FORMAL BACKGROUND FOR THE INCOMPLETENESS AND UNDEFINABILITY THEOREMS

RICHARD G. HECK, JR.

CONTENTS

The 1930s saw enormous advances in logic. The first, and perhaps 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, much of the discussion of "Gödel's theorem" is marred by a poor understanding of his results. Of course, 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 Gödel's theorem-type logic course were, for that same reason, barred from any real understanding of this important result. And I am convinced that 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. 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 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 fixed, for example: $\forall x(0 < x \rightarrow x < x + x)$. And the set of formulas in which we are interested is drawn from a fixed 'language', whose 'vocabulary' is determined in advance.

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

[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.

predicate symbols, such as '$>$' or 'F', of various numbers of arguments. (A sentence-letter is a zero-place predicate-letter.) We assume that these languages have the usual syntax of first-order logic. We regard languages as *fixed* and identify them with their stock of non-logical expressions.

One important language is the *language of arithmetic*, whose (basic) non-logical symbols are the constant '0', the one-place function-symbol 'S', two two-place function-symbols '$+$', and '$\times$', and a two-place relation symbol '$<$'.[4] 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. 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—of any language that contains both 'F' and 'G'. But it is not a sentence of the language of arithmetic, which does not contain either.

As syntax is usually developed, formally, 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: '$(y + (Sy))$', not: '$y + Sy$', for example. 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.[5]

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, 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''.

---

[4] 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.

[5] 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 syntax complicated.

In giving proofs in syntax, one frequently uses a method of proof 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, first, that 0 has P and, second, that, if a number n has P, then the next number, $n+1$, 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—'0' or a variable—or is the result of applying certain syntatic operations to the basic terms. So suppose we knew that '0' had P and that every variable had P; and suppose we knew further that, whenever terms t and u have P, so do the terms: $\ulcorner St\urcorner$, $\ulcorner t + u\urcorner$, and $\ulcorner t \times u\urcorner$. Then it would follow that every term has P. Similarly, suppose we knew that all atomic formulae had P—the atomic formulae, in this case, are of the form '$\ulcorner t = u\urcorner$' and '$\ulcorner t < u\urcorner$'—and that, whenever two formulae A and B have P, so do: $\ulcorner\neg A\urcorner$, $\ulcorner A \vee B\urcorner$, $\ulcorner A \wedge B\urcorner$, $\ulcorner A \rightarrow B\urcorner$, $\ulcorner\exists vA\urcorner$ and $\ulcorner\forall vA\urcorner$, for any variable v. Then, again, it would follow that all formulae have 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 and extensions to function- and predicate-symbols. In this case, the domain is the set of all natural numbers; '0' denotes zero; 'S' denotes successor, or plus one; '$+$' denotes addition; '$\times$', multiplication; and '$<$', the less-than relation.[6] 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), which is false.

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 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. 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 $\mathbf{n}$ to mean: the numeral that denotes n, that is:

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

So, we may now write things like $\ulcorner\mathbf{2} \neq \mathbf{1}\urcorner$, to mean: $SS0 \neq S0$.

[6] If we want to use the standard reduction of functions to set, 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 in terms of functions.

Note the use of the funny quotes, which are called corners and which are strictly speaking necessary. If Σ is a theory in the language of arithmetic, then Σ does *not* prove '$\mathbf{2} \neq \mathbf{1}$', for the simple reason that the expression '$\mathbf{2} \neq \mathbf{1}$' is not a formula of the language of arithmetic—and that for the simple reason that '$\mathbf{2}$' 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. Quotes themselves are even written as blanks, sometimes. We may indulge in this practice ourselves from time to time. But we shall try not to overdo it.

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 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:

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

The language here contains the constant symbol 'e' 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 is commutative.

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. Indeed, one of them is complete (the one in the smaller language), whereas the other is incomplete.

(2) The theory known as *weak* (or *adjunctive*) *set theory* is: $\{\exists x \forall y (y \notin x), \forall u \forall v \exists x \forall y (y \in x \equiv y \in u \vee y = v)\}$. Its language is the language of set theory.

(3) 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 will not be if Γ is infinite.

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: $\Sigma \vdash B$.

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 Σ proves A 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.

As noted, Σ implies B if, and only if, B 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 B if, and only if, B is true in every model of Σ. If there is a model of Σ in which B is not true, then Σ does not imply B, so B is not derivable from Σ (by the soundness theorem), so B is not a theorem of Σ. Such a model is called a *counter-model* for B.

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. For then, Σ implies A if, and only if, Σ proves A, using any

such system of axioms and rules. We shall appeal to this fact frequently, without any explicit notice, and without bothering to formalize the logic.

3. Some Important Properties of Theories

Definition 3.1. We say that 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.

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

The Proof is left as an exercise.

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. Exercise 15.2 asks you to re-write the argument to make those appeals explicit.

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*.

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,[7] so let us mean by a complete theory a complete *consistent* theory.

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).

