bew}_\Sigma(x, y) \wedge \neg \exists z[z < x \wedge \exists w(\textsc{Neg}(y, w) \wedge \neg \textsc{Bew}_\Sigma(z, w))]$$

which we used in the proof of Rosser's Theorem, also represents Σ-proof in Σ (or in Q).

Proof. We need to show that, if p codes a proof of y, then Σ proves

$$\textsc{Bew}_\Sigma(\overline{p}, \overline{y}) \wedge \neg\exists z[z < \overline{p} \wedge \exists w(\textsc{Neg}(\overline{y}, w) \wedge \neg\textsc{Bew}_\Sigma(z, w))]$$

and, if it does not, then Σ proves

$$\neg\{\textsc{Bew}_\Sigma(\overline{p}, \overline{y}) \wedge \neg\exists z[z < \overline{p} \wedge \exists w(\textsc{Neg}(\overline{y}, w) \wedge \neg\textsc{Bew}_\Sigma(z, w))]\}$$

The latter is trivial: If p does not code a proof of y, then Σ proves $\neg\textsc{Bew}_\Sigma(\overline{p}, \overline{y})$, and the second conjunct is irrelevant.

Suppose, then, that p does code a proof of the formula with Gödel number y; let that formula be A. Then, of course, Σ proves $\textsc{Bew}_\Sigma(\overline{p}, \ulcorner A \urcorner)$. So we need to see that Σ also proves

$$\neg\exists z(z < \overline{p} \wedge \exists w(\textsc{Neg}(\ulcorner A \urcorner, w) \wedge \neg\textsc{Bew}_\Sigma(z, w)))$$

for which, as before, it is enough to see that it proves

$$\neg\exists z(z < \overline{p} \wedge \neg\textsc{Bew}_\Sigma(z, \ulcorner \neg A \urcorner)$$

But, as before (again), it will do so if it proves each of:

$$\neg\textsc{Bew}_\Sigma(0, \ulcorner \neg A \urcorner)$$
$$\neg\textsc{Bew}_\Sigma(\overline{1}, \ulcorner \neg A \urcorner)$$
$$\vdots$$
$$\neg\textsc{Bew}_\Sigma(\overline{p-1}, \ulcorner \neg A \urcorner)$$

And none of the mentioned numbers does code a proof of $\neg A$, since Σ is consistent. So Σ does prove all of these sentences, and we are done. □

12. Undecidability

The first incompleteness theorem is often stated in a somewhat different way, which we shall now formulate.

Definition. A theory Σ is said to be *decidable* if the set of its theorems, $\mathsf{Cl}(\Sigma)$, is recursive, and *undecidable* otherwise.

Σ is *essentially undecidable* if not only is Σ undecidable but every consistent theory Θ extending Σ is also undecidable.

Proposition 12.1. *Every complete formal theory is decidable.*

Proof. Let Σ be a complete formal theory. If Σ is inconsistent, then it is trivially decidable. (All sentences are theorems!) So suppose Σ is consistent.

Here, then, is a procedure for deciding whether A is a theorem of Σ. Starting with 0, look successively at each number and see if it is the code of a Σ-proof of either A or $\neg A$. We can decide this because T is a formal theory, so we can tell whether something is a Σ-proof or not, and we can tell what its last line is. Since Σ is complete, there is either a Σ-proof of A or a Σ-proof of $\neg A$, and we will eventually find one or the other. If we

find a proof of A, then of course $A \in \Sigma$. If we find a proof of $\neg A$, then it is not, since Σ is consistent. □

Note that this same sort of construction shows that there is a recursive procedure for listing the theorems of any formal theory Σ: One simply works one's way through the proofs, writing down their last lines. The set of theorems is therefore said to be *semi-recursive* or *recursively enumerable*.

Lemma 12.2. *Let Σ be a consistent theory containing* Q*. Then the set of Σ's theorems, $\mathsf{Cl}(\Sigma)$, is not representable in Σ.*

That is: There is no formula $\textsc{Thm}_\Sigma(x)$ of the language of Σ such that, whenever Σ proves A, it also proves $\textsc{Thm}_\Sigma(\ulcorner A \urcorner)$, and whenever it does not prove A, it proves $\neg\textsc{Thm}_\Sigma(\ulcorner A \urcorner)$.

Proof. Suppose there were such a formula. Then by the diagonal lemma there is a formula G such that

$$\Sigma \vdash G \equiv \neg\textsc{Thm}_\Sigma(\ulcorner G \urcorner)$$

The rest of the proof is Exercise 15.18. □

Theorem 12.3. Q *is essentially undecidable.*

Proof. If Q were decidable, then $\mathsf{Cl}(\mathsf{Q})$ would be recursive. But then it would be representable in Q, contradicting Lemma 12.2.

Now let Σ be a consistent theory extending Q. If Σ were decidable, then $\mathsf{Cl}(\Sigma)$ would be recursive and so representable in Q and so representable in Σ, again contradicting Lemma 12.2. □

13. Gödel's Second Incompleteness Theorem

The first incompleteness theorem tells us that no 'sufficiently strong', consistent formal theory decides every question that can be posed in its language. Here, a 'sufficiently strong' theory is one that is capable of representing every recursive function. The second incompleteness theorem gives us a specific, and interesting, example of such an incompleteness. Informally, what it says is that no 'sufficiently strong', consistent formal theory can prove its own consistency.

How should we formalize the statement that Σ is consistent? The obvious formulation would be something like: $\exists y \forall x(\neg\textsc{Bew}_\Sigma(x, y))$, i.e.: There is a formula that is not Σ-provable. When the language is that of arithmetic, however, it is traditional to formalize it as:

$$(\mathrm{Con}_\Sigma) \qquad \neg\exists x(\textsc{Bew}_\Sigma(x, \ulcorner 0 = \overline{1} \urcorner))$$

So long as Σ proves $0 \neq \overline{1}$, it will be inconsistent if it proves $0 = \overline{1}$. And of course any theory containing Q, or even just containing Q1, will prove $0 \neq \overline{1}$.

The second incompleteness theorem, which one often sees called just 'G2', can then be stated as:

Theorem 13.1 (G2). *If Σ is 'sufficiently strong', then $\Sigma \nvdash Con_\Sigma$.*

Exactly what 'sufficiently strong' means in this case is a nice question, about which we shall say a word below.

