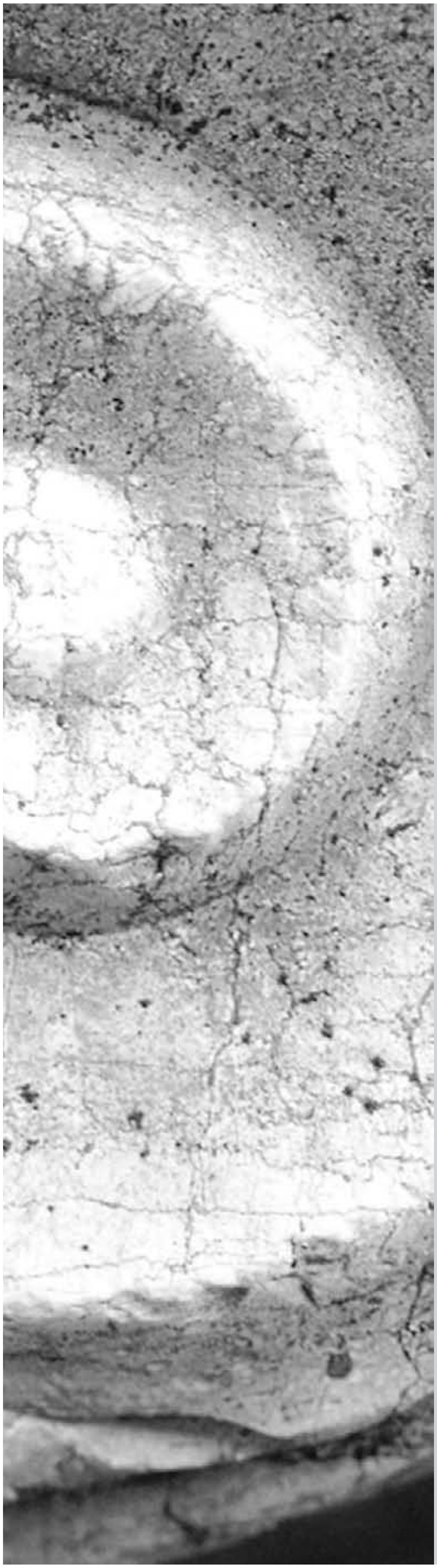


Chapter THREE

Undecidability

SECTION 3.0

Number Theory

In this chapter we will focus our attention on a specific language, the language of number theory. This will be the first-order language with equality and with the following parameters:

$\forall$, intended to mean "for all natural numbers." (Recall that the set $\mathbb{N}$ of natural numbers is the set $\{0, 1, 2, \ldots\}$. Zero is natural.)

$\mathbf{0}$, a constant symbol intended to denote the number 0.

$\mathbf{S}$, a one-place function symbol intended to denote the successor function $S : \mathbb{N} \to \mathbb{N}$, i.e., the function for which $S(n) = n + 1$.

$<$, a two-place predicate symbol intended to denote the usual (strict) ordering relation on $\mathbb{N}$.

$+, \cdot, \mathbf{E}$, two-place function symbols intended to denote the operations $+$, $\cdot$, and E of addition, multiplication, and exponentiation, respectively.

We will let $\mathfrak{N}$ be the intended structure for this language. Thus we may informally write

$$\mathfrak{N} = (\mathbb{N}; 0, S, <, +, \cdot, E).$$

(More precisely, $|\mathfrak{N}| = \mathbb{N}$, $\mathbf{0}^{\mathfrak{N}} = 0$, and so forth.)

By *number theory* we mean the theory of this structure, $\text{Th}\,\mathfrak{N}$. As warmup exercises we will study (in Sections 3.1 and 3.2) certain reducts of $\mathfrak{N}$, i.e., restrictions of $\mathfrak{N}$ to sublanguages:

$$\begin{aligned}\mathfrak{N}_S &= (\mathbb{N}; 0, S),\\ \mathfrak{N}_L &= (\mathbb{N}; 0, S, <),\\ \mathfrak{N}_A &= (\mathbb{N}; 0, S, <, +).\end{aligned}$$

Finally, in Section 3.8 we will consider

$$\mathfrak{N}_M = (\mathbb{N}; 0, S, <, +, \cdot).$$

For each of these structures we will raise the same questions:

(A) Is the theory of the structure decidable? If so, what is a nice set of axioms for the theory? Is there a finite set of axioms?

(B) What subsets of $\mathbb{N}$ are definable in the structure?

(C) What do the nonstandard models of the theory of the structure look like? (By "nonstandard" we mean "not isomorphic to the intended structure.")

Our reason for choosing number theory (rather than, say, group theory) for special study is this: We can show that a certain subtheory of number theory is an undecidable set of sentences. We will also be able to infer that any satisfiable theory that is at least as strong as this fragment of number theory (e.g., the full number theory or set theory) must be undecidable. In particular, such a theory cannot be both complete and axiomatizable.

In order to show that our subtheory of number theory is undecidable, we will show that it is strong enough to represent (in a sense to be made precise) facts about sequences of numbers, certain operations on numbers, and ultimately facts about decision procedures. This last feature then lets us perform a diagonal argument that demonstrates undecidability.

We could alternatively use, in place of a subtheory of number theory, some other theory (such as a fragment of the theory of finite sets) in which we could conveniently represent facts about decision procedures.

Before giving examples of the expressiveness of the language of number theory, it is convenient to introduce some notational conventions. As a concession to everyday usage, we will write

$$x < y, \quad x + y, \quad x \cdot y, \quad \text{and} \quad x\,\mathbf{E}\,y$$

in place of the official

$$<xy, \quad +xy, \quad \cdot xy, \quad \text{and} \quad \mathbf{E}xy.$$

For each natural number k we have a term $\mathbf{S}^k\mathbf{0}$ (the *numeral* for k) that denotes it:

$$\mathbf{S}^0\mathbf{0} = \mathbf{0}, \quad \mathbf{S}^1\mathbf{0} = \mathbf{S0}, \quad \mathbf{S}^2\mathbf{0} = \mathbf{SS0}, \quad \text{etc.}$$

(The set of numerals is generated from $\{\mathbf{0}\}$ by the operation of prefixing $\mathbf{S}$.) The fact that every natural number can be named in the language will be a useful feature.

Even though only countably many relations on $\mathbb{N}$ are definable in $\mathfrak{N}$, almost all the familiar relations are definable. For example, the set of primes is defined in $\mathfrak{N}$ by

$$v_1 \neq \mathbf{S}^1\mathbf{0} \wedge \forall\, v_2 \,\forall\, v_3(v_1 = v_2 \cdot v_3 \rightarrow v_2 = \mathbf{S}^1\mathbf{0} \vee v_3 = \mathbf{S}^1\mathbf{0}).$$

Later we will find it important to show that many other specific relations are definable in $\mathfrak{N}$.

One naturally expects the expressiveness of the language to be severely restricted when some of the parameters are omitted. For example, the set of primes, as we shall see, is not definable in $\mathfrak{N}_A$. On the other hand, in Section 3.8 we will show that any relation definable in $\mathfrak{N}$ is also definable in $\mathfrak{N}_M$.

Preview

The main theorems of this chapter — the theorems associated with the names of Gödel, Tarski, and Church — are proved in Section 3.5. But we can already sketch here some of the ideas involved. We want to compare the concepts of *truth* and *proof*; that is, we want to compare the set of sentences *true* in $\mathfrak{N}$ with the set of sentences that might be *provable* from an appropriate set A of axioms.

We can assign to each formula α of the language of number theory an integer $\sharp\alpha$, called the Gödel number of α. Any sufficiently straightforward way of assigning distinct integers to formulas would suffice for our purposes; a particular assignment is adopted at the beginning of Section 3.4. What *is* important is that from α we can effectively find the number $\sharp\alpha$, and conversely. Similarly, to each finite sequence D of formulas (such as a deduction) we assign an integer $\mathcal{G}(D)$. Note that for any *set* A of formulas, we can form the corresponding set $\{\sharp\alpha \mid \alpha \in A\}$ of numbers.

There are now three ways in which to proceed: the *self-reference approach*, the *diagonalization approach*, and the *computability approach*. It will be argued later, however, that the three approaches are more closely related than they appear — they are three aspects of one approach.

First, in the *self-reference* approach, we make a sentence σ that can be thought of as saying, "I am unprovable." More specifically, we have the following:

Theorem 30A Let $A \subseteq \operatorname{Th}\mathfrak{N}$ be a set of sentences true in $\mathfrak{N}$, and assume that the set $\{\sharp\alpha \mid \alpha \in A\}$ of Gödel numbers of members of A is a set definable in $\mathfrak{N}$. Then we can find a sentence σ such that σ is true in $\mathfrak{N}$ but σ is not deducible from A.

Proof. We will construct σ to express (in an indirect way) that σ itself is not a theorem of A. Then the argument will go roughly as follows: If $A \vdash \sigma$, then what σ says is false, contradicting the fact that A consists of true sentences. And so $A \nvdash \sigma$, whence σ is true.

To construct σ, we begin by considering the ternary relation R defined by

$\langle a, b, c\rangle \in R$ iff a is the Gödel number of some formula α and c is the value of $\mathcal{G}$ at some deduction from A of $\alpha(\mathbf{S}^b\mathbf{0})$.

Then because $\{\sharp\alpha \mid \alpha \in A\}$ is definable in $\mathfrak{N}$, it follows that R is definable also. (The details of this step must wait until later sections.) Let ρ be a formula that defines R in $\mathfrak{N}$. Let q be the Gödel number of

$$\forall \boldsymbol{v}_3 \neg \rho(\boldsymbol{v}_1, \boldsymbol{v}_1, \boldsymbol{v}_3).$$

(We use here the notation: $\varphi(t) = \varphi^{\boldsymbol{v}_1}_t$, $\varphi(t_1, t_2) = (\varphi^{\boldsymbol{v}_1}_{t_1})^{\boldsymbol{v}_2}_{t_2}$, and so forth.) Then let σ be

$$\forall \boldsymbol{v}_3 \neg \rho(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \boldsymbol{v}_3).$$

Thus σ says that no number is the value of $\mathcal{G}$ at a deduction from A of the result of replacing, in formula number q, the variable $\boldsymbol{v}_1$ by the numeral for q; i.e., no number is the value of $\mathcal{G}$ at a deduction of σ.

Suppose that, contrary to our expectations, there is a deduction of σ from A. Let k be the value of $\mathcal{G}$ at a deduction. Then $\langle q, q, k\rangle \in R$ and hence

$$\models_{\mathfrak{N}} \rho(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \mathbf{S}^k\mathbf{0}).$$

It is clear that

$$\sigma \vdash \neg \rho(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \mathbf{S}^k\mathbf{0})$$

and the two displayed lines tell us that σ is false in $\mathfrak{N}$. But $A \vdash \sigma$ and the members of A are true in $\mathfrak{N}$, so we have a contradiction.

Hence there is no deduction of σ from A. And so for every k, we have $\langle q, q, k\rangle \notin R$. Thus for every k

$$\models_{\mathfrak{N}} \neg \rho(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \mathbf{S}^k\mathbf{0}),$$

from which it follows (with the help of the substitution lemma) that

$$\models_{\mathfrak{N}} \forall \boldsymbol{v}_3 \neg \rho(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \boldsymbol{v}_3);$$

i.e., σ is true in $\mathfrak{N}$. ⊣

We will argue later — using something called Church's thesis — that any decidable set of natural numbers must be definable in $\mathfrak{N}$. The conclusion will then be that Th $\mathfrak{N}$ is not axiomatizable.