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

[7] 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 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.

A consistent theory that is both complete and closed is said to be *maximal consistent*, because, as the following shows, it is *maximal* with respect to consistency: No consistent theory in the same language properly contains it.

Proposition 3.6. *Let Σ be a complete, closed theory. Let Θ be any theory in the same language that properly contains Σ. Then Θ is inconsistent.*

Proof. Let Σ and Θ be as in the statement of the proposition. Then there is some formula A in the common language of Σ and Θ such that $A \in \Theta$ but $A \notin \Sigma$. Now, if $\Sigma \vdash A$, then $A \in \Sigma$, since Σ is closed, so $\Sigma \nvdash A$. Since Σ is complete, we must then have that $\Sigma \vdash \neg A$. But then, since $\Sigma \subseteq \Theta$, $\Theta \vdash \neg A$, too. But obviously $\Theta \vdash A$, since $A \in \Theta$, and so $\Theta \vdash A \wedge \neg A$, and hence Θ is inconsistent. □

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.

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 sentences in the language of $\mathcal{T}$. So the most for which one could hope is that these are the *only* 'new' theorems of $\mathcal{T}^*$, 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.[8]

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.

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

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 constants 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 value (object or set, as appropriate) to the non-logical constants 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 constants 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 $\mathcal{M}^*$. 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. □

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 $\boldsymbol{Q}$, is the theory with the following eight axioms:[9]

(1) $\forall x(Sx \neq 0)$
(2) $\forall x \forall y(Sx = Sy \rightarrow x = y)$
(3) $\forall x(x + 0 = x)$
(4) $\forall x \forall y(x + Sy = S(x + y))$[10]
(5) $\forall x(x \times 0 = 0)$
(6) $\forall x \forall y(x \times Sy = (x \times y) + x)$[11]
(7) $\forall x(x \neq 0 \rightarrow \exists y(x = Sy))$
(8) $x < y \equiv \exists z(y = Sz + x)$

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

In some ways, $\boldsymbol{Q}$ is a very weak theory. It is easy to see that $\boldsymbol{Q}$ does not, for example, prove that addition is commutative: That is, '$\forall x \forall y(x + y = y + x)$' is not a theorem of $\boldsymbol{Q}$. Even '$\forall x(0 + x = x)$' turns out not to be a theorem of $\boldsymbol{Q}$. In another sense, however, $\boldsymbol{Q}$ is quite a powerful theory, as we shall see.

$\boldsymbol{Q}$ 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, $\boldsymbol{Q}$ proves $\ulcorner \mathbf{n} \neq \mathbf{m} \urcorner$. So, for example, since $2 \neq 1$, $\boldsymbol{Q}$ proves $\ulcorner \mathbf{2} \neq \mathbf{1} \urcorner$, or: $SS0 \neq S0$.

Proposition 5.1. *Q proves all basic arithmetical equalities and inequalities. That is:*

(1) If $n = m$, $Q \vdash \mathbf{n} = \mathbf{m}$, and if $n \neq m$, $Q \vdash \mathbf{n} \neq \mathbf{m}$.
(2) If $n + m = k$, then $Q \vdash \mathbf{n} + \mathbf{m} = \mathbf{k}$, and if $n + m \neq k$, then $Q \vdash \mathbf{n} + \mathbf{m} \neq \mathbf{k}$.
(3) If $n \times m = k$, then $Q \vdash \mathbf{n} \times \mathbf{m} = \mathbf{k}$, and if $n \times m \neq k$, then $Q \vdash \mathbf{n} \times \mathbf{m} \neq \mathbf{k}$.
(4) If $m < n$, then $Q \vdash \mathbf{m} < \mathbf{n}$.

Proof. We prove (1) and leave the remainder to the reader.

[9] When '$<$' is not included in the language, the eighth is taken to define it.
[10] I.e., $x + (y + 1) = (x + y) + 1$.
[11] I.e., $x \times (y + 1) = (x \times y) + x$.

That $\boldsymbol{Q}$ proves all true numerical identities is obvious: These are just logical truths of the form: $S \ldots S0 = S \ldots S0$. For the case of non-identities, the proof is by induction on m.[12]

For the basis, we must show that $\boldsymbol{Q}$ proves $\mathbf{n} \neq \mathbf{0}$, if $n \neq 0$. But in that case, $\mathbf{n}$ is $S \ldots S0$, and $S \ldots S0 \neq 0$ will follow from axiom Q1.

For the induction step, we suppose that $\boldsymbol{Q}$ proves $\mathbf{n} \neq \mathbf{m}$ when $n \neq m$ and show $\boldsymbol{Q}$ proves $\mathbf{n} \neq S\mathbf{m}$ when $n \neq m+1$. Note first that this certainly holds if $n = 0$, by the preceding. So suppose $n \neq 0$ and $n \neq m+1$. Then $\mathbf{n}$ is $S\mathbf{k}$, for some numeral $\mathbf{k}$, and we need to show that $\boldsymbol{Q}$ proves $S\mathbf{k} \neq S\mathbf{m}$, whenever, in fact, $n = k+1 \neq m+1$.