Note that it is perfectly possible for a consistent theory to prove its own *in*consistency. This is, in fact, a consequence of G2. For if Σ does not prove Con_Σ, then the theory $\Sigma \cup \{\neg\mathrm{Con}_\Sigma\}$ is consistent. And it can be shown that $\Sigma \cup \{\neg\mathrm{Con}_\Sigma\}$ proves $\neg\mathrm{Con}_{\Sigma\cup\{\neg\mathrm{Con}_\Sigma\}}$.

However, we do have the following:

Theorem 13.2. *Let Σ be an ω-consistent formal theory containing* Q*; let $\textsc{Bew}_\Sigma(x, y)$ be a formula that represents Σ-provability in Σ; and let Con_Σ be the corresponding consistency statement. Then Σ does not prove $\neg Con_\Sigma$.*

Proof. Suppose $\Sigma \vdash \neg\mathrm{Con}_\Sigma$, i.e., suppose

$$\Sigma \vdash \exists x(\textsc{Bew}_\Sigma(x, \ulcorner 0 = \overline{1}\urcorner)) \tag{1}$$

Since Σ is consistent, there is, in fact, no Σ proof of $0 = \overline{1}$. So since $\textsc{Bew}_\Sigma(x, y)$ represents Σ-provability in Σ, we have:

$$\Sigma \vdash \neg\textsc{Bew}_\Sigma(\overline{n}, \ulcorner 0 = \overline{1}\urcorner) \tag{2}$$

for each n. But then (1) and (2) show Σ to be ω-inconsistent. □

A complete proof of Theorem 13.1 would be extremely long, and one almost never sees anyone work through all the details. But the general shape of the proof is fairly easy to understand. In the proof of the first incompleteness theorem, we constructed a formula G_Σ by diagonalizing on $\neg\exists y(\textsc{Bew}_\Sigma(y, x))$:

$$\Sigma \vdash G_\Sigma \equiv \neg\exists y(\textsc{Bew}_\Sigma(y, \ulcorner G_\Sigma\urcorner))$$

We then showed that, if Σ is consistent, then Σ does not prove G_Σ. That is, we proved:

$$\mathrm{Con}_\Sigma \rightarrow G_\Sigma \tag{*}$$

We did not, of course, prove (*) in Σ; we just proved it informally. But one might well ask in what kinds of theories this kind of statement can be proven: Could it, for example, be proven in Q, or in PA? There is no real chance it will be provable in Q, since there are so many simple facts that Q does not prove. But it turns out that this statement *is* provable in PA, for any Σ you like. In particular, PA proves:

$$\mathrm{Con}_{\mathsf{PA}} \rightarrow G_{\mathsf{PA}} \tag{**}$$

But if (**) is provable in PA, then $\mathrm{Con}_{\mathsf{PA}}$ had better *not* be provable in PA, since then PA would prove G_{PA} and would be inconsistent: We know from the proof of the first incompleteness theorem that, if PA is consistent, it does not prove G_{PA}. More generally, no consistent theory Σ capable of

proving (*)—that is a first stab at what 'sufficiently strong' means in the statement of G2—can prove Con_Σ, since then it would prove G_Σ.

Careful analysis of the proof we gave of (*) shows that it appeals only to three facts about provability. To make them easier to state, let us abbreviate $\exists y(\text{BEW}_\Sigma(y, x))$ as: $\text{PRV}_\Sigma(x)$. Then the three facts we used were informal versions of the following statements:

D1: If A is Σ-provable, then $\Sigma \vdash \text{PRV}_\Sigma(\ulcorner A \urcorner)$
D2: $\Sigma \vdash \text{PRV}_\Sigma(\ulcorner A \to B \urcorner) \wedge \text{PRV}_\Sigma(\ulcorner A \urcorner) \to \text{PRV}_\Sigma(\ulcorner B \urcorner)$
D3: $\Sigma \vdash \text{PRV}_\Sigma(\ulcorner A \urcorner) \to \text{PRV}_\Sigma(\ulcorner \text{PRV}_\Sigma(\ulcorner A \urcorner) \urcorner)$

These are known as the Hilbert-Bernays-Löb *derivability conditions*.

It is easy to see that (D1) will be true so long as $\text{BEW}_\Sigma(x, y)$ represents provability in Σ. For suppose that A is Σ-provable, and let p be the Gödel number of some proof of it. Then $\Sigma \vdash \text{BEW}_\Sigma(\overline{p}, \ulcorner A \urcorner)$, so $\Sigma \vdash \exists y(\text{BEW}_\Sigma(y, \ulcorner A \urcorner))$, by logic.

(D2) is sometimes known as 'formalized *modus ponens*': It says that Σ knows that provability is closed under *modus ponens*. How hard this is to prove will depend upon exactly what formalization of logic we are using. But most such formalizations have *modus ponens* as a rule of inference, so the proof will just formalize the following argument:

Proof of (D2). Suppose $\text{PRV}_\Sigma(\ulcorner A \to B \urcorner)$ and $\text{PRV}_\Sigma(\ulcorner A \urcorner)$. Then there is an m-term sequence σ_1 each of whose members is either a logical axiom or an axiom of Σ, or else 'follows' from earlier members of the sequence by one of the 'rules of inference', and whose last member is $A \to B$; and there is a similar n-term sequence σ_2 whose last member is A.

Now consider the $n+m+1$ member sequence τ whose first m members are the members of σ_1 and whose next n members are the members of σ_2, and whose last member is B. Then τ is a Σ-proof of B. That is: It is a sequence each of whose members is either a logical axiom or an axiom of Σ, or else is the result of applying one of 'rules of inference' to earlier members of the sequence, and whose last member is B. That each of the first m terms satisfies this condition follows from the fact that σ_1 does; that each of the next n terms does, from the fact that σ_2 does; and that the last term of the sequence, B, satisfies the condition follows from the fact that it is the result of applying *modus ponens* to the m^{th} and $(m+n)^{\text{th}}$ members, which are just $A \to B$ and A. □

This is very elementary reasoning. As it turns out, it cannot be formalized in Q, because Q is too weak to prove that we can always stick sequences together in this way (that we can always 'concatenate' them, as it is said). But this can certainly be done in PA.

Finally, (D3) is just the formalization of (D1): It says that Σ knows that, whenever a formula is Σ-provable, then it is Σ-provable that that formula is Σ-provable. It is the proof of (D3) that is so tedious that one

almost never sees it carried out in detail. We will not even try to describe the proof here.