Corollary 30B The set $\{\sharp\tau \mid \models_{\mathfrak{N}} \tau\}$ of Gödel numbers of sentences true in $\mathfrak{N}$ is a set that is not definable in $\mathfrak{N}$.

Proof. If this set were definable, we could take $A = \mathrm{Th}\,\mathfrak{N}$ in the preceding theorem to obtain a contradiction. ⊣

Section 3.5 will follow the self-reference approach, but with a variation in which the sentence σ tries to say, "I am false." (The well-known liar paradox is relevant here!)

But if this "self-reference" construction seems too much like a magic trick, there is a second way to describe the situation: the *diagonalization approach*, which does not use an obvious self-reference.

We start by defining the following binary relation P on the natural numbers:

$$\langle a, b\rangle \in P \iff \begin{array}{l} a \text{ is the Gödel number of a formula } \alpha(\boldsymbol{v}_1) \\ \text{(with just } \boldsymbol{v}_1 \text{ free) and } \models_{\mathfrak{N}} \alpha(\mathbf{S}^b\mathbf{0}). \end{array}$$

(More informally, $\langle a, b\rangle \in P \Leftrightarrow$ "a is true of b.") Then any set of natural numbers that is definable in $\mathfrak{N}$ equals, for some a, the "vertical section"

$$P_a = \{b \mid \langle a, b\rangle \in P\}$$

of P. Namely, we take a to be the Gödel number of a formula defining the set, and use the fact that $\models_{\mathfrak{N}} \alpha(\mathbf{S}^b\mathbf{0}) \Leftrightarrow \models_{\mathfrak{N}} \alpha(\boldsymbol{v}_1)[\![b]\!]$.

So any definable (in $\mathfrak{N}$) set of natural numbers is somewhere on the list $P_1, P_2, \ldots$. Now we "diagonalize out" of the list. Define the set:

$$H = \{b \mid \langle b, b\rangle \notin P\}.$$

(More informally, $b \in H \Leftrightarrow$ "b is not true of b.") Then H is nowhere on the list $P_1, P_2, \ldots$. ($H \neq P_3$ because $3 \in H \Leftrightarrow 3 \notin P_3$, so the number 3 belongs to exactly one of these two sets and not to the other.) Therefore H is not definable in $\mathfrak{N}$.

Why is H undefinable? After all, we have above specified that

$$b \in H \iff \begin{array}{l} \text{not } [b \text{ is not the Gödel number of a formula } \alpha(\boldsymbol{v}_1) \\ \qquad \text{(with just } \boldsymbol{v}_1 \text{ free) and } \models_{\mathfrak{N}} \alpha(\mathbf{S}^b\mathbf{0})]. \end{array}$$

What is the barrier to translating this specification into the language of arithmetic? We will show that the barrier is *not* being the Gödel number of a formula — we can translate that — and the barrier is not having $\boldsymbol{v}_1$ free and not substituting the numeral $\mathbf{S}^b\mathbf{0}$ into a formula. By the process of elimination, we will show that the only possible barrier is saying of a sentence that is true in $\mathfrak{N}$.

Theorem 30C (a) The set $\{\sharp\tau \mid \models_{\mathfrak{N}}\tau\}$ of Gödel numbers of sentences true in $\mathfrak{N}$ is not definable in $\mathfrak{N}$.

*(b) The theory $\mathrm{Th}\,\mathfrak{N}$ is undecidable.

*(c) The theory $\mathrm{Th}\,\mathfrak{N}$ is not axiomatizable.

PROOF. Part (a), which is the same as Corollary 30B in the self-reference approach, has the diagonal proof sketched above. That is, if to the contrary Th $\mathfrak{N}$ were definable in $\mathfrak{N}$, then the above set H would also be definable, which it is not.

Part (b) will then follow, after we argue every decidable set of natural numbers must be definable in $\mathfrak{N}$. If Th $\mathfrak{N}$ were decidable, then the corresponding set $\{\sharp\tau \mid \models_{\mathfrak{N}} \tau\}$ of numbers would be decidable and hence definable in $\mathfrak{N}$, which it is not.

And part (c) is an immediate consequence of part (b) and Corollary 26I, since Th $\mathfrak{N}$ is a complete theory. ⊣

And thirdly, the *computability approach* presents us with a stark difference between what is true and what is provable. From Section 2.6 we know that whenever A is a decidable set (or even an effectively enumerable set) of axioms we might choose for Th $\mathfrak{N}$, the set Cn A of provable sentence will be an effectively enumerable set.

In contrast, the computability approach will show — using Church's thesis — that the set Th $\mathfrak{N}$ of all true sentences is *not* effectively enumerable. This fact, which is closely related to Theorem 30C, will follow from another diagonal argument in Section 3.6.

***THEOREM 30D** For any decidable (or even effectively enumerable) set A of axioms,

$$\text{Cn}\,A \neq \text{Th}\,\mathfrak{N}$$

because the set on the left is effectively enumerable and the set on the right is not.

Theorem 30D presents the dilemma: Either the axioms are lying to us by allowing us to deduce false sentences, or else the axioms are incomplete, in the sense that some true sentences cannot be deduced from those axioms.

This computability approach is implicit in parts of Section 3.5, but it is in Section 3.6 that the approach explicitly appears, and where it is compared with the other two approaches.

SECTION 3.1

Natural Numbers with Successor

We begin with a situation that is simple enough to let us give reasonably complete answers to our questions. We reduce the set of parameters to just $\forall$, $\mathbf{0}$, and $\mathbf{S}$, eliminating $<$, $+$, $\cdot$, and $\mathbf{E}$. The corresponding reduct of $\mathfrak{N}$ is

$$\mathfrak{N}_S = (\mathbb{N}; 0, S).$$

In this restricted language we still have the numerals, naming each point in $\mathbb{N}$. But the sentences we can express in the language are, from the viewpoint of arithmetic, uninteresting.

We want to ask about $\mathfrak{N}_S$ the same questions that interest us in the case of $\mathfrak{N}$. We want to know about the complexity of the set $\text{Th}\,\mathfrak{N}_S$; we want to study definability in $\mathfrak{N}_S$; and we want to survey the nonstandard models of $\mathfrak{N}_S$.

To study the theory of the natural numbers with successor ($\text{Th}\,\mathfrak{N}_S$), we begin by listing a few of its members, i.e., sentences true in $\mathfrak{N}_S$. (These sentences will ultimately provide an axiomatization for the theory.)

S1. $\forall x\, \mathbf{S}x \neq \mathbf{0}$, a sentence asserting that zero has no predecessor.

S2. $\forall x\, \forall y(\mathbf{S}x = \mathbf{S}y \rightarrow x = y)$. This asserts that the successor function is one-to-one.

S3. $\forall y(y \neq \mathbf{0} \rightarrow \exists x\; y = \mathbf{S}x)$. This asserts that any nonzero number is the successor of something.

S4.1 $\forall x\, \mathbf{S}x \neq x$.

S4.2 $\forall x\, \mathbf{SS}x \neq x$.

. . .

S4.n $\forall x\, \mathbf{S}^n x \neq x$, where the superscript n indicates that the symbol $\mathbf{S}$ occurs at n consecutive places.

Let A_S be the set consisting of the above sentences S1, S2, S3, S4.n ($n = 1, 2, \ldots$). Clearly these sentences are true in $\mathfrak{N}_S$; i.e., $\mathfrak{N}_S$ is a model of A_S. Hence

$$\text{Cn}\, A_S \subseteq \text{Th}\,\mathfrak{N}_S.$$

(Anything true in every model of A_S is true in this model.) What is not so obvious is that equality holds. We will prove this by considering arbitrary models of A_S.

What can be said of an arbitrary model

$$\mathfrak{A} = (|\mathfrak{A}|; \mathbf{0}^{\mathfrak{A}}, \mathbf{S}^{\mathfrak{A}})$$

of the axioms A_S? $\mathbf{S}^{\mathfrak{A}}$ must be a one-to-one map of $|\mathfrak{A}|$ onto $|\mathfrak{A}| - \{\mathbf{0}^{\mathfrak{A}}\}$, by S1, S2, and S3. And by S4.n, there can be no loops of size n. Thus $|\mathfrak{A}|$ must contain the "standard" points:

$$\mathbf{0}^{\mathfrak{A}} \rightarrow \mathbf{S}^{\mathfrak{A}}(\mathbf{0}^{\mathfrak{A}}) \rightarrow \mathbf{S}^{\mathfrak{A}}(\mathbf{S}^{\mathfrak{A}}(\mathbf{0}^{\mathfrak{A}})) \rightarrow \ldots,$$

which are all distinct. The arrow here indicates the action of $\mathbf{S}^{\mathfrak{A}}$. There may or may not be other points. If there is another point a in $|\mathfrak{A}|$, then there will be the successor of a, its successor, etc. Not only that, but since (by S3) each nonzero element has a predecessor (something of which it is the successor) which is (by S2) unique, $|\mathfrak{A}|$ must contain the predecessor of a, its predecessor, etc. These must all be distinct lest there be a finite loop. Thus a belongs to a "Z-chain":

$$\cdots * \to * \to a \to \mathbf{S}^{\mathfrak{A}}(a) \to \mathbf{S}^{\mathfrak{A}}(\mathbf{S}^{\mathfrak{A}}(a)) \to \cdots .$$

(We refer to these as Z-chains because they are arranged like the set $\mathbb{Z}$ of all integers $\{\ldots, -1, 0, 1, 2, \ldots\}$.) There can be any number of Z-chains. But any two Z-chains must be disjoint, as S2 prohibits merging. Similarly, any Z-chain must be disjoint from the standard part.

This can be restated in another way. Say that two points a and b in $|\mathfrak{A}|$ are *equivalent* if the function $\mathbf{S}^{\mathfrak{A}}$ can be applied a finite number of times to one point to yield the other point. This *is* an equivalence relation. (It is clearly reflexive and symmetric; the transitivity follows from the fact that $\mathbf{S}^{\mathfrak{A}}$ is one-to-one.) The standard part of $|\mathfrak{A}|$ is the equivalence class containing $\mathbf{0}^{\mathfrak{A}}$. For any other point (if any) a in $|\mathfrak{A}|$, the equivalence class of a is the set generated from $\{a\}$ by $\mathbf{S}^{\mathfrak{A}}$ and its inverse. This equivalence class is the Z-chain described above.

Conversely, any structure $\mathfrak{B}$ (for this language) that has a standard part

$$\mathbf{0}^{\mathfrak{B}} \to \mathbf{S}^{\mathfrak{B}}(\mathbf{0}^{\mathfrak{B}}) \to \mathbf{S}^{\mathfrak{B}}(\mathbf{S}^{\mathfrak{B}}(\mathbf{0}^{\mathfrak{B}})) \to \cdots$$

and a nonstandard part consisting of any number of separate Z-chains is a model of A_S. (Check through the list of axioms in A_S, and note that each is true in $\mathfrak{B}$.) We thus have a complete characterization of what the models of A_S must look like.

If a model $\mathfrak{A}$ of A_S has only countably many Z-chains, then $|\mathfrak{A}|$ is countable. In general, if the set of Z-chains has cardinality[1] λ, then altogether the number of points in $|\mathfrak{A}|$ is $\aleph_0 + \aleph_0 \cdot \lambda$. By facts of cardinal arithmetic (cf. Chapter 0) this number is the larger of $\aleph_0$ and λ. Hence

$$\operatorname{card}|\mathfrak{A}| = \begin{cases} \aleph_0 & \text{if } \mathfrak{A} \text{ has countably many Z-chains,} \\ \lambda & \text{if } \mathfrak{A} \text{ has an uncountable number } \lambda \text{ of Z-chains.} \end{cases}$$

Lemma 31A If $\mathfrak{A}$ and $\mathfrak{A}'$ are models of A_S having the same number of Z-chains, then they are isomorphic.