Since $k+1 \neq m+1$, also $k \neq m$. So, by the induction hypothesis, $\boldsymbol{Q}$ proves $\mathbf{k} \neq \mathbf{m}$. By Q2, however, $\boldsymbol{Q}$ proves $S\mathbf{k} = S\mathbf{m} \rightarrow \mathbf{k} = \mathbf{m}$. But then it proves $S\mathbf{k} \neq S\mathbf{m}$, by *modus tollens*. □

Note carefully here what $\boldsymbol{Q}$ is being said to prove. It follows from what has been said, for example, that, for each n and m, $\boldsymbol{Q}$ proves $\mathbf{n}+\mathbf{m} = \mathbf{m}+\mathbf{n}$. So $\boldsymbol{Q}$ is being said to prove each of infinitely many theorems. But, as already said, $\boldsymbol{Q}$ 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 5.2. 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, $\mathcal{T} \vdash A(\mathbf{n})$ 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, $\mathcal{U} \vdash A(\mathbf{n})$; 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 5.3. 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, $\mathcal{T} \vdash A(\mathbf{n})$ but also $\mathcal{T} \vdash \exists x \neg A(x)$. Otherwise, $\mathcal{T}$ is ω-consistent.

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

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

[12] Note that this is an induction we are carrying out in the language we are speaking. It is not an argument we are carrying out in $\boldsymbol{Q}$, or one that can even be 'formalized' in $\boldsymbol{Q}$. It can't be, since $\boldsymbol{Q}$ doesn't have any axioms corresponding to mathematical induction.

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

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

where $A(x)$ is a formula of the language of arithmetic. Note that the standard interpretation of the language of arithmetic is a model of PA: All of the infinitely many axioms of PA are true in it.

Proposition 5.4. *Axiom 7 of Q is provable in PA.*

Proof. This is exercise 15.5. □

Arithmetic (sometimes called 'true arithmetic') is the theory containing all truths of 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, 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, for some z, $2^z > \varphi(z)$, so long as $z > y$.*

We can put this by saying that exponentiation grows faster than any polynomial.

Proof. Exercise in basic number theory. A careful proof actually shows how to calculate such a y, given a specific polynomial $\varphi(x)$. □

It then follows, obviously, 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 *relation* $y = 2^x$,[13] 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

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

can be proven in PA.[14] So we can think of PA as proving that $x < 2^x$ if it proves

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

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.[15]

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) $\Phi(\mathbf{k}_1, \ldots, \mathbf{k}_n, \mathbf{m})$ is true (in the standard interpretation) iff $\phi(k_1, \ldots, k_n) = m$;

(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.

---

[13] This is often called the 'graph' of the exponentiation function.

[14] Strictly speaking, of course, PA does not prove the mentioned formula, since '$\textsc{Exp}$' 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.

[15] The contrast between representability and definability here is *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.

Note that the notion of definability is *semantic*: It is given in terms of the notion of *truth*. 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) $\Sigma \vdash \Phi(\mathbf{k}_1, \ldots, \mathbf{k}_n, \mathbf{m})$

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

or, equivalently:

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

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:

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

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

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

(ii) $\forall y(\textsc{Exp}(\mathbf{m}, y) \to y = \mathbf{n})$

But it is an important fact about $\boldsymbol{Q}$—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:

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

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* $m \neq l$, Σ *proves* $\ulcorner \mathbf{m} \neq \mathbf{l} \urcorner$. *Then whenever* $\Phi(k_1, \ldots, k_n, y) \neq m$, $\Sigma \vdash \neg\Phi(\mathbf{k}_1, \ldots, \mathbf{k}_n, \mathbf{m})$.

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

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

Proof. Immediate from 5.1 and the preceding proposition. □

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

Definition. A formula $\Phi(x_1, \ldots, x_n)$ *defines* the n-place relation $F(x_1, \ldots, x_n)$ if $\Phi(\mathbf{k}_1, \ldots, \mathbf{k}_n)$ 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* $F(x_1, \ldots, x_n)$ 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. 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 $F(k_1, \ldots, k_n)$, $\Sigma \vdash \Phi(\mathbf{k}_1, \ldots, \mathbf{k}_n)$
(ii) whenever $\neg F(k_1, \ldots, k_n)$, $\Sigma \vdash \neg\Phi(\mathbf{k}_1, \ldots, \mathbf{k}_n)$

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. *A set or relation is definable (or representable in Σ) iff its characteristic function is definable (or representable in Σ).*

Proof. This is exercise 15.8. □

As noted above, representability is the way we 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 $\forall x[\exists y(\textsc{Exp}(x, y) \wedge \forall z(\textsc{Exp}(x, z) \rightarrow y = z)]$ 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 $F(x, y)$.*[16] *Suppose further that Σ proves:*

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

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 15.9. □

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.[17] 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 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 would need a mathematical definition of the set of computable functions, and that is precisely what the notion of recursive

[16] The extension of many-place functions is trivial.