What we will do is show how (D1)–(D3) allow us to prove the second incompleteness theorem. A 'sufficiently strong' theory is thus one that allows us to prove the derivability conditions.

Lemma 13.3. *Let Σ be a consistent formal theory that satisfies the derivability conditions:*

D1: *If A is Σ-provable, then* $\Sigma \vdash \textsc{Prv}_\Sigma(\ulcorner A \urcorner)$.
D2: $\Sigma \vdash \textsc{Prv}_\Sigma(\ulcorner A \to B \urcorner) \wedge \textsc{Prv}_\Sigma(\ulcorner A \urcorner) \to \textsc{Prv}_\Sigma(\ulcorner B \urcorner)$
D3: $\Sigma \vdash \textsc{Prv}_\Sigma(\ulcorner A \urcorner) \to \textsc{Prv}_\Sigma(\ulcorner \textsc{Prv}_\Sigma(\ulcorner A \urcorner) \urcorner)$

Then Σ does not prove Con_Σ.

The proof is easiest if we first make this observation.

Proposition 13.4. *Let Σ be a consistent formal theory that proves (D1) and (D2). Then if $A_1, \ldots, A_n$ logically imply B, Σ proves:*

$$\textsc{Prv}_\Sigma(\ulcorner A_1 \urcorner) \wedge \cdots \wedge \textsc{Prv}_\Sigma(\ulcorner A_n \urcorner) \to \textsc{Prv}_\Sigma(\ulcorner B \urcorner)$$

We'll only need the case where $n = 2$, so we'll prove it and leave the generalization as Exercise 15.19.

Proof. If A_1 and A_2 together imply B, then $A_1 \to (A_2 \to B)$ is logically valid, and so certainly $\Sigma \vdash A_1 \to (A_2 \to B))$. So, by (D1),

$$\Sigma \vdash \textsc{Prv}_\Sigma(\ulcorner A_1 \to (A_2 \to B)) \urcorner)$$

So, by (D2),

$$\Sigma \vdash \textsc{Prv}_\Sigma(\ulcorner A_1 \urcorner) \to \textsc{Prv}_\Sigma(\ulcorner A_2 \to B \urcorner)$$

so by (D2) again,

$$\Sigma \vdash \textsc{Prv}_\Sigma(\ulcorner A_1 \urcorner) \to (\textsc{Prv}_\Sigma(\ulcorner A_2 \urcorner) \to \textsc{Prv}_\Sigma(\ulcorner B \urcorner))$$

But then

$$\Sigma \vdash \textsc{Prv}_\Sigma(\ulcorner A_1 \urcorner) \wedge \textsc{Prv}_\Sigma(\ulcorner A_2 \urcorner) \to \textsc{Prv}_\Sigma(\ulcorner B \urcorner)$$

by logic. □

As said earlier, the proof of G2 is simply a formalization of the reasoning used in the proof of (the first half of) G1.

Proof of 13.3. Consider the formula $\neg\textsc{Prv}_\Sigma(x)$, and diagonalize to get a formula G such that

$$\Sigma \vdash G \equiv \neg\textsc{Prv}_\Sigma(\ulcorner G \urcorner) \tag{1}$$

in which case certainly

$$\Sigma \vdash G \to \neg\textsc{Prv}_\Sigma(\ulcorner G \urcorner)$$

By (D1):

$$\Sigma \vdash \textsc{Prv}_\Sigma(\ulcorner G \to \neg\textsc{Prv}_\Sigma(\ulcorner G \urcorner) \urcorner)$$

and so by (D2):

(2) $$\Sigma \vdash \mathrm{PRV}_\Sigma(\ulcorner G \urcorner) \to \mathrm{PRV}_\Sigma(\ulcorner \neg \mathrm{PRV}_\Sigma(\ulcorner G \urcorner) \urcorner)$$

But by (D3):

(3) $$\Sigma \vdash \mathrm{PRV}_\Sigma(\ulcorner G \urcorner) \to \mathrm{PRV}_\Sigma(\ulcorner \mathrm{PRV}_\Sigma(\ulcorner G \urcorner) \urcorner)$$

But $\mathrm{PRV}_\Sigma(\ulcorner G \urcorner)$ and $\neg \mathrm{PRV}_\Sigma(\ulcorner G \urcorner)$ logically imply $0 = \overline{1}$, so by Proposition 13.4,

$$\Sigma \vdash \mathrm{PRV}_\Sigma(\ulcorner \mathrm{PRV}_\Sigma(\ulcorner G \urcorner) \urcorner) \wedge \mathrm{PRV}_\Sigma(\ulcorner \neg \mathrm{PRV}_\Sigma(\ulcorner G \urcorner) \urcorner) \to \mathrm{PRV}_\Sigma(\ulcorner 0 = \overline{1} \urcorner)$$

so putting that together with (2) and (3), we have, by logic:

$$\Sigma \vdash \mathrm{PRV}_\Sigma(\ulcorner G \urcorner) \to \mathrm{PRV}_\Sigma(\ulcorner 0 = \overline{1} \urcorner)$$

or, to put it differently:

$$\Sigma \vdash \mathrm{PRV}_\Sigma(\ulcorner G \urcorner) \to \neg \mathrm{Con}_\Sigma$$

and so, by logic:

$$\Sigma \vdash \mathrm{Con}_\Sigma \to \neg \mathrm{PRV}_\Sigma(\ulcorner G \urcorner)$$

But then, by (1):

$$\Sigma \vdash \mathrm{Con}_\Sigma \to G$$

Hence, if $\Sigma \vdash \mathrm{Con}_\Sigma$, then also $\Sigma \vdash G$, contradicting the first incompleteness theorem. □

So no consistent formal theory that proves the derivability conditions proves that it does not prove that $0 = 1$.

But something much stronger is true.