Proof. There is a unique isomorphism between the standard part of $\mathfrak{A}$ and the standard part of $\mathfrak{A}'$. By hypothesis we are given a one-to-one correspondence between the set of Z-chains of $\mathfrak{A}$ and the set of Z-chains of $\mathfrak{A}'$; thus each chain of $\mathfrak{A}$ is paired with a chain of $\mathfrak{A}'$. Clearly any two Z-chains are isomorphic. By combining all the individual isomorphisms (which uses the axiom of choice) we have an isomorphism of $\mathfrak{A}$ onto $\mathfrak{A}'$. ⊣

Thus a model of A_S is determined to within isomorphism by its number of Z-chains. For $\mathfrak{N}_S$ this number is zero, but any number is possible.

[1] To avoid uncountable cardinals, see Exercise 3.

The reader should note that there is no sentence of the language which says, "There are no Z-chains." In fact, there is no set Σ of sentences such that a model $\mathfrak{A}$ of A_S satisfies Σ iff $\mathfrak{A}$ has no Z-chains. For by the LST theorem there is an uncountable structure $\mathfrak{A}$ with $\mathfrak{A} \equiv \mathfrak{N}_S$. But $\mathfrak{A}$ has uncountably many Z-chains and $\mathfrak{N}_S$ has none.

Theorem 31B Let $\mathfrak{A}$ and $\mathfrak{B}$ be uncountable models of A_S of the same cardinality. Then $\mathfrak{A}$ is isomorphic to $\mathfrak{B}$.

Proof. By the above discussion, $\mathfrak{A}$ has card $\mathfrak{A}$ Z-chains, and $\mathfrak{B}$ has card $\mathfrak{B}$ Z-chains. Since card $\mathfrak{A}$ = card $\mathfrak{B}$, they have the same number of Z-chains and hence are isomorphic. ⊣

Theorem 31C Cn A_S is a complete theory.

Proof. Apply the Łoś–Vaught test of Section 2.6. The preceding theorem asserts that the theory Cn A_S is categorical in any uncountable power. Furthermore, A_S has no finite models. Hence the Łoś–Vaught test applies. ⊣

Corollary 31D Cn A_S = Th $\mathfrak{N}_S$.

Proof. We have Cn $A_S \subseteq$ Th $\mathfrak{N}_S$; the first theory is complete and the second is satisfiable. ⊣

***Corollary 31E** Th $\mathfrak{N}_S$ is decidable.

Proof. Any complete and axiomatizable theory is decidable (by Corollary 25G). A_S is a decidable set of axioms for this theory. ⊣

Elimination of Quantifiers

Once one knows a theory to be decidable, it is tempting to try to find a realistically practical decision procedure. We will give such a procedure for Th $\mathfrak{N}_S$, based on "elimination of quantifiers."

Definition. A theory T admits elimination of quantifiers iff for every formula φ there is a quantifier-free formula ψ such that

$$T \models (\varphi \leftrightarrow \psi).$$

Actually it is enough to consider only formulas ψ of a rather special form:

Theorem 31F Assume that for every formula φ of the form

$$\exists x(\alpha_0 \wedge \cdots \wedge \alpha_n),$$

where each α_i is an atomic formula or the negation of an atomic formula, there is a quantifier-free formula ψ such that $T \models (\varphi \leftrightarrow \psi)$. Then T admits elimination of quantifiers.

Proof. First we claim that we can find a quantifier-free equivalent for any formula of the form $\exists x\, \theta$ for quantifier-free θ. We begin by putting θ into disjunctive normal form (Corollary 15C). The resulting formula,

$$\exists x[(\alpha_0 \wedge \cdots \wedge \alpha_m) \vee (\beta_0 \wedge \cdots \wedge \beta_n) \vee \cdots \vee (\xi_0 \wedge \cdots \wedge \xi_t)],$$

is logically equivalent to

$$\exists x(\alpha_0 \wedge \cdots \wedge \alpha_m) \vee \exists x(\beta_0 \wedge \cdots \wedge \beta_n) \vee \cdots$$
$$\vee \exists x(\xi_0 \wedge \cdots \wedge \xi_t).$$

By assumption, each disjunct of this formula can be replaced by a quantifier-free formula.

We leave it to the reader to show (in Exercise 2) that by using the above paragraph one can obtain a quantifier-free equivalent for an arbitrary formula. ⊣

In the special case where the theory in question is the theory Th $\mathfrak{A}$ of a structure $\mathfrak{A}$, the definition can be restated: Th $\mathfrak{A}$ admits elimination of quantifiers iff for every formula φ there is a quantifier-free formula ψ such that φ and ψ are "equivalent in $\mathfrak{A}$"; i.e.,

$$\models_{\mathfrak{A}} (\varphi \leftrightarrow \psi)[s]$$

for any map s of the variables into $|\mathfrak{A}|$.

Theorem 31G Th $\mathfrak{N}_S$ admits elimination of quantifiers.

Proof. By the preceding theorem, it suffices to consider a formula

$$\exists x(\alpha_0 \wedge \cdots \wedge \alpha_q),$$

where each α_i is atomic or is the negation of an atomic formula. We will describe how to replace this formula by another that is quantifier-free. The equivalence of the new formula to the given one will, in fact, be a consequence of A_S; see Exercise 3.

In the language of $\mathfrak{N}_S$ the only terms are of the form $\mathbf{S}^k u$, where u is $\mathbf{0}$ or a variable. The only atomic formulas are equations. We may suppose that the variable x occurs in each α_i. For if x does not occur in α, then

$$\exists x(\alpha \wedge \beta) \models\!\!\!\dashv \alpha \wedge \exists x\, \beta.$$

Thus each α_i has the form

$$\mathbf{S}^m x = \mathbf{S}^n u$$

or the negation of this equation, where u is $\mathbf{0}$ or a variable. We may further suppose u is different from x, since $\mathbf{S}^m x = \mathbf{S}^n x$ could be replaced by $\mathbf{0} = \mathbf{0}$ if $m = n$, and by $\mathbf{0} \neq \mathbf{0}$ if $m \neq n$.

Case 1: Each α_i is the negation of an equation. Then the formula may be replaced by $\mathbf{0} = \mathbf{0}$. (Why?)

Case 2: There is at least one α_i not negated; say α_0 is

$$\mathbf{S}^m x = t,$$

where the term t does not contain x. Since the solution for x must be non-negative, we replace α_0 by

$$t \neq \mathbf{0} \wedge \cdots \wedge t \neq \mathbf{S}^{m-1}\mathbf{0}$$

(or by $\mathbf{0} = \mathbf{0}$ if $m = 0$). Then in each other α_j we replace, say,

$$\mathbf{S}^k x = u$$

first by

$$\mathbf{S}^{k+m} x = \mathbf{S}^m u,$$

which in turn becomes

$$\mathbf{S}^k t = \mathbf{S}^m u.$$

We now have a formula in which x no longer occurs, so the quantifier may be omitted. $\dashv$

There are several interesting by-products of the quantifier elimination procedure. For one, we get an alternative proof of the completeness of $\mathrm{Cn}\, A_S$. Suppose we begin with a sentence σ. The quantifier elimination procedure gives a quantifier-free *sentence* τ such that (by Exercise 3) $A_S \models (\sigma \leftrightarrow \tau)$. Now we claim that either $A_S \models \tau$ or $A_S \models \neg\tau$. For τ is built up from atomic sentences by means of $\neg$ and $\rightarrow$. An atomic sentence must be of the form $\mathbf{S}^k\mathbf{0} = \mathbf{S}^l\mathbf{0}$ and is deducible from A_S if $k = l$, but is refutable (i.e., its negation is deducible) from A_S if $k \neq l$. (In fact, just {S1, S2} suffices for this.) Since every atomic sentence can be deduced or refuted, so can every quantifier-free sentence. This establishes the claim. And so either $A_S \models \sigma$ or $A_S \models \neg\sigma$.

Another by-product concerns the problem of definability in $\mathfrak{N}_S$; see Exercises 4 and 5. For any formula φ in which just $\mathbf{v}_1$ and $\mathbf{v}_2$ occur free we now can find a quantifier-free ψ (with the same variables free) such that

$$\mathrm{Th}\,\mathfrak{N}_S \models \forall\, \mathbf{v}_1 \forall\, \mathbf{v}_2 (\varphi \leftrightarrow \psi);$$

i.e.,

$$\models_{\mathfrak{N}_S} \forall\, \mathbf{v}_1 \forall\, \mathbf{v}_2 (\varphi \leftrightarrow \psi).$$

Thus the relation φ defined is also definable by a quantifier-free formula.

Exercises

1. Let A_S^* be the set of sentences consisting of S1, S2, and all sentences of the form

$$\varphi(\mathbf{0}) \to \forall\, \mathbf{v}_1(\varphi(\mathbf{v}_1) \to \varphi(\mathbf{S}\mathbf{v}_1)) \to \forall\, \mathbf{v}_1\, \varphi(\mathbf{v}_1),$$

where φ is a wff (in the language of $\mathfrak{N}_S$) in which no variable except $\mathbf{v}_1$ occurs free. Show that $A_S \subseteq \text{Cn}\, A_S^*$. Conclude that $\text{Cn}\, A_S^* = \text{Th}\, \mathfrak{N}_S$. (Here $\varphi(t)$ is by definition $\varphi_t^{\mathbf{v}_1}$. The sentence displayed above is called the *induction axiom* for φ.)

2. Complete the proof of Theorem 31F. *Suggestion*: Use induction.

3. The proof of quantifier elimination for $\text{Th}\, \mathfrak{N}_S$ showed how, given a formula φ, to find a quantifier-free ψ. Show that

$$A_S \models (\varphi \leftrightarrow \psi)$$

without using the completeness of $\text{Cn}\, A_S$. (This yields an alternative proof of the completeness of $\text{Cn}\, A_S$, not involving Z-chains or the Łoś–Vaught test.)

4. Show that a subset of $\mathbb{N}$ is definable in $\mathfrak{N}_S$ iff either it is finite or its complement (in $\mathbb{N}$) is finite.

5. Show that the ordering relation $\{\langle m, n\rangle \mid m < n \text{ in } \mathbb{N}\}$ is not definable in $\mathfrak{N}_S$. *Suggestion*: It suffices to show there is no quantifier-free definition of ordering. Call a relation $R \subseteq \mathbb{N} \times \mathbb{N}$ *linear* if it can be covered by a finite number of lines. Call R *colinear* if it is the complement of a linear relation. Show that any relation definable in $\mathfrak{N}_S$ is either linear or colinear. And that the ordering relation is neither linear nor colinear.

6. Show that $\text{Th}\, \mathfrak{N}_S$ is not finitely axiomatizable. *Suggestion*: Show that no finite subset of A_S suffices, and then apply Section 2.6.

SECTION 3.2

Other Reducts of Number Theory[1]

First let us add the ordering symbol $<$ to the language. The intended structure is

$$\mathfrak{N}_L = (\mathbb{N}; 0, S, <).$$

[1] This section may be omitted without disastrous effects.

We want to show that the theory of this structure is (like Th $\mathfrak{N}_S$) decidable and also admits elimination of quantifiers. But unlike Th $\mathfrak{N}_S$, it is finitely axiomatizable and is not categorical in any infinite cardinality.