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

function is supposed to be. (This is a form of the paradox of analysis.) 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.

As said, 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, 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 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.

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***. 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. The standard

representation uses something called the American Standard Code for Information Interchange. ASCII, as it is known, defines a correspondence between letters (and other symbols) and numbers. 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.

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 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 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 now 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, paren('$)+0\exists\vee$') 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:

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

Here 0 is a 'throwaway' value, provided so that $\text{PAREN}(x)$ is a total function. Since $\text{PAREN}(x)$ is clearly computable, it is representable in $\boldsymbol{Q}$. So there is a formula $\text{PAREN}(x, y)$ such that, whenever $n = \text{PAREN}(m)$, $\boldsymbol{Q}$ proves: $\text{PAREN}(\mathbf{m}, \mathbf{n})$. Thus, $\boldsymbol{Q}$ proves $\text{PAREN}(\mathbf{2db35_{16}}, \mathbf{12db352_{16}})$. Or, as we may also put it: $\boldsymbol{Q}$ 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 on strings. Then there is a formula $\Phi(x_1, \ldots, x_n, y)$ of the language of arithmetic that represents it in $\boldsymbol{Q}$, 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) $Q \vdash \Phi(\sigma_1, \ldots, \sigma_n, \tau)$

(ii) $Q \vdash \forall x(\Phi(\sigma_1, \ldots, \sigma_n, x) \to x = \tau)$

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. Let φ be as in the statement of the Lemma. We define a numerical function φ^* as follows:

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

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 $\boldsymbol{Q}$. So there is a formula $\Phi(x_1, \ldots, x_n, y)$ such that, whenever $\varphi^*(\sigma_1, \ldots, \sigma_n) = \tau$:

(i) $Q \vdash \Phi(\sigma_1, \ldots, \sigma_n, \tau)$

(ii) $Q \vdash \forall x(\Phi(\sigma_1, \ldots, \sigma_n, x) \to x = \tau)$

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 $\boldsymbol{Q}$, in the sense that, where $\sigma_1, \ldots, \sigma_n$ are the Gödel numbers of $s_1, \ldots, s_n$:*

(i) *whenever* $\varphi(s_1, \ldots, s_n)$, $Q \vdash \Phi(\sigma_1, \ldots, \sigma_n)$
(ii) *whenever* $\neg\varphi(s_1, \ldots, s_n)$, $Q \vdash \neg\Phi(\sigma_1, \ldots, \sigma_n)$

Proof. This is exercise 15.10. □

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 $\Theta \vdash A$.

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: $Cl(\Sigma)$—is the set of its theorems, that is, of the formulae derivable from it.

We can easily prove that $Cl(\Sigma)$ is always closed, 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 $Cl(\Sigma) = Cl(\Theta)$.

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 and Tarski's 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; 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 $\boldsymbol{Q}$, since $\boldsymbol{Q}$ 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*)[18] 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: $A(\ulcorner A(x) \urcorner)$.[19] This operation is obviously recursive, so (we will pretend that) there is an open term, $\textsc{Pdiag}(x)$, 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 $\textsc{Pdiag}(\mathbf{m}) = \mathbf{n}$.[20] Now consider the expression:[21]

$$(P) \qquad A(\textsc{Pdiag}(x))$$

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

$$(G) \qquad A(\textsc{Pdiag}(\ulcorner A(\textsc{Pdiag}(x)) \urcorner))$$

which contains no free variables[22] 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 $\ulcorner A(\textsc{Pdiag}(x) \urcorner$. Hence:

$$\Sigma \vdash \textsc{Pdiag}(\ulcorner A(\textsc{Pdiag}(x)) \urcorner) = \ulcorner G \urcorner$$

and so, by the laws of identity:

$$\Sigma \vdash A(\textsc{Pdiag}(\ulcorner A(\textsc{Pdiag}(x)) \urcorner)) \equiv A(\ulcorner G \urcorner)$$

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* '$\textsc{Diag}(x, y)$' instead of the pretend term '$\textsc{Pdiag}(x)$'. 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.

---

[18] 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.

[19] If the arugment 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.

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

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

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

Definition 10.2. 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 '$\exists x$'.

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 10.3 (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 $\Sigma \vdash G \equiv A(\ulcorner G \urcorner)$.*

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, $\Sigma \vdash \forall x(\textsc{Diag}(\mathbf{m}, x) \equiv x = \mathbf{n})$.[23] 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:

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

[23] 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.

And now, from (1) and (3), Σ proves:

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

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

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

as wanted. □

Of course, the diagonal lemma, as we originally stated it, 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 this version of the diagonal lemma.

The diagonal lemma can be generalized in two ways: We can allow other free variables 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:

Lemma 10.4. *(Generalized Diagonal Lemma) Let Σ be a theory that represents every recursive function and, for $i = 1, \ldots n$, 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. 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 to the reader. (It is not easy, but it is not impossible, either.)