Corollary 13.5. *Let Σ be a consistent formal theory that proves the derivabilty conditions and contains* Q*. Then for no formula A does Σ prove $\neg \mathrm{PRV}_\Sigma(\ulcorner A \urcorner)$.*

Proof. By Proposition 13.4, Σ proves $\mathrm{PRV}_\Sigma(\ulcorner 0 = \overline{1} \urcorner) \wedge \mathrm{PRV}_\Sigma(\ulcorner 0 \neq \overline{1} \urcorner) \to \mathrm{PRV}_\Sigma(\ulcorner A \urcorner)$. But since Σ contains Q, certainly $\Sigma \vdash 0 \neq \overline{1}$, so by (D1), Σ proves $\mathrm{PRV}_\Sigma(\ulcorner 0 \neq \overline{1} \urcorner)$, and so Σ proves $\mathrm{PRV}_\Sigma(\ulcorner 0 = \overline{1} \urcorner) \to \mathrm{PRV}_\Sigma(\ulcorner A \urcorner)$. I.e,. $\Sigma \vdash \neg \mathrm{Con}_\Sigma \to \mathrm{PRV}_\Sigma(\ulcorner A \urcorner)$ and so, by logic, $\Sigma \vdash \neg \mathrm{PRV}_\Sigma(\ulcorner A \urcorner) \to \mathrm{Con}_\Sigma$. So if $\Sigma \vdash \neg \mathrm{PRV}_\Sigma(\ulcorner A \urcorner)$, then $\Sigma \vdash \mathrm{Con}_\Sigma$, contradicting G2. □

Corollary 13.6. *Let Σ be a consistent formal theory that proves the derivabilty conditions and contains* Q*. Then Σ does not prove*

$$\mathrm{PRV}_\Sigma(\ulcorner 0 = \overline{1} \urcorner) \to 0 = \overline{1}$$

Proof. If it did, then, since it contains Q and therefore proves $0 \neq \overline{1}$, it would prove $\neg \mathrm{PRV}_\Sigma(\ulcorner 0 = \overline{1} \urcorner)$, i.e., Con_Σ, contradicting G2. □

This is somewhat surprising. One might have thought, for example, that PA, say, should prove $\mathrm{PRV}_{\mathsf{PA}}(\ulcorner A \urcorner) \to A$, for *all* sentences A and, in particular, for $0 = \overline{1}$. After all, shouldn't what is provable be true? Well, yes, of course, but what 13.6 tells us is that PA cannot prove this fact about itself.

Indeed, something yet stronger is true. Obviously, if Σ actually proves A, then of course it will prove $\text{PRV}_\Sigma(\ulcorner A \urcorner) \rightarrow A$, by simple logic: $B \rightarrow A$ always follows from A. So what would be helpful is if Σ proved $\text{PRV}_\Sigma(\ulcorner A \urcorner) \rightarrow A$ in at least *some* cases in which it does *not* already prove A. The following result tells us that this *never* happens, unless Σ is inconsistent.

Theorem 13.7 (Löb's Theorem)**.** *Let Σ be a consistent formal theory that satisfies the derivabilty conditions. Then if $\Sigma \vdash \text{PRV}_\Sigma(\ulcorner A \urcorner) \rightarrow A$, then $\Sigma \vdash A$.*

Proof. Consider the formula $\text{PRV}_\Sigma(x) \rightarrow A$. Diagonalize to obtain a formula C such that:

(1) $$\Sigma \vdash C \equiv \text{PRV}_\Sigma(\ulcorner C \urcorner) \rightarrow A$$

and so, of course:

$$\Sigma \vdash C \rightarrow (\text{PRV}_\Sigma(\ulcorner C \urcorner) \rightarrow A)$$

Hence, by (D1):

$$\Sigma \vdash \text{PRV}_\Sigma(\ulcorner C \rightarrow (\text{PRV}_\Sigma(\ulcorner C \urcorner) \rightarrow A) \urcorner)$$

and so, by two applications of (D2):

$$\Sigma \vdash \text{PRV}_\Sigma(\ulcorner C \urcorner) \rightarrow (\text{PRV}_\Sigma(\ulcorner \text{PRV}_\Sigma(\ulcorner C \urcorner) \urcorner) \rightarrow \text{PRV}_\Sigma(\ulcorner A \urcorner))$$

But, by (D3):

$$\Sigma \vdash \text{PRV}_\Sigma(\ulcorner C \urcorner) \rightarrow \text{PRV}_\Sigma(\ulcorner \text{PRV}_\Sigma(\ulcorner C \urcorner) \urcorner)$$

So the last two yield:

$$\Sigma \vdash \text{PRV}_\Sigma(\ulcorner C \urcorner) \rightarrow \text{PRV}_\Sigma(\ulcorner A \urcorner)$$

And if, as we are assuming, $\Sigma \vdash \text{PRV}_\Sigma(\ulcorner A \urcorner) \rightarrow A$, then we have:

(2) $$\Sigma \vdash \text{PRV}_\Sigma(\ulcorner C \urcorner) \rightarrow A$$

But then (1) gives us:

$$\Sigma \vdash C$$

and so, by (D1):

$$\Sigma \vdash \text{PRV}_\Sigma(\ulcorner C \urcorner)$$

But then, by (2):

$$\Sigma \vdash A$$

as claimed. □

It's easy to derive G2 from Löb's Theorem. suppose $\Sigma \vdash \neg\text{PRV}_\Sigma(\ulcorner 0 = \overline{1} \urcorner)$. Then, trivially, $\Sigma \vdash \text{PRV}_\Sigma(\ulcorner 0 = \overline{1} \urcorner) \rightarrow 0 = \overline{1}$, and then Löb's Theorem implies that $\Sigma \vdash 0 = \overline{1}$, in which case Σ is inconsistent. So one might well see Löb's Theorem as a kind of generalization of G2.

It is also possible to derive Löb's Theorem from the second incompleteness theorem, a fact first noted around 1966 by Saul Kripke, but since rediscovered by several other people.

We know that, if $\Sigma \cup \{C\}$ proves some sentence B, then Σ itself proves $C \to B$, and of course the converse is also true. So, in particular, $\Sigma \cup \{C\}$ proves $0 = 1$ iff Σ proves $C \to 0 = 1$. We will need to know below that Σ is strong enough to prove such elementary facts, that is, that:

$$\Sigma \vdash \mathrm{PRV}_{\Sigma \cup \{C\}}(\ulcorner B \urcorner) \equiv \mathrm{PRV}_{\Sigma}(\ulcorner C \to B \urcorner)$$

and so, in particular:

$$\Sigma \vdash \mathrm{PRV}_{\Sigma \cup \{C\}}(\ulcorner 0 = \overline{1} \urcorner) \equiv \mathrm{PRV}_{\Sigma}(\ulcorner C \to 0 = \overline{1} \urcorner)$$

In fact, what we need to know is just the left-to-right direction:

$$\Sigma \vdash \mathrm{PRV}_{\Sigma \cup \{C\}}(\ulcorner 0 = \overline{1} \urcorner) \to \mathrm{PRV}_{\Sigma}(\ulcorner C \to 0 = \overline{1} \urcorner)$$

As with the second derivability condition, how hard this is to prove will depend upon how we formalize the logic. But this will certainly be provable in any sensible theory capable of proving the derivability conditions, which of course we already need for the second incompleteness theorem.