As axioms of Th $\mathfrak{N}_L$ we will take the finite set A_L consisting of the six sentences listed below. Here $x \leq y$ is, of course, an abbreviation for $(x < y \vee x = y)$, and $x \nleq y$ abbreviates the negation of this formula.

$$\forall y \quad (y \neq \mathbf{0} \to \exists x\, y = \mathbf{S}x) \tag{S3}$$

$$\forall x \forall y \quad (x < \mathbf{S}y \leftrightarrow x \leq y) \tag{L1}$$

$$\forall x \quad x \nless \mathbf{0} \tag{L2}$$

$$\forall x \forall y \quad (x < y \vee x = y \vee y < x) \tag{L3}$$

$$\forall x \forall y \quad (x < y \to y \nless x) \tag{L4}$$

$$\forall x \forall y \forall z \quad (x < y \to y < z \to x < z) \tag{L5}$$

On the one hand, it is easy to see that all six axioms are true in $\mathfrak{N}_L$. Thus Cn $A_L \subseteq$ Th $\mathfrak{N}_L$. On the other hand, the opposite inclusion is not obvious, and requires proof. We begin by listing some consequence of these axioms.

(1) $A_L \vdash \forall x\, x < \mathbf{S}x$.

Proof. In L1 take y to be x. ⊣

(2) $A_L \vdash \forall x\, x \nless x$.

Proof. In L4 take y to be x. ⊣

(3) $A_L \vdash \forall x \forall y (x \nless y \leftrightarrow y \leq x)$ (trichotomy).

Proof. For "→" use L3. For "←" use L4 and (2). ⊣

(4) $A_L \vdash \forall x \forall y (x < y \leftrightarrow \mathbf{S}x < \mathbf{S}y)$.

Proof. From A_L we can deduce the biconditionals:

$$\begin{aligned} x < y &\leftrightarrow y \nleq x && \text{by (3);} \\ &\leftrightarrow y \nless \mathbf{S}x && \text{by L1;} \\ &\leftrightarrow \mathbf{S}x \leq y && \text{by (3);} \\ &\leftrightarrow \mathbf{S}x < \mathbf{S}y && \text{by L1.} \end{aligned}$$

⊣

(5) $A_L \vdash$ S1 and $A_L \vdash$ S2.

Proof. S1 follows from L2 and (1). S2 comes from (4) by use of L3 and (2). ⊣

(6) $A_L \vdash$ S4.n for $n = 1, 2, \ldots$.

Proof. This follows from (1) and (2), by using L5. ⊣

Thus any model $\mathfrak{A}$ of A_L is (when we ignore $<^{\mathfrak{A}}$) also a model of A_S. So it must consist of a standard part plus zero or more Z-chains. In addition, it is ordered by $<^{\mathfrak{A}}$.

Theorem 32A The theory Cn A_L admits elimination of quantifiers.

Proof. Again we consider a formula

$$\exists x(\beta_0 \wedge \cdots \wedge \beta_p),$$

where each β_i is atomic or the negation of an atomic formula. The terms are, as in Section 3.1, of the form $\mathbf{S}^k u$, where u is $\mathbf{0}$ or a variable. There are two possibilities for atomic formulas,

$$\mathbf{S}^k u = \mathbf{S}^l t \quad \text{and} \quad \mathbf{S}^k u < \mathbf{S}^l t.$$

1. We can eliminate the negation symbol. Replace $t_1 \not< t_2$ by $t_2 < t_1 \vee t_1 = t_2$ and replace $t_1 \neq t_2$ by $t_1 < t_2 \vee t_2 < t_1$. (This is justified by L3 and L4.) By regrouping the atomic formulas and noting that

$$\exists x(\varphi \wedge \psi) \models\!\!\!\dashv \exists x\, \varphi \wedge \exists x\, \psi,$$

we may again reach formulas of the form

$$\exists x(\alpha_0 \wedge \cdots \wedge \alpha_q),$$

where now each α_i is atomic.

2. We may suppose that the variable x occurs in each α_i. This is because if x does not occur in α, then

$$\exists x(\alpha \wedge \beta) \models\!\!\!\dashv \alpha \wedge \exists x\, \beta.$$

Furthermore, we may suppose that x occurs on only one side of the equality or inequality α_i. For $\mathbf{S}^k x = \mathbf{S}^l x$ can be dealt with as in Section 3.1. $\mathbf{S}^k x < \mathbf{S}^l x$ can be replaced by $\mathbf{0} = \mathbf{0}$ if $k < l$, and $\mathbf{0} \neq \mathbf{0}$ otherwise. (This is justified by L1 and L4.)

Case 1: Suppose that some α_i is an equality. Then we can proceed as in case 2 of the quantifier-elimination proof of Theorem 31G.

Case 2: Otherwise each α_i is an inequality. Then the formula can be rewritten

$$\exists x\left(\bigwedge_i t_i < \mathbf{S}^{m_i} x \wedge \bigwedge_j \mathbf{S}^{n_j} x < u_j\right).$$

(Here $\bigwedge_i$ indicates the conjunction of formulas indexed by i, so $\gamma_0 \wedge \gamma_1 \wedge \cdots \wedge \gamma_k$ can be abbreviated $\bigwedge_i \gamma_i$.) In the first conjunction, $\bigwedge_i t_i < \mathbf{S}^{m_i} x$, we have the lower bounds on x; in the second conjunction, $\bigwedge_j \mathbf{S}^{n_j} x < u_j$, we have the upper bounds. If the second conjunction is empty (i.e., if there are no upper bounds on x), then we can replace the formula by $\mathbf{0} = \mathbf{0}$. (Why?) If the first conjunction is empty (i.e., if there are no lower bounds on x), then we can replace the formula by

$$\bigwedge_j \mathbf{S}^{n_j}\mathbf{0} < u_j,$$

which asserts that zero satisfies the upper bounds. Otherwise, we rewrite the formula successively as

$$\exists x \bigwedge_{i,j} (t_i < \mathbf{S}^{m_i}x \wedge \mathbf{S}^{n_j}x < u_j). \tag{1}$$

$$\exists x \bigwedge_{i,j} (\mathbf{S}^{n_j}t_i < \mathbf{S}^{m_i+n_j}x < \mathbf{S}^{m_i}u_j). \tag{2}$$

$$\left(\bigwedge_{i,j} \mathbf{S}^{n_j+1}t_i < \mathbf{S}^{m_i}u_j\right) \wedge \bigwedge_j \mathbf{S}^{n_j}\mathbf{0} < u_j. \tag{3}$$

This last formula says "any lower bound plus one satisfies any upper bound, and furthermore zero satisfies any upper bound." This implies that there is a gap between the greatest lower bound and the least upper bound, whence there is a solution for x. The second part guarantees that the solution for x is not forced to be negative.

In each case, we have arrived at a quantifier-free version of the given formula. ⊣

Corollary 32B (a) Cn A_L is complete.
(b) Cn $A_L = $ Th $\mathfrak{N}_L$.
*(c) Th $\mathfrak{N}_L$ is decidable.

Proof. (a) The argument that followed the proof of Theorem 31G is applicable here also. (b) This follows from (a), since Cn $A_L \subseteq$ Th $\mathfrak{N}_L$ and Th $\mathfrak{N}_L$ is satisfiable. For (c), we can use the fact that any complete axiomatizable theory is decidable. But the quantifier elimination proof yields a more efficient decision procedure. ⊣

Corollary 32C A subset of $\mathbb{N}$ is definable in $\mathfrak{N}_L$ iff it is either finite or has finite complement.

Proof. Compare Exercise 4 of the preceding section. ⊣

On the other hand, $\mathfrak{N}_L$ has more definable binary relations than has $\mathfrak{N}_S$. For the ordering relation $\{\langle m, n\rangle \mid m < n\}$ is not definable in $\mathfrak{N}_S$, by Exercise 5 of the preceding section.

Corollary 32D The addition relation,

$$\{\langle m, n, p\rangle \mid m + n = p\},$$

is not definable in $\mathfrak{N}_L$.

Proof. If we could define addition, we could then define the set of even natural numbers. But this set is neither finite nor has finite complement. ⊣

Now suppose we augment the language by the addition symbol $+$. The intended structure is

$$\mathfrak{N}_A = (\mathbb{N}; 0, S, <, +).$$

The theory of this structure is also decidable, as we will prove shortly. But to keep matters from getting even more complicated, we will avoid listing any convenient set of axioms for the theory.

The nonstandard models of $\operatorname{Th}\mathfrak{N}_A$ must also be models of $\operatorname{Th}\mathfrak{N}_L$. So they have a standard part, followed by some Z-chains. But ordering among the Z-chains can no longer be arbitrary. Let $\mathfrak{A}$ be a nonstandard model of $\operatorname{Th}\mathfrak{N}_A$. The ordering $<^{\mathfrak{A}}$ induces a well-defined ordering on the set of Z-chains. (See Exercise 3.) We claim that there is no largest Z-chain, there is no smallest Z-chain, and there is between any two Z-chains another one. The reasons, in outline, can be stated simply: If a belongs to some Z-chain (i.e., is an infinite element of $\mathfrak{A}$), then $a +^{\mathfrak{A}} a$ is in a larger Z-chain. There must be some b such that $b +^{\mathfrak{A}} b$ is either a or its successor; b must be in a smaller Z-chain. If a_1 and a_2 belong to different Z-chains, then there must be some b such that $b+^{\mathfrak{A}} b$ is either $a_1 +^{\mathfrak{A}} a_2$ or its successor. And b will lie in a Z-chain between that of a_1 and that of a_2. (These statements should seem quite plausible. The reader who enjoys working with infinite numbers might supply some details.)

***Theorem 32E (Presburger, 1929)** The theory of the structure $\mathfrak{N}_A = (\mathbb{N}; 0, S, <, +)$ is decidable.

The proof is again based on a quantifier elimination procedure. The theory of $\mathfrak{N}_A$ itself does *not* admit elimination of quantifiers. For example, the formula defining the set of even numbers

$$\exists y\, \boldsymbol{v}_1 = y + y$$

is not equivalent to any quantifier-free formula. We can overcome this by adding a new symbol $\equiv_2$ for congruence modulo 2. Similarly, we add symbols $\equiv_3, \equiv_4, \ldots$. The intended structure for this expanded language is

$$\mathfrak{N}^{\equiv} = (\mathbb{N}; 0, S, <, +, \equiv_2, \equiv_3, \ldots),$$

where $\equiv_k$ is the binary relation of congruence modulo k. It turns out that the theory of this structure does admit elimination of quantifiers.

This by itself does not imply that the theory of either structure is decidable. After all, we can start with *any* structure, and expand it to a structure having additional relations until a structure is obtained that admits elimination of quantifiers. To obtain decidability, we must show that we can, given a sentence σ, (1) effectively find a quantifier-free equivalent σ', and then (2) effectively decide if σ' is true.

We will now give the quantifier elimination procedure for $\mathrm{Th}\,\mathfrak{N}^{\equiv}$. For a term t and a natural number n, let nt be the term $t + t + \cdots + t$, with n summands. $0t$ is $\mathbf{0}$. Then any term can be expanded to one of the form

$$\mathbf{S}^{n_0}\mathbf{0} + n_1 x_1 + \cdots + n_k x_k$$

for $k \geq 0$, $n_i \geq 0$ (where the x_i's are variables). For example,

$$\mathbf{S}(x + \mathbf{S0}) + \mathbf{S}y$$

becomes

$$\mathbf{S}^3\mathbf{0} + x + y.$$

As usual we begin with a formula $\exists\, y(\beta_1 \wedge \cdots \wedge \beta_n)$, where β_i is an atomic formula or the negation of one.