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*. Fix some formal system of logic. It doesn't really matter which one it is, so long as a proof in the system consists of a sequence 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's before the '0' in this numeral is the Gödel number of '$\forall x(x = x)$'; what's after it is the Gödel number of '$0 = 0$'. Note that, in many 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 is a proof and, if so, checks what is on its last line.

Hence, in $\boldsymbol{Q}$, and in any other system that represents all recursive functions, there will be a formula $\textsc{Bew}_\Sigma(x, y)$[24] 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 formula 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. Consider the following formula:

$$\neg\exists y(\textsc{Bew}_\Sigma(y, x))$$

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

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

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 $\textsc{Bew}_\Sigma(x, y)$ represents Σ-proof in Σ:

$$\Sigma \vdash \textsc{Bew}_\Sigma(p, \ulcorner G \urcorner)$$

and so

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

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),

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

[24] It is traditional to use 'Bew', which is taken from the German word for 'proof': *Beweis*.

Now, if Σ is ω-consistent, it is also consistent; so no natural number is the Gödel number of a Σ-proof of G, since, otherwise, there actually would be a Σ-proof of G, and Σ would be inconsistent. So, since $\textsc{Bew}_\Sigma(x, y)$ represents Σ-proof in Σ:

$$\Sigma \vdash \neg\textsc{Bew}_\Sigma(\mathbf{n}, \ulcorner G \urcorner) \tag{3}$$

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

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 $\boldsymbol{Q}$. 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 $\boldsymbol{Q}$.

Proposition 11.3. *For each n,*

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

Note carefully, again, what this says, namely, that $\boldsymbol{Q}$ proves *each* case of the displayed formula. Thus, for example,

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

and so forth for the other cases. This is *not* to say that $\boldsymbol{Q}$ proves the generalization:

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

Indeed, $\boldsymbol{Q}$ does *not* prove the 'law of trichotomy'; nor does it prove the generalization corresponding to the other part of 11.3.

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 $\boldsymbol{Q}$, 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 (i.e., no proof with a smaller Gödel number) of x's negation. By the diagonal

lemma, there is a formula R such that Σ proves:

(1) $R \equiv \neg\exists y[\mathrm{BEW}_\Sigma(y, \ulcorner R \urcorner) \wedge$
$\neg\exists z(z < y \wedge \exists w(\mathrm{NEG}(\ulcorner R \urcorner, w) \wedge \neg\mathrm{BEW}_\Sigma(z, w)))]$

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

$$\forall y(\mathrm{NEG}(\ulcorner R \urcorner, y) \equiv y = \ulcorner \neg R \urcorner)$$

So, by logic, it proves:

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

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 $\Sigma \nvdash R$. For suppose 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 $\mathrm{BEW}_\Sigma(x, y)$ represents Σ-proof in Σ:

(3) $$\Sigma \vdash \mathrm{BEW}_\Sigma(\mathbf{p}, \ulcorner R \urcorner)$$

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

$$\Sigma \vdash \neg\mathrm{BEW}_\Sigma(\mathbf{0}, \ulcorner \neg R \urcorner)$$
$$\Sigma \vdash \neg\mathrm{BEW}_\Sigma(\mathbf{1}, \ulcorner \neg R \urcorner)$$
$$\vdots$$
$$\Sigma \vdash \neg\mathrm{BEW}_\Sigma(\mathbf{p} - \mathbf{1}, \ulcorner \neg R \urcorner)$$

By 11.3, then:

$$\Sigma \vdash \neg\exists z < \mathbf{p}(\mathrm{BEW}_\Sigma(z, \ulcorner \neg R \urcorner))$$

Putting this together with (3):

$$\Sigma \vdash \mathrm{BEW}_\Sigma(\mathbf{p}, \ulcorner R \urcorner) \wedge \neg\exists z < \mathbf{p}(\mathrm{BEW}_\Sigma(z, \ulcorner \neg R \urcorner))$$

and so:

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

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 otherwise. We shall show that Σ proves the right-hand side of (2), in which case Σ is inconsistent, since then it proves R, as well. Now, the right-hand side of (2) is logically equivalent to:

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

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 $Q \vdash \forall y(y < \mathbf{p} \vee y = \mathbf{p} \vee \mathbf{p} < y)$, Σ proves (3) iff it proves:

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

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

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

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

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

and so proves

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

and therefore proves

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

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:

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

and so proves

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

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

Finally, since

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

Σ proves (4a) iff it proves each of:

$$\textsc{Bew}_\Sigma(\mathbf{0}, \ulcorner R \urcorner) \rightarrow \exists z(z < \mathbf{0} \wedge \textsc{Bew}_\Sigma(z, \ulcorner \neg R \urcorner))$$
$$\textsc{Bew}_\Sigma(\mathbf{1}, \ulcorner R \urcorner) \rightarrow \exists z(z < \mathbf{1} \wedge \textsc{Bew}_\Sigma(z, \ulcorner \neg R \urcorner))$$