Proof of Löb's Theorem from G2. Let Σ be a consistent extension of Q that satisfies the derivability conditions. Suppose $\Sigma \vdash \mathrm{PRV}_{\Sigma}(\ulcorner A \urcorner) \to A$, and suppose for *reductio* that Σ does *not* prove A. Then the theory $\Sigma \cup \{\neg A\}$ is also consistent. Moreover, since $\Sigma \cup \{\neg A\}$ is an extension of Σ, certainly:

$$\Sigma \cup \{\neg A\} \vdash \mathrm{PRV}_{\Sigma}(\ulcorner A \urcorner) \to A$$

and so by logic:

$$\Sigma \cup \{\neg A\} \vdash \neg \mathrm{PRV}_{\Sigma}(\ulcorner A \urcorner) \tag{1}$$

since obviously $\Sigma \cup \{\neg A\} \vdash \neg A$. However, since $\Sigma \vdash 0 \neq \overline{1}$, by logic, again:

$$\Sigma \vdash (\neg A \to 0 = \overline{1}) \to A$$

so by (D1):

$$\Sigma \vdash \mathrm{PRV}_{\Sigma}(\ulcorner (\neg A \to 0 = \overline{1}) \to A \urcorner)$$

whence by (D2):

$$\Sigma \vdash \mathrm{PRV}_{\Sigma}(\ulcorner \neg A \to 0 = \overline{1} \urcorner) \to \mathrm{PRV}_{\Sigma}(\ulcorner A \urcorner)$$

Since $\Sigma \cup \{\neg A\}$ is an extension of Σ, again:

$$\Sigma \cup \{\neg A\} \vdash \mathrm{PRV}_{\Sigma}(\ulcorner \neg A \to 0 = \overline{1} \urcorner) \to \mathrm{PRV}_{\Sigma}(\ulcorner A \urcorner)$$

and so by (1):

$$\Sigma \cup \{\neg A\} \vdash \neg \mathrm{PRV}_{\Sigma}(\ulcorner \neg A \to 0 = \overline{1} \urcorner)$$

But we are assuming that Σ is strong enough to prove:

$$\Sigma \vdash \mathrm{PRV}_{\Sigma \cup \{\neg A\}}(\ulcorner 0 = \overline{1} \urcorner) \to \mathrm{PRV}_{\Sigma}(\ulcorner \neg A \to 0 = \overline{1} \urcorner)$$

and, again, $\Sigma \cup \{\neg A\}$ will prove the same thing, since it is an extension of Σ. So

$$\Sigma \cup \{\neg A\} \vdash \neg \mathrm{PRV}_{\Sigma \cup \{\neg A\}}(\ulcorner 0 = \overline{1} \urcorner)$$

i.e.,

$$\Sigma \cup \{\neg A\} \vdash \mathrm{Con}_{\Sigma \cup \{\neg A\}}$$

contrary to G2. □

Consider the formula

$$\text{RBEW}_\Sigma(x,y) \stackrel{df}{\equiv} \text{BEW}_\Sigma(x,y) \wedge \neg\exists z[z < x \wedge \exists w(\text{NEG}(y,w) \wedge \neg\text{BEW}_\Sigma(z,w))]$$

used in the proof of Rosser's Theorem. As we saw earlier (see Proposition 11.4), $\text{RBEW}_\Sigma(x,y)$ represents the Σ-proof relation whenever $\text{BEW}_\Sigma(x,y)$ does. So we can formulate a notion of 'Rosser consistency' that corresponds to the notion of 'Rosser provability'. Toward that end, let RCON_Σ abbreviate $\neg\exists y(\text{RBEW}_\Sigma(y, \ulcorner 0 = \overline{1} \urcorner)$. Then we have the following.

Theorem 13.8. *Let Σ be a consistent theory. Then* $\mathsf{Q} \vdash \text{RCON}_\Sigma$.

It follows of course that $\mathsf{Q} \vdash \text{RCON}_\mathsf{Q}$ and that, if Σ contains Q, then $\Sigma \vdash \text{RCON}_\Sigma$.

Proof. Unpacking all the definitions, what we need to show is that Q proves

$$\neg\exists y[\text{BEW}_\Sigma(y, \ulcorner 0 = \overline{1} \urcorner) \wedge \neg\exists z[z < y \wedge \exists w(\text{NEG}(\ulcorner 0 = \overline{1} \urcorner, w) \wedge \neg\text{BEW}_\Sigma(z,w))]$$

What this says, note, is that there is no Σ-proof of $0 = \overline{1}$ that is not preceded by a proof of $0 \neq \overline{1}$. But Q can prove this by the same 'divide and conquer' strategy used in the proof of Rosser's Theorem itself. See that proof for the details. I'll just sketch the general idea.

Since Σ contains Q, it does prove $0 \neq \overline{1}$. So let p be the least Gödel number of a proof of it. Then of course Q knows that p is a Σ-proof of $0 \neq \overline{1}$. Moreover, Q knows that all numbers are either less than p, equal to p, or greater than p. But, since Σ is consistent, none of the finitely many numbers less than p can be the Gödel number of a proof of $0 = \overline{1}$, and Q will be able to verify that fact. And certainly p itself isn't a proof of $0 = \overline{1}$, since it is actually a proof of its negation. Finally, just by logic, any number greater than p that is the Gödel number of a proof of $0 = \overline{1}$ will be greater than a number that is the Gödel number of a proof of $0 \neq \overline{1}$, namely, p. □

What I'm calling 'Rosser consistency' is thus completely trivial. It's no achievement for a theory to be able to prove its own Rosser consistency. Q already proves, of each consistent theory, that it is Rosser consistent.

But Theorem 13.8 does raise an important question, namely: When we say that no sufficiently strong consistent theory proves its own consistency, what exactly do we mean by "proves its own consistency"? What exactly, that is to say, are these 'consistency statements' of whose non-provability G2 is supposed to assure us? Well, consistency statements are just built, in the obvious say, from notions of provability: To say that a theory is consistent is just to say that it does not prove $0 = \overline{1}$; so we would know what a consistency statement was if we knew what a 'notion of provability' was. More precisely, if we knew when to say that some

formula $\text{BEW}_\Sigma(x, y)$ appropriately formalized the relation: x is a Σ-proof of y, then we could just say that a sentence is a consistency statement if it built from such a formula.