1. Eliminate negation. Replace $\neg(t_1 = t_2)$ by $(t_1 < t_2 \vee t_2 < t_1)$. Replace $\neg(t_1 < t_2)$ by $(t_1 = t_2 \vee t_2 < t_1)$. And replace $\neg(t_1 \equiv_m t_2)$ by

$$t_1 \equiv_m t_2 + \mathbf{S}^1\mathbf{0} \vee \cdots \vee t_1 \equiv_m t_2 + \mathbf{S}^{m-1}\mathbf{0}.$$

Then regroup into a disjunction of formulas of the form

$$\exists\, y(\alpha_1 \wedge \cdots \wedge \alpha_m),$$

where each α_i is atomic. We may further suppose, as before, that y occurs in each α_i, and in fact that α_i has one of the four forms

$$\begin{aligned} ny + t &= u, \\ ny + t &\equiv_m u, \\ ny + t &< u, \\ u &< ny + t, \end{aligned}$$

where u and t are terms not containing y. In what follows we will take the liberty of writing these formulas with a subtraction symbol:

$$\begin{aligned} ny &= u - t, \\ ny &\equiv_m u - t, \\ ny &< u - t, \\ u - t &< ny. \end{aligned}$$

These are merely abbreviations for the formulas without subtraction obtained by transposing terms.

For example, we might have at this point the formula

$$\exists\, y(w < 4y \wedge 2y < u \wedge 3y < v \wedge y \equiv_3 t),$$

where t, u, v, and w are terms not containing y.

2. Uniformize the coefficients of y. Let p be the least common multiple of the coefficients of y. Each atomic formula can be converted to one in which the coefficient of y is p, by "multiplying through" by the appropriate factor. This is obviously legitimate for equalities and inequalities. In the case of congruences one must remember to raise the modulus also:

$$a \equiv_m b \quad \text{iff} \quad ka \equiv_{km} kb.$$

In the example above p is 12, and we obtain

$$\exists\, y(3w < 12y \wedge 12y < 6u \wedge 12y < 4v \wedge 12y \equiv_{36} 12t).$$

3. Eliminate the coefficient of y. Replace py by x and add the new conjunct $x \equiv_p \mathbf{0}$. (In place of $\exists\, y \cdots 12y \cdots$ we can equally well have, "There exists a multiple x of 12 such that $\cdots x \cdots$.") Our example is now converted to

$$\exists\, x(3w < x \wedge x < 6u \wedge x < 4v \wedge x \equiv_{36} 12t \wedge x \equiv_{12} \mathbf{0}).$$

4. Special case. If one of the atomic formulas is an equality, $x + t = u$, then we can replace

$$\exists\, x\, \theta$$

by

$$\theta^x_{u-t} \wedge t \leq u.$$

Here replacement of x by "$u - t$" is the natural thing; we transpose terms to compensate for the absence of subtraction. For example,

$$(x \equiv_m v)^x_{u-t} \quad \text{is} \quad u \equiv_m v + t.$$

5. We may assume henceforth that $=$ does not occur. So we have a formula of the form

$$\begin{aligned}\exists\, x[&r_0 - s_0 < x \wedge \cdots \wedge r_{l-1} - s_{l-1} < x \\ &\wedge x < t_0 - u_0 \wedge \cdots \wedge x < t_{k-1} - u_{k-1} \\ &\wedge x \equiv_{m_0} v_0 - w_0 \wedge \cdots \wedge x \equiv_{m_{n-1}} v_{n-1} - w_{n-1}],\end{aligned}$$

where r_i, s_i, t_i, u_i, v_i, and w_i are terms not containing x. This can be abbreviated

$$\exists\, x \left[\bigwedge_{j<l} r_j - s_j < x \;\wedge\; \bigwedge_{i<k} x < t_i - u_i \;\wedge\; \bigwedge_{i<n} x \equiv_{m_i} v_i - w_i \right].$$

If there are no congruences (i.e., $n = 0$), then the formula asserts that there is a nonnegative space between the lower and upper bounds. We can replace the formula by the quantifier-free formula:

$$\bigwedge_{i<k}\bigwedge_{j<l}(r_j - s_j) + \mathbf{S0} < t_i - u_i \wedge \bigwedge_{i<k} \mathbf{0} < t_i - u_i .$$

Let M be the least common multiple of the moduli $m_0, \ldots, m_{n-1}$. Then $a + M \equiv_{m_i} a$. So as a increases, the pattern of residues of a modulo $m_0, \ldots, m_{n-1}$ has period M. Thus, in searching for a solution to the congruences, we need only search M consecutive integers.

We now have a formula that asserts the existence of a natural number which is not less than certain lower bounds $L_1, \ldots, L_l$ and which satisfies certain upper bounds and certain congruences. If there is such a solution, then one of the following is a solution:

$$\begin{array}{l} L_1, L_1 + 1, \ldots, L_1 + M - 1, \\ L_2, L_2 + 1, \ldots, L_2 + M - 1, \\ \ldots \\ L_l, L_l + 1, \ldots, L_l + M - 1, \\ 0, 1, \ldots, M - 1. \end{array}$$

(The last line is needed to cover the case in which every L_j is negative. To avoid having to treat this line as a special case, we will add a new lower bound of 0. That is, let $r_l = \mathbf{0}$ and $s_l = \mathbf{S0}$ so that

$$r_l - s_l < x$$

is a formula $\mathbf{0} < x + \mathbf{S0}$ asserting that x is nonnegative. We now have $l + 1$ lower bounds.)

Our formula (asserting the existence of a solution for x) can now be replaced by a quantifier-free disjunction that asserts that one of the numbers in the above matrix is a nonnegative solution:

$$\bigvee_{j\le l}\ \bigvee_{1\le q\le M}\left[\bigwedge_{i\le l} r_i - s_i < (r_j - s_j) + \mathbf{S}^q\mathbf{0} \right.$$
$$\wedge \bigwedge_{i<k}(r_j - s_j) + \mathbf{S}^q\mathbf{0} < t_i - u_i$$
$$\left. \wedge \bigwedge_{i<n}(r_j - s_j) + \mathbf{S}^q\mathbf{0} \equiv_{m_i} v_i - w_i \right].$$

In our continuing example we have, after adding the new lower bound on x,

$$\exists x(3w < x \ \wedge\ \mathbf{0} < x + \mathbf{S0} \ \wedge\ x < 6u \ \wedge\ x < 4v$$
$$\wedge\, x \equiv_{36} 12t \ \wedge\ x \equiv_{12} \mathbf{0}).$$

The quantifier-free equivalent is a disjunction of 72 conjunctions. Each conjunction has six constituents.

This proves half of the theorem. If we are given a sentence σ, the above procedure tells us how to find effectively a quantifier-free sentence τ (in the language of $\mathfrak{N}^{\equiv}$) that is true (in the intended structure) iff σ is. Now we must decide if τ is true.

But this is easy. It is enough to look at atomic sentences. Any variable-free term can be put in the form $\mathbf{S}^n\mathbf{0}$. Then, for example,

$$\mathbf{S}^n\mathbf{0} \equiv_m \mathbf{S}^p\mathbf{0}$$

is true iff $n \equiv_m p$. ⊣

Thus we have a decision procedure for $\text{Th}\,\mathfrak{N}_A$. In 1974 Michael Fischer and Michael Rabin showed, however, that there is no decision procedure that is fast enough to be feasible for very long formulas.

A set D of natural numbers is said to be *periodic* if for some positive p, any number n is in D iff $n + p$ is in D. D is *eventually periodic* iff there exist positive numbers M and p such that for all n greater than M, $n \in D$ iff $n + p \in D$.

Theorem 32F A set of natural numbers is definable in $(\mathbb{N}; 0, S, <, +)$ iff it is eventually periodic.

Proof. Exercise 1 asserts that every eventually periodic set is definable. Conversely, suppose D is definable. Then D is definable in $\mathfrak{N}^{\equiv}$ by a quantifier-free formula (whose only variable is $\boldsymbol{v}_1$). Since the class of eventually periodic sets is closed under union, intersection, and complementation, it suffices to show that every atomic formula in the language of $\mathfrak{N}^{\equiv}$ whose only variable is $\boldsymbol{v}_1$ defines an eventually periodic set. There are only four possibilities:

$$\begin{aligned} n\boldsymbol{v}_1 + t &= u, \\ n\boldsymbol{v}_1 + t &< u, \\ u &< n\boldsymbol{v}_1 + t, \\ n\boldsymbol{v}_1 + t &\equiv_m u, \end{aligned}$$

where u and t are numerals. The first two formulas define finite sets (which eventually have period 1), the third defines a set with finite complement, and the last formula defines a periodic set with period m. ⊣

Corollary 32G The multiplication relation

$$\{\langle m, n, p\rangle \mid p = m \cdot n \text{ in } \mathbb{N}\}$$

is not definable in $(\mathbb{N}; 0, S, <, +)$.

Proof. If we had a definition of multiplication, we could then use that to define the set of squares. But the set of squares is not eventually periodic. ⊣

Exercises

1. Show that any eventually periodic set of natural numbers is definable in the structure $\mathfrak{N}_A$.

2. Show that in the structure $(\mathbb{N}; +)$ the following relations are definable:
   (a) Ordering, $\{\langle m, n\rangle \mid m < n\}$.
   (b) Zero, $\{0\}$.
   (c) Successor, $\{\langle m, n\rangle \mid n = S(m)\}$.

3. Let $\mathfrak{A}$ be a model of $\operatorname{Th}\mathfrak{N}_L$ (or equivalently a model of A_L). For a and b in $|\mathfrak{A}|$ define the equivalence relation:

   $a \sim b \iff$ $\mathbf{S}^{\mathfrak{A}}$ can be applied a finite number of times to one of a, b to reach the other.

   Let $[a]$ be the equivalence class to which a belongs. Order equivalence classes by

   $$[a] \prec [b] \quad \text{iff} \quad a <^{\mathfrak{A}} b \quad \text{and} \quad a \not\sim b.$$

   Show that this is a well-defined ordering on the set of equivalence classes.

4. Show that the theory of the real numbers with its usual ordering, $\operatorname{Th}(\mathbb{R}; <)$, admits elimination of quantifiers. (Assume that the language includes equality.)

SECTION 3.3

A Subtheory of Number Theory

We now return to the full language of number theory, as described in Section 3.0. The parameters of the language are $\forall$, $\mathbf{0}$, $\mathbf{S}$, $<$, $+$, $\cdot$, and $\mathbf{E}$. The intended structure for this language is

$$\mathfrak{N} = (\mathbb{N}; 0, S, <, +, \cdot, E).$$

Actually in $(\mathbb{N}; \cdot, E)$ we can define $\{0\}$, S, $<$, and $+$. (See Exercise 1.) As we will show in Section 3.8, in $(\mathbb{N}; +, \cdot)$ we can define E as well as 0, S, and $<$. So there are ways in which we could economize. The luxury of having all these parameters (particularly $\mathbf{E}$) will simplify some of the proofs.

As we shall see, $\operatorname{Th}\mathfrak{N}$ is a very strong theory and is neither decidable nor axiomatizable. In order to prove this fact (and a number of related results), it will be strategically wise to select for study a finitely axiomatizable subtheory of $\operatorname{Th}\mathfrak{N}$. As hinted at in Section 3.0, this subtheory should be strong enough to represent (in a sense to be made precise) facts about decidable sets. The subtheory we have selected is $\operatorname{Cn} A_E$, where A_E is the set consisting of the eleven sentences listed below. (As in the preceding section, $x \leq y$ abbreviates $x < y \vee x = y$.)