(5) $\quad \vdots$

$$\textsc{Bew}_\Sigma(\mathbf{p} - \mathbf{1}, \ulcorner R \urcorner) \rightarrow \exists z(z < \mathbf{p} - \mathbf{1} \wedge \textsc{Bew}_\Sigma(z, \ulcorner \neg R \urcorner))$$

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

$$\Sigma \vdash \neg \textsc{Bew}_\Sigma(\mathbf{0}, \ulcorner R \urcorner)$$
$$\Sigma \vdash \neg \textsc{Bew}_\Sigma(\mathbf{1}, \ulcorner R \urcorner)$$

(6) $\quad \vdots$

$$\Sigma \vdash \neg \textsc{Bew}_\Sigma(\mathbf{p} - \mathbf{1}, \ulcorner R \urcorner)$$

and each of the formulae in group (5) follows logically from the corresponding formula in group (5). 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.

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

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

also represents Σ-proof in Σ.

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

$$\text{BEW}_\Sigma(\mathbf{p}, \mathbf{y}) \wedge \neg\exists z[z < \mathbf{p} \wedge \exists w(\text{NEG}(\mathbf{y}, w) \wedge \neg\text{BEW}_\Sigma(z, w))]$$

and, if it does not, then Σ proves

$$\neg\{\text{BEW}_\Sigma(\mathbf{p}, \mathbf{y}) \wedge \neg\exists z[z < \mathbf{p} \wedge \exists w(\text{NEG}(\mathbf{y}, w) \wedge \neg\text{BEW}_\Sigma(z, w))]\}$$

The latter is trivial: If p does not code a proof of y, then Σ proves $\neg\text{BEW}_\Sigma(\mathbf{p}, \mathbf{y})$, 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 $\text{BEW}_\Sigma(\mathbf{p}, \ulcorner A \urcorner)$. So we need to see that Σ also proves

$$\neg\exists z(z < \mathbf{p} \wedge \exists w(\text{NEG}(\ulcorner A \urcorner, w) \wedge \neg\text{BEW}_\Sigma(z, w)))$$

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

$$\neg z(z < \mathbf{p} \wedge \neg\text{BEW}_\Sigma(z, \ulcorner \neg A \urcorner)$$

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

$$\neg\text{BEW}_\Sigma(\mathbf{0}, \ulcorner \neg A \urcorner)$$
$$\neg\text{BEW}_\Sigma(\mathbf{1}, \ulcorner \neg A \urcorner)$$
$$\vdots$$
$$\neg\text{BEW}_\Sigma(\mathbf{p} - \mathbf{1}, \ulcorner \neg A \urcorner)$$

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, $Cl(\Sigma)$, 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. 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. □

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 theory containing $\boldsymbol{Q}$. Then the set of Σ's theorems is not representable in Σ. I.e., 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 there is a formula G such that

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

by the diagonal lemma.... □

Exercise 12.3. Complete the proof that was just started.

Theorem 12.4. *$\boldsymbol{Q}$ is essentially undecidable.*

Proof. If $\boldsymbol{Q}$ were decidable, then $Cl(Q)$ would be recursive. But then it would be representable in $\boldsymbol{Q}$, contradicting Lemma 12.2. Now let Σ be a consistent theory extending $\boldsymbol{Q}$. If Σ were decidable, then $Cl(\Sigma)$ would be recursive and so representable in $\boldsymbol{Q}$ 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, exampe 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:

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

So long as Σ contains (i.e., proves) the two axioms concerning successor, it will certainly prove that $0 \neq 1$. So if it proves that $0 = 1$, it will be inconsistent.

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. But it is completely trivial that, if $\neg\mathrm{Con}_\Sigma$, then $\neg\mathrm{Con}_{\Sigma\cup\{\neg\mathrm{Con}_\Sigma\}}$, since every Σ-proof is obviously a $\Sigma \cup \{\neg\mathrm{Con}_\Sigma\}$-proof; even $\boldsymbol{Q}$ will prove that. So, since $\Sigma \cup \{\neg\mathrm{Con}_\Sigma\}$ trivially proves $\neg\mathrm{Con}_\Sigma$, it also 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 $\boldsymbol{Q}$; 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

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

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

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

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 $\neg\exists y(\textsc{Bew}_\Sigma(y, x))$:

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

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

(*) $$\mathrm{Con}_\Sigma \rightarrow G_\Sigma$$

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 $\boldsymbol{Q}$, or in PA? There is no real chance it will be provable in $\boldsymbol{Q}$, since there are so many simple facts that $\boldsymbol{Q}$ does not prove. But it turns out that this statement is provable in PA, for any Σ you like. In particular, PA proves:

(**) $$\mathrm{Con}_{PA} \rightarrow G_{PA}$$

But if (**) is provable in PA, then Con_{PA} had better *not* be provable in PA, since then PA would prove G_{PA} and would be inconsistent. 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 $\exists y(\textsc{Bew}_\Sigma(y,x))$ as: $\textsc{Prv}_\Sigma(x)$. Then the three facts we used are:

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

This will, in fact, be true so long as $\textsc{Bew}_\Sigma(x,y)$ represents provability in Σ. For suppose that A is Σ-provable, and let p be the Gödel number of some proof of it. Then $\Sigma \vdash \textsc{Bew}_\Sigma(\mathbf{p}, \ulcorner A \urcorner)$, so $\Sigma \vdash \exists y(\textsc{Bew}_\Sigma(y, \ulcorner A \urcorner))$, by logic.