In the case of the first incompleteness theorem, it is enough for us to assume that $\text{BEW}_\Sigma(x, y)$ *represents* the relation: x is a Σ-proof of y. Given just that much, we can show that, if Σ is a consistent extension of Q, then the result of diagonalizing on $\neg\exists y(\text{BEW}_\Sigma(y, x))$ will be undecidable by Σ. That is, the arguments we gave above establish the following:

Theorem 13.9. *Let Σ be an ω-consistent extension of* Q *and suppose that $\text{BEW}_\Sigma(x, y)$ is a formula that represents in Σ the relation: x is a Σ-proof of y. Then the sentence G that is Σ-provably equivalent to $\neg\exists y(\text{BEW}_\Sigma(y, \ulcorner G \urcorner))$ is not decidable by Σ.*

What Theorem 13.8 shows, however, is that this is too weak a condition for the purposes of G2. It is *not true* that, if $\text{BEW}_\Sigma(x, y)$ represents in Σ the relation: x is a Σ-proof of y, then the consistency statement built from that formula will not be provable in Σ, no matter *what* reasonable conditions we put on Σ. The problem is not that $\text{RBEW}_\Sigma(x, y)$ does not *represent* the proof relation; it does. So what conditions do we need to put on $\text{BEW}_\Sigma(x, y)$ if we are to get a satisfying, general statement of G2?

That is a good question.

14. Tarski's Theorem

We now prove Tarski's Theorem.

Lemma 14.1 (The Liar Paradox)**.** *Let Σ be a theory in which diagonalization is representable and so for which the diagonal lemma holds. Suppose that there is a formula $T(x)$ of the language of Σ such that, for every sentence A of the language of Σ:*

(T) $$\Sigma \vdash A \equiv T(\ulcorner A \urcorner)$$

Then Σ is inconsistent.

Proof. Let $T(x)$ be as in the statement of the theorem. Consider: $\neg T(x)$. By the diagonal lemma, there is a sentence λ such that Σ proves:

(1) $$\lambda \equiv \neg T(\ulcorner \lambda \urcorner)$$

But since Σ proves all instances of the T-schema, we have that Σ proves:

(2) $$\lambda \equiv T(\lambda)$$

But if Σ proves both (1) and (2), it also proves

$$\lambda \equiv \neg\lambda$$

and so is (clasically) inconsistent. □

There is another way of stating this result. First, we need a definition.

Definition. A theory Σ *contains its own truth-predicate* if there is a formula $T(x)$ of the language of Σ such that, for every sentence A of that language, Σ proves: $A \equiv T(\ulcorner A \urcorner)$.

The restatement is then as follows.

Corollary 14.2. *Let Σ be a consistent theory in which diagonalization is representable. Then Σ does not contain its own truth-predicate.*

Proof. Immediate from the Liar Paradox. □

Note that it is not assumed that Σ is a *formal* theory. So we get:

Corollary 14.3. *Arithmetic does not contain its own truth-predicate. That is, there is no formula $T(x)$ of the language of arithmetic such that all instances of schema (T) are true (in the standard interpretation).*

Proof. Arithmetic is the theory whose members are all the sentences of the language of arithmetic that are true in the standard interpretation. As we saw earlier, arithmetic is closed: So its theorems are just its members. And so, for every sentence B of the language of arithmetic, B is true in the standard interpretation iff Arithmetic $\vdash B$.

Now, suppose there is a formula $T(x)$ in the language of arithmetic such that, for every sentence A of that language, '$A \equiv T(\ulcorner A \urcorner)$' is true in the standard interpretation. Then, for every sentence A, Arithmetic $\vdash A \equiv T(\ulcorner A \urcorner)$. But arithmetic extends Q and so represents all recursive functions and so represents diagonalization. So that contradicts corollary 14.2, since arithmetic is consistent. □

Theorem 14.4. *(Tarski's Theorem, Special Case) Arithmetical truth is not arithmetically definable. That is: There is no formula $T(x)$ of the language of arithmetic such that $T(\overline{n})$ is true in the standard interpretation when and only when n is the Gödel number of a sentence of the language of arithmetic that is itself true in the standard interpretation.*

Proof. Suppose there were such a formula. Let A be arbitrary. A is either true or false in the standard interpretation. If it is true, then $T(\ulcorner A \urcorner)$ must be true in the standard interpretation, whence '$A \equiv T(\ulcorner A \urcorner)$' is true, as well. Similarly, if A is false, then $T(\ulcorner A \urcorner)$ is false and so '$A \equiv T(\ulcorner A \urcorner)$' is again true. So all instances of schema (T) are true, contradicting corollary 14.3. □

Here's an application of the generalized diagonal lemma. Suppose Σ represents diagonalization. Let $A_1(x_1, x_2)$ be: $\neg T(x_2)$, and let $A_2(x_1, x_2)$ be: $T(x_1)$. By the generalized diagonal lemma, there are sentences λ and μ such that Σ proves:

(a) $\lambda \equiv \neg T(\ulcorner \mu \urcorner)$
(b) $\mu \equiv T(\ulcorner \lambda \urcorner)$

Suppose now that Σ proves:

(c) $\lambda \equiv T(\ulcorner \lambda \urcorner)$
(d) $\mu \equiv T(\ulcorner \mu \urcorner)$

Then Σ is inconsistent. For we have $\lambda \equiv_{(c)} T(\ulcorner \lambda \urcorner) \equiv_{(b)} \mu \equiv_{(d)} T(\ulcorner \mu \urcorner) \equiv_{(a)} \neg\lambda$, where the subscripts indicate to which of (a)–(d) we are appealing.

This is known as the Postcard Paradox: Imagine a postcard on one side of which is written "The sentence on the other side of this card is false"; on the other side is written "The sentence on the other side of this card is true".

15. Exercises

Many of the exercises that follow are not particularly difficult. The goal is thus not just to complete the problem: to prove the proposition, or whatever it might be. Rather, the goal is to produce an *elegant* proof of the proposition: one that, as far as possible, makes it clear precisely *why* the proposition is true. A good proof has a narrative structure: It tells a story, the story of why something is true. So you should aim, even in the simplest problems, to produce something that is not just 'right' just elegant.