Set A_E of Axioms

$$\forall x \quad \mathbf{S}x \neq \mathbf{0} \tag{S1}$$

$$\forall x \forall y \quad (\mathbf{S}x = \mathbf{S}y \rightarrow x = y) \tag{S2}$$

$$\forall x \forall y \quad (x < \mathbf{S}y \leftrightarrow x \leq y) \tag{L1}$$

$$\forall x \quad x \not< \mathbf{0} \tag{L2}$$

$$\forall x \forall y \quad (x < y \vee x = y \vee y < x) \tag{L3}$$

$$\forall x \quad x + \mathbf{0} = x \tag{A1}$$

$$\forall x \forall y \quad x + \mathbf{S}y = \mathbf{S}(x + y) \tag{A2}$$

$$\forall x \quad x \cdot \mathbf{0} = \mathbf{0} \tag{M1}$$

$$\forall x \forall y \quad x \cdot \mathbf{S}y = x \cdot y + x \tag{M2}$$

$$\forall x \quad x\mathbf{E0} = \mathbf{S0} \tag{E1}$$

$$\forall x \forall y \quad x\,\mathbf{E}\,\mathbf{S}y = x\,\mathbf{E}\,y \cdot x \tag{E2}$$

Since $\mathfrak{N}$ is a model of A_E, we have $\mathrm{Cn}\, A_E \subseteq \mathrm{Th}\,\mathfrak{N}$. But (as we will prove in Section 3.5) equality does not hold here. In fact, it can be shown that $A_E \nvdash$ S3, where S3 is the sentence $\forall y(y \neq \mathbf{0} \rightarrow \exists x \;\, y = \mathbf{S}x)$.

The first five axioms give us *some*, but not all, of the axioms regarding $\mathbf{S}$ and $<$ that were useful in the preceding sections. The other six axioms are the "recursion" equations for addition, multiplication, and exponentiation.

We first show that certain simple sentences in $\mathrm{Th}\,\mathfrak{N}$ are deducible from A_E.

Lemma 33A (a) $A_E \vdash \forall x\, x \not< \mathbf{0}$.

(b) For any natural number k,

$$A_E \vdash \forall x(x < \mathbf{S}^{k+1}\mathbf{0} \leftrightarrow x = \mathbf{S}^0\mathbf{0} \vee \cdots \vee x = \mathbf{S}^k\mathbf{0}).$$

Notice that (a) can be thought of as the $k = -1$ case of (b), where the empty disjunction is $\perp$. The lemma tells us that A_E "knows" that the numbers less than 7, for example, are exactly 0, 1, 2, 3, 4, 5, 6. So in any model of A_E, the standard points — the ones denoted by numerals $S^k\mathbf{0}$ — are ordered in the natural way, and (by L3) the infinite points, if any, are all larger than any standard point.

Proof. Part (a) is L2. For (b) we use induction (in English) on k.

We have as a consequence of L1,

$$x < \mathbf{S0} \leftrightarrow x < \mathbf{0} \vee x = \mathbf{0},$$

which together with L2 gives

$$x < \mathbf{S0} \leftrightarrow x = \mathbf{0},$$

which is the $k = 0$ case of (b). For the inductive step we again apply L1:

$$x < \mathbf{S}^{k+1}\mathbf{0} \leftrightarrow x < \mathbf{S}^{k}\mathbf{0} \vee x = \mathbf{S}^{k}\mathbf{0}.$$

By the inductive hypothesis, $x < \mathbf{S}^{k}\mathbf{0}$ can be replaced by

$$x = \mathbf{S}^{0}\mathbf{0} \vee \cdots \vee x = \mathbf{S}^{k-1}\mathbf{0},$$

whereby we obtain (b). ⊣

Lemma 33B For any variable-free term t, there is a unique natural number n such that

$$A_E \vdash t = \mathbf{S}^{n}\mathbf{0}.$$

Proof. The uniqueness is immediate. (Why? Because A_E, weak as it is, at least knows, by S1, that $7 \neq 0$, and by S2 80 times, that $87 \neq 80$.) For the existence half, we use induction on t. If t is $\mathbf{0}$, we take $n = 0$. If t is $\mathbf{S}u$, then by the inductive hypothesis $A_E \vdash u = \mathbf{S}^{m}\mathbf{0}$ for some m. Hence $A_E \vdash t = \mathbf{S}^{m+1}\mathbf{0}$.

Now suppose t is $u_1 + u_2$. By the inductive hypothesis $A_E \vdash t = \mathbf{S}^{m}\mathbf{0} + \mathbf{S}^{n}\mathbf{0}$ for some m and n. We now apply A2 n times and A1 once to obtain $A_E \vdash t = \mathbf{S}^{m+n}\mathbf{0}$. The arguments for multiplication and exponentiation are similar. ⊣

As a special case of this lemma we have "$2 + 2 = 4$" (i.e., $\mathbf{S}^{2}\mathbf{0} + \mathbf{S}^{2}\mathbf{0} = \mathbf{S}^{4}\mathbf{0}$) as a consequence of A_E. A_E is at least smart enough to evaluate variable-free terms. And the proof shows more than this. The proof provides exact instructions for how, given such a term t, to find effectively the unique number n such that $A_E \vdash t = \mathbf{S}^{n}\mathbf{0}$.

Theorem 33C For any quantifier-free sentence τ true in $\mathfrak{N}$, $A_E \vdash \tau$.

Proof. Exercise 2. Start with the atomic sentences; these will be of the form $t_1 = t_2$ or $t_1 < t_2$ for variable-free terms t_1 and t_2. Show that A_E proves τ if τ is true in $\mathfrak{N}$, and refutes τ (i.e., proves $\neg\,\tau$) if τ is false in $\mathfrak{N}$. ⊣

Later on, we will improve on Theorem 33C by allowing τ to contain "bounded quantifiers"; see Theorem 33I.

A simplified notation (used earlier in Section 2.7) for substitution will be helpful in the coming pages:

$$\varphi(t) = \varphi^{\boldsymbol{v}_1}_{t},$$
$$\varphi(t_1, t_2) = \left(\varphi^{\boldsymbol{v}_1}_{t_1}\right)^{\boldsymbol{v}_2}_{t_2},$$

and so forth. Thus $\varphi = \varphi(\boldsymbol{v}_1) = \varphi(\boldsymbol{v}_1, \boldsymbol{v}_2)$. Usually the term substituted will be a numeral, for example

$$\varphi(\mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}) = \left(\varphi^{v_1}_{\mathbf{S}^a\mathbf{0}}\right)^{v_2}_{\mathbf{S}^b\mathbf{0}}.$$

But at times we will also substitute other terms, e.g., $\varphi(x) = \varphi^{v_1}_x$, where x is a variable. If, however, x is not substitutable for v_1 in φ, then we must take $\varphi(x) = \psi^{v_1}_x$, where ψ is a suitable alphabetic variant of φ.

In the next proof (and elsewhere in this chapter) we make use of the following consequence of the substitution lemma of Section 2.5: For a formula φ in which at most $v_1, \ldots, v_n$ occur free and for natural numbers $a_1, \ldots, a_n$,

$$\models_{\mathfrak{N}} \varphi[\![a_1, \ldots, a_n]\!] \Leftrightarrow \models_{\mathfrak{N}} \varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}).$$

An existential ($\exists_1$) formula is one of the form $\exists x_1 \cdots \exists x_k \theta$, where θ is quantifier-free. The following result improves Theorem 33C:

Corollary 33D If τ is an existential sentence true in $\mathfrak{N}$, then $A_E \vdash \tau$.

Proof. If $\exists v_1 \exists v_2 \theta$ is true in $\mathfrak{N}$, then for some natural numbers m and n, $\theta(\mathbf{S}^m\mathbf{0}, \mathbf{S}^n\mathbf{0})$ is true in $\mathfrak{N}$. As this is a quantifier-free true sentence, it is deducible from A_E. But it in turn logically implies $\exists v_1 \exists v_2 \theta$. ⊣

On the other hand, it is known that there are true universal ($\forall_1$) sentences (i.e., of the form $\forall x_1 \cdots \forall x_k \theta$ for quantifier-free θ) that are *not* in $\operatorname{Cn} A_E$.

Representable Relations

Let R be an m-ary relation on $\mathbb{N}$; i.e., $R \subseteq \mathbb{N}^m$. We know that a formula ρ (in which only $v_1, \ldots, v_m$ occur free) defines R in $\mathfrak{N}$ iff for every $a_1, \ldots, a_m$ in $\mathbb{N}$,

$$\begin{aligned} \langle a_1, \ldots, a_m \rangle \in R &\Leftrightarrow \models_{\mathfrak{N}} \rho[\![a_1, \ldots, a_m]\!] \\ &\Leftrightarrow \models_{\mathfrak{N}} \rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}). \end{aligned}$$

(The last condition here is equivalent to the preceding one by the substitution lemma.) We can recast this into two implications:

$$\begin{aligned} \langle a_1, \ldots, a_m \rangle \in R &\Rightarrow \models_{\mathfrak{N}} \rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}), \\ \langle a_1, \ldots, a_m \rangle \notin R &\Rightarrow \models_{\mathfrak{N}} \neg \rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}). \end{aligned}$$

We will say that ρ also represents R in the theory $\operatorname{Cn} A_E$ if in these two implications the notion of truth in $\mathfrak{N}$ can be replaced by the stronger notion of deducibility from A_E.

More generally, let T be any theory in a language with $\mathbf{0}$ and $\mathbf{S}$. Then ρ *represents* R in T iff for every $a_1, \ldots, a_m$ in $\mathbb{N}$:

$$\begin{aligned} \langle a_1, \ldots, a_m \rangle \in R &\Rightarrow \rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}) \in T, \\ \langle a_1, \ldots, a_m \rangle \notin R &\Rightarrow (\neg \rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0})) \in T. \end{aligned}$$

For example, ρ represents R in the theory $\operatorname{Th}\mathfrak{N}$ iff ρ defines R in $\mathfrak{N}$. But ρ represents R in $\operatorname{Cn} A_E$ iff for all $a_1, \ldots, a_m$:

$$\begin{aligned}
\langle a_1, \ldots, a_m \rangle \in R &\Rightarrow A_E \vdash \rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}),\\
\langle a_1, \ldots, a_m \rangle \notin R &\Rightarrow A_E \vdash \neg\,\rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}).
\end{aligned}$$

The equality relation on $\mathbb{N}$, for example, is represented in $\operatorname{Cn} A_E$ by the formula $\boldsymbol{v}_1 = \boldsymbol{v}_2$. For

$$\begin{aligned}
m = n &\Rightarrow\ \vdash \mathbf{S}^m\mathbf{0} = \mathbf{S}^n\mathbf{0},\\
m = n &\Rightarrow \{\mathrm{S1}, \mathrm{S2}\} \vdash \neg\,\mathbf{S}^m\mathbf{0} = \mathbf{S}^n\mathbf{0}.
\end{aligned}$$

A relation is *representable* in T iff there exists some formula that represents it in T.

The concept of representability should be compared with that of definability. In both cases we are somehow describing relations on the natural numbers by formulas. In the case of definability, we ask about the *truth* of sentences in the interpretation. In the case of representability in $\operatorname{Cn} A_E$, we ask instead about the *deducibility* of sentences from the axioms.

Say that a formula φ, in which no variables other than $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ occur free, is *numeralwise determined* by A_E iff for every m-tuple $a_1, \ldots, a_m$ of natural numbers, either

$$A_E \vdash \varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0})$$

or

$$A_E \vdash \neg\,\varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0})$$

Theorem 33E A formula ρ represents a relation R in $\operatorname{Cn} A_E$ iff

(1) ρ is numeralwise determined by A_E, and
(2) ρ defines R in $\mathfrak{N}$.