D2: $\Sigma \vdash \textsc{Prv}_\Sigma(\ulcorner A \to B \urcorner) \wedge \textsc{Prv}_\Sigma(\ulcorner A \urcorner) \to \textsc{Prv}_\Sigma(\ulcorner B \urcorner)$

This 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 a proper formalization of the following argument:

Proof of (D2). Suppose $\textsc{Prv}_\Sigma(\ulcorner A \to B \urcorner)$ and $\textsc{Prv}_\Sigma(\ulcorner A \urcorner)$. 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 B. Consider the 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 B does 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, and in much weaker theories, too.

D3: $\Sigma \vdash \textsc{Prv}_\Sigma(\ulcorner A \urcorner) \to \textsc{Prv}_\Sigma(\ulcorner \textsc{Prv}_\Sigma(\ulcorner A \urcorner) \urcorner)$

This, note, 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), which are together known as the *derivability conditions*, 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 proves*
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:*

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

Proof. We'll only need the case where $n = 2$, so we'll prove it and leave the generalization as an exercise. If A_1 and A_2 together imply B, then $A_1 \to (A_2 \to B)$ is 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):

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

But by (D3):

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

But $\text{PRV}_\Sigma(\ulcorner G \urcorner)$ and $\neg\text{PRV}_\Sigma(\ulcorner G \urcorner)$ logically entail $\mathbf{0} = \mathbf{1}$, so by 13.4,

$$\Sigma \vdash \text{PRV}_\Sigma(\ulcorner \text{PRV}_\Sigma(\ulcorner G \urcorner) \urcorner) \wedge \text{PRV}_\Sigma(\ulcorner \neg\text{PRV}_\Sigma(\ulcorner G \urcorner) \urcorner) \rightarrow \text{PRV}_\Sigma(\ulcorner \mathbf{0} = \mathbf{1} \urcorner)$$

so, by logic:

$$\Sigma \vdash \text{PRV}_\Sigma(\ulcorner G \urcorner) \rightarrow \text{PRV}_\Sigma(\ulcorner \mathbf{0} = \mathbf{1} \urcorner)$$

or, to put it differently:

$$\Sigma \vdash \text{PRV}_\Sigma(\ulcorner G \urcorner) \rightarrow \neg\text{Con}_\Sigma$$

and so, by logic:

$$\Sigma \vdash \text{Con}_\Sigma \rightarrow \neg\text{PRV}_\Sigma(\ulcorner G \urcorner)$$

But then, by (1):

$$\Sigma \vdash \text{Con}_\Sigma \rightarrow G$$

Hence, if $\Sigma \vdash \text{Con}_\Sigma$, 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 $\neg\text{PRV}_\Sigma(\ulcorner A \urcorner)$.*

Proof. By 13.4, Σ proves $\text{PRV}_\Sigma(\ulcorner \mathbf{0} = \mathbf{1} \urcorner) \wedge \text{PRV}_\Sigma(\ulcorner \mathbf{0} \neq \mathbf{1} \urcorner) \rightarrow \text{PRV}_\Sigma(\ulcorner A \urcorner)$. But since Σ contains **Q**, certainly $\Sigma \vdash \mathbf{0} \neq \mathbf{1}$, so $\text{PRV}_\Sigma(\ulcorner \mathbf{0} = \mathbf{1} \urcorner) \rightarrow \text{PRV}_\Sigma(\ulcorner A \urcorner)$. I.e,. $\Sigma \vdash \neg\text{Con}_\Sigma \rightarrow \text{PRV}_\Sigma(\ulcorner A \urcorner)$ and so, by logic, $\Sigma \vdash \neg\text{PRV}_\Sigma(\ulcorner A \urcorner) \rightarrow \text{Con}_\Sigma$. So if $\Sigma \vdash \neg\text{PRV}_\Sigma(\ulcorner A \urcorner)$, then $\Sigma \vdash \text{Con}_\Sigma$, 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 $\text{PRV}_\Sigma(\ulcorner \mathbf{0} = \mathbf{1} \urcorner) \rightarrow \mathbf{0} = \mathbf{1}$.*

Proof. If it did, then, since it contains **Q** and therefore proves $\mathbf{0} \neq \mathbf{1}$, it would prove $\neg\text{PRV}_{PA}(\ulcorner \mathbf{0} = \mathbf{1} \urcorner)$, contradicting G2. □