Exercise 15.1.

(i) Show by induction that, in the standard interpretaton of the language of arithmetic, $\overline{n}$ always denotes n.
(ii) Show that $\overline{n} + \overline{m}$ always denotes the sum of n and m.
(iii) Show that $\overline{n} \times \overline{m}$ always denotes the product of n and m.

Exercise 15.2. Prove Proposition 3.2.

Exercise 15.3. Since we have not said what formal system of deduction we are using, the proof above of Proposition 3.3 tacitly appeals several times to soundness and completeness. Re-write the argument to make those appeals explicit.

Exercise 15.4. Prove Proposition 3.3 purely syntactically for the system of deduction described in Appendix A.

Exercise 15.5. A theory Σ is said to be *maximal consistent* if it is maximal with respect to consistency. I.e., Σ is maximal consistent iff, for every theory Θ in the same language, if Σ is a *proper* subset of Θ, then Θ is inconsistent. Prove that every complete closed theory is maximal consistent, and that every maximal consistent theory is both complete and closed.

Exercise 15.6. Prove parts (4)–(7) of Proposition 5.2.

Exercise 15.7. Show that '$\forall x(0 + x = x)$' is not a theorem of Q. (Hint: The domain will need to be expanded, so that it contains not just the natural numbers but also some 'rogue' objects for which addition behaves strangely.)

The model you give is likely to falsify other arithmetical truths, or easily to be adapatable so that it does so. Play around with this a bit, and prove that Q doesn't prove some other simple arithmetical facts.

Exercise 15.8. Prove Proposition 5.3.

Hint 1: It is enough to show that Q proves every true equality of the form '$t = \overline{n}$', where $\overline{n}$ is a numeral. This, together with what we already know from Proposition 5.2, immediately implies Proposition 5.3. Why?

Hint 2: The proof uses induction on the complexity of terms. What Proposition 5.2 says is that Q proves every true equality of the form

'$t = \overline{n}$', where t contains only *one* operator (S, $+$, or $\times$). What you need to do is show that this holds no matter how many operators t contains.

Exercise 15.9. Prove proposition 5.8. Note that you may reason informally (i.e., you do not have to give a fully formal proof). Just make sure you are appealing only to axioms of PA (and logic).

Exercise 15.10. Show that PA proves the commutativity of addition, i.e., $\forall x \forall y(y + x = x + y)$. The proof uses induction on x, so the inductive formula is $\forall y(y + x = x + y)$, and you want to show that PA proves $\forall y(y + 0 = 0 + y)$ and $\forall y(y + n = n + y) \rightarrow \forall y(y + Sn = Sn + y)$. You may again reason informally.

Hint: You may also need to prove $\forall x \forall y(x + Sy = Sx + y)$. The proof is again by induction.

There are many more exercises of this type that one can do, for example, the fact mentioned in the text that $x < 1 \rightarrow x < x^2$.

Exercise 15.11. Show that condition (ii) in Definition 6.3 actually follows from condition (i).

Exercise 15.12. Prove Proposition 6.7.

Exercise 15.13. Prove Theorem 6.8 using Theorem 4.6. Better yet, and optionally, prove it for the case of many-place functions, and not just for one-place functions.

Exercise 15.14. Prove Corollary 8.2. (Hint: Since this is a Corollary of Lemma 8.1, it is supposed to be an easy consequence of it.)

Exercise 15.15. Let Σ be an arbitrary theory. Prove the following facts about its closure:

(i) $Cl(\Sigma)$ is closed.
(ii) $Cl(\Sigma) = Cl(Cl(\Sigma))$.
(iii) Suppose $\Theta \supseteq \Sigma$, and that Θ is closed. Then $\Theta \supseteq Cl(\Sigma)$. (Hint: Show that if $\Sigma \subseteq \Delta$, then $Cl(\Sigma) \subseteq Cl(\Delta)$, and then apply (ii).)

Exercise 15.16. (Challenging) Prove Lemma 10.3.

Exercise 15.17. Prove the second part of Proposition 11.3.

Exercise 15.18. Complete the proof of Lemma 12.2.

Exercise 15.19. Prove Proposition 13.4 in full generality.

Appendix A. The Sequent Calculus

A *sequent* is an expression of the form: $A_1, \ldots, A_n \Rightarrow B$. We think of the formulae to the left of the arrow as describing a *set*, so they can be freely re-arranged, and each formula need be written only once. We shall use capital Greek letters to abbreviate sequences of formulas, thus: $\Sigma \Rightarrow A$.

A *derivation* is a list of sequents each of which is either a *basic* sequent of the form $A \Rightarrow A$ or else follows from earlier sequents by one of the rules below. A derivation of A from Σ is a derivation whose last line is $\Gamma \Rightarrow A$, where $\Gamma \subseteq \Sigma$. This allows for the possibility that Σ might be infinite.

Note that, for each logical connective, there are two rules for it. These are known as *introduction* and *elimination* rules. The *introduction* rule for a given connective tells you how, most fundamentally, to derive a formula in which that is the main connective. The *elimination* rule similarly tells you how, most fundamentally, to derive something *from* a formula in which that is the main connective.

$$\frac{}{A \Rightarrow A}$$
(Basic Sequents)

$$\frac{\Gamma \Rightarrow A \quad \Delta \supseteq \Gamma}{\Delta \Rightarrow A}$$
(Thinning)

$$(\neg\text{+}) \quad \frac{\Gamma, A \Rightarrow B \quad \Delta, A \Rightarrow \neg B}{\therefore \Gamma, \Delta \Rightarrow \neg A} \qquad (\neg\text{-}) \quad \frac{\Gamma \Rightarrow \neg\neg A}{\therefore \Gamma \Rightarrow A}$$

$$(\vee\text{+}) \quad \frac{\Gamma \Rightarrow A}{\therefore \Gamma \Rightarrow A \vee B} \qquad (\vee\text{-}) \quad \frac{\Gamma, A \Rightarrow C \quad \Delta, B \Rightarrow C \quad \Theta \Rightarrow A \vee B}{\therefore \Gamma, \Delta, \Theta \Rightarrow C}$$

$$(\vee\text{+}) \quad \frac{\Gamma \Rightarrow B}{\therefore \Gamma \Rightarrow A \vee B}$$

$$(\wedge\text{+}) \quad \frac{\Gamma \Rightarrow A \quad \Delta \Rightarrow B}{\Gamma, \Delta \Rightarrow A \wedge B} \qquad (\wedge\text{-}) \quad \frac{\Gamma \Rightarrow A \wedge B}{\therefore \Gamma \Rightarrow A}$$

$$(\wedge\text{-}) \quad \frac{\Gamma \Rightarrow A \wedge B}{\therefore \Gamma \Rightarrow B}$$