Proof. We use the fact that $\mathfrak{N}$ is a model of A_E. If ρ represents R in $\operatorname{Cn} A_E$, then it is clear that (1) holds; (2) holds since "$A_E \vdash$" implies "$\models_{\mathfrak{N}}$." Conversely, if (1) and (2) hold, then we have

$$\begin{aligned}
\langle a_1, \ldots, a_m \rangle \in R &\Rightarrow\ \models_{\mathfrak{N}} \rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}) && \text{by (2)}\\
&\Rightarrow A_E \nvdash \neg\,\rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}) && \text{since } \mathfrak{N} \text{ is a model}\\
& && \text{of } A_E\\
&\Rightarrow A_E \vdash \rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}) && \text{by (1).}
\end{aligned}$$

Similarly for the complement of R and $\neg\,\rho$. ⊣

Church's Thesis

We now turn to the relationship of the concepts of representability and decidability.

***Theorem 33F** Assume that R is a relation representable in a consistent axiomatizable theory. Then R is decidable.

Proof. Say that ρ represents R in the consistent axiomatizable theory T. Recall that T is effectively enumerable (Corollary 25F). The decision procedure is as follows:

Given $a_1, \ldots, a_m$, enumerate the members of T. If, in the enumeration, $\rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0})$ is found, then we are done and $\langle a_1, \ldots, a_m \rangle \in R$. If, in the enumeration, $\neg\, \rho(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0})$ is found, then we are done and $\langle a_1, \ldots, a_m \rangle \notin R$.

By the representability, one sentence or the other always appears eventually, whereupon the procedure terminates. Since T is consistent, the answer given by the procedure is correct. ⊣

*Corollary 33G Any relation representable in a consistent finitely axiomatizable theory is decidable.

What about the converse to the above corollary? We cannot really hope to *prove* the converse on the basis of our informal notion of decidability. For our informal approach is usable only for giving lower bounds on the class of decidable relations (i.e., for showing that certain relations *are* decidable) and is unsuited to giving upper bounds (i.e., for showing *undecidability*).

It is nevertheless possible to make plausibility arguments in support of the converse. This will be easier to do at the end of Section 3.4 than here. Roughly, the idea is that in a finite number of axioms we could capture the (finitely long) instructions for the decision procedure.

The assertion that both the above corollary and its converse are correct is generally known as *Church's thesis*. This assertion is not really a mathematical statement susceptible to proof or disproof; rather it is a judgment that the correct formalization of the informal notion of decidability is by means of representability in consistent and finitely axiomatizable theories.

Definition. A relation R on the natural numbers is *recursive* iff it is representable in some consistent finitely axiomatizable theory (in a language with $\mathbf{0}$ and $\mathbf{S}$).

Church's thesis now can be put more succinctly: A relation is decidable iff it is recursive. Or perhaps more accurately: The concept of recursiveness is the correct precise counterpart to the informal concept of decidability. The situation is analogous to one encountered in calculus. An intuitively continuous function (defined on an interval) is one whose graph you can draw without lifting your pencil off the paper. But to prove theorems, some formal counterpart of this notion is needed. And so one gives the usual definition of ε-δ-continuity. One should ask if the precise notion of ε-δ-continuity is an accurate formalization of intuitive continuity. If anything, the class of ε-δ-continuous functions is too broad. It includes nowhere differentiable functions, whose graphs cannot be drawn without lifting the pencil. But accurate or not, the class of ε-δ-continuous functions has been found to be a natural and important class in mathematical analysis.

Very much the same situation occurs with recursiveness. One should ask if the precise notion of recursiveness is an accurate formalization of the informal notion of decidability. Again, the precisely defined class (of recursive relations) appears to be, if anything, too broad. It includes relations for which any decision procedure will, for large inputs, require so much computing time and memory ("scratchpad") space as to make implementation absurd. Recursiveness corresponds to decidability in an idealized world, where length of computation and amount of memory are disregarded. But in any case, the class of recursive relations has been found to be a natural and important class in mathematical logic.

Empirical evidence that the class of recursive relations is not too narrow is provided by the following:

1. Any relation considered thus far that mathematicians have felt was decidable has been found to be recursive.

2. Several people have tried giving precise definitions of idealized computing agents. The best-known such idealized agents are the "Turing machines," introduced by Alan Turing in 1936. (A variation on that idea leads to the register machines described in Section 3.6.) The idea was to devise something that could carry out any effective procedure. In all cases, the class of relations having decision procedures executable by such a computing agent has been exactly the class of recursive relations. (Because of the importance of Turing's analysis of effective computability, Church's thesis is often called the *Church–Turing thesis*.)

The fact that so many different (yet equivalent) definitions for the class of recursive relations have been found is some indication of the naturalness and importance of the concept.

In this book we will continue to exclude the informal notion of decidability from nonstarred theorems. But in the remainder of the exposition we will accept Church's thesis. For example, we will speak of a set's being undecidable when we have a theorem stating it to be nonrecursive.

Obviously any relation representable in Cn A_E is recursive. We will prove later that the converse also holds; if a relation is representable in any consistent finitely axiomatizable theory, then it is representable in the one theory we have selected for special study. (This was, of course, a motivating factor in our selection.)

The use of the word "recursive" in this context is the result of historical accident — even of historical error. Recently several mathematicians have argued that the word "computable" would more accurately reflect the intended ideas. But in the present context, we want to reserve the word "computable" for an informal concept, to be defined next. For relations we have the informal concept of decidability; for functions the analogous concept is computability. (As notational shorthand, a string $a_1, \ldots, a_k$ can be written as $\vec{a}$.)

*DEFINITION. A function $f : \mathbb{N}^k \to \mathbb{N}$ is *computable* iff there is an effective procedure that, given any k-tuple $\vec{a}$ of natural numbers, will produce $f(\vec{a})$.

For example, addition and multiplication are computable. Effective procedures, using base-10 notation, for these functions are taught in the elementary schools. (Strictly speaking, in the concept of computability one should refer to being given *numerals*, not numbers. For it is numerals — strings of symbols like the triple 317 or the triple XCI — that can be communicated. Nonetheless, we will suppress this point.) On the other hand, of the uncountably many functions from $\mathbb{N}^k$ into $\mathbb{N}$, only countably many can be computable, because there are only countably many effective procedures.

We want to give a mathematical counterpart to the informal concept of computability, just as in the case of decidable relations. The clue to the correct counterpart is provided by the next theorem. Recall that any function $f : \mathbb{N}^k \to \mathbb{N}$ is also a $(k + 1)$-ary relation on $\mathbb{N}$:

$$\langle a_1, \ldots, a_k, b\rangle \in f \iff f(a_1, \ldots, a_k) = b.$$

At one time it was popular to distinguish between the function and the relation (which was called the *graph* of the function). Current set-theoretic usage takes a function to be the same thing as its graph. But we still have the two ways of looking at the function.

***THEOREM 33H** The following three conditions on a function $f : \mathbb{N}^k \to \mathbb{N}$ are equivalent:

(a) f is computable.

(b) When viewed as a relation, f is a decidable relation.

(c) When viewed as a relation, f is an effectively enumerable relation.

PROOF. (a) ⇒ (b): Assume that f is computable; we will describe the decision procedure. Given $\langle a_1, \ldots, a_k, b\rangle$, first compute $f(a_1, \ldots, a_k)$. Then look to see if the result is equal to b. If it is say "yes," otherwise say "no."

(b) ⇒ (c): Any decidable relation is effectively enumerable. For we can enumerate the set of all $(k+1)$-tuples of numbers, and place on the output list those which meet the test of belonging to the relation.

(c) ⇒ (a): Assume that we have an effective enumeration of (the graph of) f. To compute $f(a_1, \ldots, a_k)$ we examine the $(k+1)$-tuples in the enumeration until we find the one that begins with $a_1 \ldots, a_k$. Its last component is then the desired function value. ⊣

Thus by using Church's thesis, we can say that f is computable iff f (viewed as a relation) is recursive. The class of recursive functions is an interesting class even apart from its connection with incompleteness theorems of logic. It represents an upper bound to the class of functions that can actually be computed by programs for digital computers. The recursive functions are those which are calculable by digital computers, provided one ignores practical limitations on computing time and memory space.

We can now describe our plans for this section and the next. Our basic goal is to obtain the theorems of Section 3.5. But some groundwork is required before we can prove those theorems; we must verify that a number of relations (intuitively decidable) and a number of functions (intuitively computable) are representable in $\mathrm{Cn}\, A_E$ and hence are recursive. In the process we will show (Theorem 34A) that recursiveness is equivalent to representability in $\mathrm{Cn}\, A_E$. In the remainder of the present section we will establish general facts about representability, and will show, for example, that certain functions for encoding finite sequences of numbers into single numbers are representable. Then in Section 3.4 we apply these results to particular relations and functions related to the syntactical features of the formal language.

The author is sufficiently realistic to know that many readers will be more interested in the theorems of Section 3.5 than in the preliminary spadework. If the reader is willing to believe that intuitively decidable relations are all representable in $\mathrm{Cn}\, A_E$, and intuitively computable functions are functionally representable (a concept we will define shortly) there, then most if not all of the proofs in this spadework become unnecessary. But it is hoped that the definitions and the statements of the results will still receive some attention.

Numeralwise Determined Formulas

Theorem 33E tells us that we can show a relation to be representable in $\mathrm{Cn}\, A_E$ by finding a formula that defines it in $\mathfrak{N}$ and is numeralwise determined by A_E. The next theorem will be useful in establishing numeralwise determination.

Theorem 33I (a) Any atomic formula is numeralwise determined by A_E.

(b) If φ and ψ are numeralwise determined by A_E, then so are $\neg\varphi$ and $\varphi \rightarrow \psi$.

(c) If φ is numeralwise determined by A_E, then so are the following formulas (obtained from φ by "bounded quantification"):

$$\forall x(x < y \rightarrow \varphi),$$
$$\exists x(x < y \wedge \varphi).$$

Proof. Part (a) follows from Theorem 33C. Part (b) is easy. It remains to prove part (c). We will consider a formula

$$\exists x(x < y \wedge \varphi(x, y, z))$$

in which just the variables y and z occur free. Consider two natural numbers a and b; we must show that either

$$\boldsymbol{A}_E \vdash \exists x(x < \mathbf{S}^a\mathbf{0} \wedge \varphi(x, \mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}))$$

or

$$A_E \vdash \neg \exists x(x < \mathbf{S}^a\mathbf{0} \wedge \varphi(x, \mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0})).$$

Case 1: For some c less than a,

$$A_E \vdash \varphi(\mathbf{S}^c\mathbf{0}, \mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}). \tag{1}$$

(This case occurs iff $\exists x(x < \mathbf{S}^a\mathbf{0} \wedge \varphi(x, \mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}))$ is true in $\mathfrak{N}$.) We also have

$$A_E \vdash \mathbf{S}^c\mathbf{0} < \mathbf{S}^a\mathbf{0}. \tag{2}$$

And the sentences in (1) and (2) logically imply the sentence

$$\exists x(x < \mathbf{S}^a\mathbf{0} \wedge \varphi(x, \mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0})).$$

Case 2: Otherwise for every c less than a,

$$A_E \vdash \neg \varphi(\mathbf{S}^c\mathbf{0}, \mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}). \tag{3}$$

(This case occurs iff $\forall x(x < \mathbf{S}^a\mathbf{0} \rightarrow \neg \varphi(x, \mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}))$ is true in $\mathfrak{N}$.) We know from Lemma 33A that

$$A_E \vdash \forall x(x < \mathbf{S}^a\mathbf{0} \rightarrow x = \mathbf{S}^0\mathbf{0} \vee \cdots \vee x = \mathbf{S}^{a-1}\mathbf{0}). \tag{4}$$

The sentence in (4) together with the sentences in (3) (for $c = 0, \ldots, a-1$) logically imply

$$\forall x(x < \mathbf{S}^a\mathbf{0} \rightarrow \neg \varphi(x, \mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0})).$$

And this is equivalent to

$$\neg \exists x(x < \mathbf{S}^a\mathbf{0} \wedge \varphi(x, \mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0})).$$

This shows that $\exists x(x < y \wedge \varphi(x, y, z))$ is numeralwise determined by A_E. By applying this result to $\neg \varphi$ we obtain the fact that the dual formula, $\forall x(x < y \rightarrow \varphi(x, y, z))$, is numeralwise determined by A_E as well. ⊣

The argument in case 2 relied on the fact that the x quantifier was bounded by $\mathbf{S}^a\mathbf{0}$. We will see that it is possible for

$$\neg \psi(\mathbf{S}^0\mathbf{0}), \neg \psi(\mathbf{S}^1\mathbf{0}), \ldots$$

all to be consequences of A_E without having

$$\forall x \neg \psi(x)$$

be a consequence.