This is somewhat surprising. One might have thought, for example, that PA, say, should prove $\text{PRV}_{PA}(\ulcorner A \urcorner) \rightarrow A$, for *all* sentences A and, in particular, for $\mathbf{0} = \mathbf{1}$. 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 proves 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. For suppose $\Sigma \vdash \neg \text{PRV}_\Sigma(\ulcorner \mathbf{0} = \mathbf{1} \urcorner)$. Then, trivially, $\Sigma \vdash \text{PRV}_\Sigma(\ulcorner \mathbf{0} = \mathbf{1} \urcorner) \rightarrow \mathbf{0} = \mathbf{1}$, and then Löb's Theorem implies that $\Sigma \vdash \mathbf{0} = \mathbf{1}$, 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 $C \rightarrow B$, 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:

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

and so, in particular:

$$\Sigma \vdash \textsc{Prv}_{\Sigma \cup \{C\}}(\ulcorner \mathbf{0} = \mathbf{1} \urcorner) \equiv \textsc{Prv}_{\Sigma}(\ulcorner C \to \mathbf{0} = \mathbf{1} \urcorner)$$

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

$$\Sigma \vdash \textsc{Prv}_{\Sigma \cup \{C\}}(\ulcorner \mathbf{0} = \mathbf{1} \urcorner) \to \textsc{Prv}_{\Sigma}(\ulcorner C \to \mathbf{0} = \mathbf{1} \urcorner)$$

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 $\boldsymbol{Q}$ that proves the derivability conditions. Suppose $\Sigma \vdash \textsc{Prv}_{\Sigma}(\ulcorner A \urcorner) \to A$, 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:

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

and so by logic:

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

However, since $\Sigma \vdash \neg \mathbf{0} = \mathbf{1}$, by logic, again:

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

so by (D1):

$$\Sigma \vdash \textsc{Prv}_{\Sigma}(\ulcorner (\neg A \to \mathbf{0} = \mathbf{1}) \to A \urcorner)$$

whence by (D2):

$$\Sigma \vdash \textsc{Prv}_{\Sigma}(\ulcorner \neg A \to \mathbf{0} = \mathbf{1} \urcorner) \to \textsc{Prv}_{\Sigma}(\ulcorner A \urcorner)$$

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

$$\Sigma \cup \{\neg A\} \vdash \textsc{Prv}_{\Sigma}(\ulcorner \neg A \to \mathbf{0} = \mathbf{1} \urcorner) \to \textsc{Prv}_{\Sigma}(\ulcorner A \urcorner)$$

and so by (1):

$$\Sigma \cup \{\neg A\} \vdash \neg \textsc{Prv}_{\Sigma}(\ulcorner \neg A \to \mathbf{0} = \mathbf{1} \urcorner)$$

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

$$\Sigma \vdash \textsc{Prv}_{\Sigma \cup \{\neg A\}}(\ulcorner \mathbf{0} = \mathbf{1} \urcorner) \to \textsc{Prv}_{\Sigma}(\ulcorner \neg A \to \mathbf{0} = \mathbf{1} \urcorner)$$

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

$$\Sigma \cup \{\neg A\} \vdash \neg \textsc{Prv}_{\Sigma \cup \{\neg A\}}(\ulcorner \mathbf{0} = \mathbf{1} \urcorner)$$

i.e.,

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

contrary to G2. □

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 $T(\mathbf{n})$ 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

Exercise 15.1. Prove Proposition 3.2.

Exercise 15.2. 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 15.3. Show that '$\forall x(0 + x = x)$' is not a theorem of $\boldsymbol{Q}$. (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 $\boldsymbol{Q}$ doesn't prove some other things.

Exercise 15.4. It follows from 5.1 that $\boldsymbol{Q}$ proves every true equality of the form '$t = u$', and every true inequality of the forms '$t \neq u$' and $t < u$, where t and u are arbitrary terms of the language of arithmetic. Prove this fact (by induction on the complexity of these terms).

Exercise 15.5. Prove proposition 5.4. (Hint: You will need to use an instance of the induction axiom.)

Exercise 15.6. Show that PA proves the commutativity of addition, i.e., that $\forall x \forall y(x + y = y + x)$. The easiest proof uses induction on y, so you want to show is that PA proves $\forall x(x + 0 = 0 + x)$ and $\forall x(x + n = n + x) \rightarrow \forall x(x + Sn = Sn + x)$.

Exercise 15.7. Show that condition (ii) in Definition 6.3 actually follows from condition (i).

Exercise 15.8. Prove Proposition 6.7.

Exercise 15.9. (Optional) Prove Theorem 6.8 using Theorem 4.6.

Exercise 15.10. Prove Corollary 8.2.

Exercise 15.11. 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) If $\Theta \supseteq \Sigma$ 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 15.12. Prove Proposition 13.4 in full generality.

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.