$(\rightarrow+)$ $\quad \Gamma, A \Rightarrow B$
$\therefore \Gamma \Rightarrow A \rightarrow B$

$(\rightarrow-)$ $\quad \Gamma \Rightarrow A \rightarrow B$
$\Delta \Rightarrow A$
$\therefore \Gamma, \Delta \Rightarrow B$

$(=+)$ $\quad \emptyset \Rightarrow t = t$

$(=-)$ $\quad \Gamma \Rightarrow A(s)$
$\Delta \Rightarrow s = t$
$\therefore \Gamma, \Delta \Rightarrow A(t)$

$(\exists+)$ $\quad \Gamma \Rightarrow A(t)$
$\Gamma \Rightarrow \exists x A(x)$

$(\exists-)$ $\quad \Gamma, A(y) \Rightarrow B$
$\Delta \Rightarrow \exists x A(x)$
$\therefore \Gamma, \Delta \Rightarrow B$

$(\forall+)$ $\quad \Gamma \Rightarrow A(y)$
$\therefore \Gamma \Rightarrow \forall x A(x)$

$(\forall-)$ $\quad \Gamma \Rightarrow \forall x A(x)$
$\Gamma \Rightarrow A(t)$

Note: In the identity and quantifier rules, t and s can be any term, subject to restrictions on capturing; y can be any variable, subject to the condition, in $\exists-$, that it not be free in Γ or in B, and in $\forall+$, that it not be free in Γ.

Example Deductions.

$$\forall x(Fx \rightarrow Gx), \exists x(Fx) \Rightarrow \exists x(Gx)$$

(1)	$\forall x(Fx \rightarrow Gx) \Rightarrow \forall x(Fx \rightarrow Gx)$	Basic
(2)	$\forall x(Fx \rightarrow Gx) \Rightarrow Fa \rightarrow Ga$	$\forall-$
(3)	$Fa \Rightarrow Fa$	Basic
(4)	$\forall x(Fx \rightarrow Gx), Fa \Rightarrow Ga$	(2,3)$\rightarrow-$
(5)	$\forall x(Fx \rightarrow Gx), Fa \Rightarrow \exists x(Gx)$	(4)$\exists+$
(6)	$\exists x(Fx) \Rightarrow \exists x(Fx)$	Basic
(7)	$\forall x(Fx \rightarrow Gx), \exists x(Fx) \Rightarrow \exists x(Gx)$	(5,6)$\exists-$

$$\exists y \forall x(Lxy) \Rightarrow \forall x \exists y(Lxy)$$

(1)	$\forall x(Lxa) \Rightarrow \forall x(Lxa)$	Basic
(2)	$\forall x(Lxa) \Rightarrow Lxa$	(1)$\forall-$
(3)	$\forall x(Lxa) \Rightarrow \exists y(Lxy)$	(2)$\exists+$
(4)	$\forall x(Lxa) \Rightarrow \forall x \exists y(Lxy)$	(3)$\forall+$
(5)	$\exists y \forall x(Lxy) \Rightarrow \exists y \forall x(Lxy)$	Basic
(6)	$\exists y \forall x(Lxy) \Rightarrow \forall x \exists y(Lxy)$	(4,5)$\exists-$

$$\forall x(\exists y(Ayx) \to Axx), \forall x\forall y(Lxy \to Axy) \Rightarrow \forall x(\exists y(Lyx) \to \exists y(Ayx))$$

(1)	$Lau \Rightarrow Lau$	Basic
(2)	$\forall x\forall y(Lxy \to Axy) \Rightarrow \forall x(Lxy \to Axy)$	Basic
(3)	$\forall x\forall y(Lxy \to Axy) \Rightarrow \forall y(Lay \to Aay)$	(2)$\forall -$
(4)	$\forall x\forall y(Lxy \to Axy) \Rightarrow Lau \to Aau$	(3)$\forall -$
(5)	$\forall x\forall y(Lxy \to Axy), Lau \Rightarrow Aau$	(1,4)$\to -$
(6)	$\exists y(Lyu) \Rightarrow \exists y(Lyu)$	Basic
(7)	$\forall x\forall y(Lxy \to Axy), \exists y(Lyu) \Rightarrow Aau$	(5,6)$\exists -$
(8)	$\forall x\forall y(Lxy \to Axy), \exists y(Lyu) \Rightarrow \exists u(Ayu)$	(7)$\exists +$
(9)	$\forall x(\exists y(Ayx) \to Axx) \Rightarrow \forall x(\exists y(Ayx) \to Axx)$	Basic
(10)	$\forall x(\exists y(Ayx) \to Axx) \Rightarrow \exists y(Ayu) \to Auu$	(9)$\forall -$
(11)	$\forall x(\exists y(Ayx) \to Axx), \forall x\forall y(Lxy \to Axy), \exists y(Lyu) \Rightarrow Auu$	(8,10)$\to -$
(12)	$\forall x(\exists y(Ayx) \to Axx), \forall x\forall y(Lxy \to Axy), \exists y(Lyu) \Rightarrow \exists y(Ayu))$	(11)$\exists +$
(13)	$\forall x(\exists y(Ayx) \to Axx), \forall x\forall y(Lxy \to Axy) \Rightarrow \exists y(Lyu) \to \exists y(Ayu))$	(12)$\to +$
(14)	$\forall x(\exists y(Ayx) \to Axx), \forall x\forall y(Lxy \to Axy) \Rightarrow \forall x(\exists y(Lyx) \to \exists y(Ayx))$	(13)$\forall +$

References

Boolos, G. (1993). *The Logic of Provability*. New York, Cambridge University Press.

Boolos, G. S., Burgess, J. P., and Jeffrey, R. C. (2007). *Computability and Logic*, 5th edition. Cambridge, Cambridge University Press.

Boolos, G. S. and Jeffrey, R. C. (1989). *Computability and Logic*, 3d edition. New York, Cambridge University Press.

Tarski, A., Mostowski, A., and Robinson, A. (1953). *Undecidable Theories*. Amsterdam, North-Holland Publishing.