The preceding theorem is a useful tool for showing many relations to be representable in Cn A_E. For example, the set of primes is represented by

$$\mathbf{S}^1\mathbf{0} < \boldsymbol{v}_1 \wedge \forall x(x < \boldsymbol{v}_1 \rightarrow \forall y(y < \boldsymbol{v}_1 \rightarrow x \cdot y \neq \boldsymbol{v}_1)).$$

This formula defines the primes in $\mathfrak{N}$, and by the preceding theorem is numeralwise determined by A_E. It therefore represents the set of primes in Cn A_E.

Representable Functions

Often it is more convenient to work with functions than with relations. Let $f : \mathbb{N}^m \rightarrow \mathbb{N}$ be an m-place function on the natural numbers. A formula φ in which only $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{m+1}$ occur free will be said to *functionally represent* f (in the theory Cn A_E) iff for every $a_1, \ldots, a_m$ in $\mathbb{N}$,

$$A_E \vdash \forall \boldsymbol{v}_{m+1}\left[\varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}, \boldsymbol{v}_{m+1}) \leftrightarrow \boldsymbol{v}_{m+1} = \mathbf{S}^{f(a_1,\ldots,a_m)}\mathbf{0}\right].$$

(Observe that the "$\longleftarrow$" half of this sentence is equivalent to $\varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}, \mathbf{S}^{f(a_1,\ldots,a_m)}\mathbf{0})$. The "$\longrightarrow$" half adds an assertion of uniqueness.)

Theorem 33J If φ functionally represents f in Cn A_E, then it also represents f (as a relation) in Cn A_E.

Proof, with $m = 1$. Since φ functionally represents f, we have for any b:

$$A_E \vdash \varphi(\mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}) \leftrightarrow \mathbf{S}^b\mathbf{0} = \mathbf{S}^{f(a)}\mathbf{0}.$$

If $\langle a, b\rangle \in f$, i.e., if $f(a) = b$, then the right half of this biconditional is valid and we get

$$A_E \vdash \varphi(\mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}).$$

But otherwise the right half is refutable from A_E (i.e., its negation is deducible), whence

$$A_E \vdash \neg \varphi(\mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}). \qquad \dashv$$

The converse of this theorem is false. But we can change the formula:

Theorem 33K Let f be a function on $\mathbb{N}$ that is (as a relation) representable in Cn A_E. Then we can find a formula φ that functionally represents f in Cn A_E.

Proof. To simplify the notation we will take f to be a one-place function on $\mathbb{N}$. The desired sentence,

$$\forall\, v_2[\varphi(\mathbf{S}^a\mathbf{0},\, v_2) \leftrightarrow v_2 = \mathbf{S}^{f(a)}\mathbf{0}],$$

is equivalent to the conjunction of the two sentences

$$\varphi(\mathbf{S}^a\mathbf{0}, \mathbf{S}^{f(a)}\mathbf{0}) \tag{1}$$

and

$$\forall\, v_2[\varphi(\mathbf{S}^a\mathbf{0},\, v_2) \rightarrow v_2 = \mathbf{S}^{f(a)}\mathbf{0}]. \tag{2}$$

The sentence (1) is a theorem of A_E whenever φ represents f. The sentence (2) is an assertion of uniqueness; we must construct φ in such a way that this will also be a theorem of A_E.

Begin with a formula θ known to represent f (as a binary relation). Let φ be

$$\theta(v_1,\, v_2) \wedge \forall\, z(z < v_2 \rightarrow \neg\, \theta(v_1, z)).$$

We can then rewrite (2) as

$$\begin{aligned}\forall\, v_2[\theta(\mathbf{S}^a\mathbf{0},\, v_2) \wedge \forall\, z(z < v_2 \rightarrow \neg\, \theta(\mathbf{S}^a\mathbf{0}, z))\\ \rightarrow v_2 = \mathbf{S}^{f(a)}\mathbf{0}].\end{aligned} \tag{2$'$}$$

To show this to be a theorem of A_E it clearly suffices to show that

$$A_E \cup \{\theta(\mathbf{S}^a\mathbf{0},\, v_2), \forall\, z(z < v_2 \rightarrow \neg\, \theta(\mathbf{S}^a\mathbf{0}, z))\} \vdash v_2 = \mathbf{S}^{f(a)}\mathbf{0}.$$

Call this set of hypotheses (to the left of "$\vdash$") Γ. Since L3 $\in A_E$ it suffices to show that

$$\Gamma \vdash v_2 \not< \mathbf{S}^{f(a)}\mathbf{0} \tag{3}$$

and

$$\Gamma \vdash \mathbf{S}^{f(a)}\mathbf{0} \not< v_2. \tag{4}$$

It is easy to obtain (4), since from the last member of Γ we get

$$\mathbf{S}^{f(a)}\mathbf{0} < v_2 \rightarrow \neg\, \theta(\mathbf{S}^a\mathbf{0}, \mathbf{S}^{f(a)}\mathbf{0})$$

and we know that

$$A_E \vdash \theta(\mathbf{S}^a\mathbf{0}, \mathbf{S}^{f(a)}\mathbf{0}). \tag{5}$$

To obtain (3) we first note that we have as theorems of A_E,

$$v_2 < \mathbf{S}^{f(a)}\mathbf{0} \leftrightarrow v_2 = \mathbf{S}^0\mathbf{0} \vee \cdots \vee v_2 = \mathbf{S}^{f(a)-1}\mathbf{0} \tag{6}$$

and

$$\neg\, \theta(\mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}) \quad \text{for } b = 0, \ldots, f(a) - 1. \tag{7}$$

The formulas (6) and (7) imply the formula

$$v_2 < \mathbf{S}^{f(a)}\mathbf{0} \rightarrow \neg\, \theta(\mathbf{S}^a\mathbf{0},\, v_2). \tag{8}$$

Since $\theta(\mathbf{S}^a\mathbf{0},\, v_2) \in \Gamma$, we have (3).

This shows (2) to be a theorem of A_E; (5) and (8) show (1) to be a theorem of A_E as well. ⊣

We next want to show that certain basic functions are representable (in Cn A_E) and that the class of representable functions has certain closure properties. Henceforth in this section, when we say that a function or relation is representable, we will mean that it is representable in the theory Cn A_E. But the phrase "in Cn A_E" will usually be omitted.

In simple cases, an m-place function might be represented by an equation

$$\mathbf{v}_{m+1} = t.$$

In fact, any such equation, when the variables in t are among $\mathbf{v}_1, \ldots, \mathbf{v}_m$, defines in $\mathfrak{N}$ an m-place function f. (The value of f at $\langle a_1, \ldots, a_m\rangle$ is the number assigned in $\mathfrak{N}$ to t when $\mathbf{v}_i$ is assigned a_i, $1 \leq i \leq m$.) Furthermore, we know that any equation is numeralwise determined by A_E, so the equation represents f as a relation. In fact, it even functionally represents f, for the sentence

$$\forall\, \mathbf{v}_{m+1}[\mathbf{v}_{m+1} = t(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}) \leftrightarrow \mathbf{v}_{m+1} = \mathbf{S}^{f(a_1,\ldots,a_m)}\mathbf{0}]$$

is logically equivalent to

$$t(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}) = \mathbf{S}^{f(a_1,\ldots,a_m)}\mathbf{0},$$

which is a quantifier-free sentence true in $\mathfrak{N}$. (Here $t(u_1, \ldots, u_m)$ is the term obtained by replacing $\mathbf{v}_1$ by u_1, then $\mathbf{v}_2$ by u_2, etc.) For example:

1. The successor function is represented (functionally) by the equation

$$\mathbf{v}_2 = \mathbf{S}\mathbf{v}_1.$$

2. Any constant function is representable. The m-place function that constantly assumes the value b is represented by the equation

$$\mathbf{v}_{m+1} = \mathbf{S}^b\mathbf{0}.$$

3. The projection function (where $1 \leq i \leq m$)

$$I_i^m(a_1, \ldots, a_m) = a_i$$

is represented by the equation

$$\mathbf{v}_{m+1} = \mathbf{v}_i.$$

4. Addition, multiplication, and exponentiation are represented by the equations

$$\begin{aligned}\mathbf{v}_3 &= \mathbf{v}_1 + \mathbf{v}_2,\\ \mathbf{v}_3 &= \mathbf{v}_1 \cdot \mathbf{v}_2,\\ \mathbf{v}_3 &= \mathbf{v}_1\, \mathbf{E}\, \mathbf{v}_2,\end{aligned}$$

respectively.

The reader should not be misled by these simple examples; not every representable function is representable by an equation.

We next want to show that the family of representable functions is closed under composition. To simplify the notation, we will consider a one-place function f on $\mathbb{N}$, where

$$f(a) = g(h_1(a), h_2(a)).$$

Suppose that g is functionally represented by ψ and h_i by θ_i. To represent f it would be reasonable to try either

$$\forall y_1 \forall y_2(\theta_1(\boldsymbol{v}_1, y_1) \to \theta_2(\boldsymbol{v}_1, y_2) \to \psi(y_1, y_2, \boldsymbol{v}_2))$$

or

$$\exists y_1 \exists y_2(\theta_1(\boldsymbol{v}_1, y_1) \wedge \theta_2(\boldsymbol{v}_1, y_2) \wedge \psi(y_1, y_2, \boldsymbol{v}_2)).$$

(Think of $\psi(y_1, y_2, \boldsymbol{v}_2)$ as saying "$g(y_1, y_2) = \boldsymbol{v}_2$" and think of $\theta_i(\boldsymbol{v}_1, y_1)$ as saying "$h_i(\boldsymbol{v}_1) = y_i$." Then the first formula translates, "For any y_1, y_2, if $h_1(\boldsymbol{v}_1) = y_1$ and $h_2(\boldsymbol{v}_1) = y_2$, then $g(y_1, y_2) = \boldsymbol{v}_2$." The second formula translates, "There exist y_1, y_2 such that $h_1(\boldsymbol{v}_1) = y_1$ and $h_2(\boldsymbol{v}_1) = y_2$ and $g(y_1, y_2) = \boldsymbol{v}_2$." Either one is a reasonable way of saying, "$g(h_1(\boldsymbol{v}_1), h_2(\boldsymbol{v}_1)) = \boldsymbol{v}_2$." There are two choices, because when something is unique, either quantifier can be used for it.)

Actually either formula would work; let φ be

$$\forall y_1 \forall y_2(\theta_1(\boldsymbol{v}_1, y_1) \to \theta_2(\boldsymbol{v}_1, y_2) \to \psi(y_1, y_2, \boldsymbol{v}_2)).$$

Consider any natural number a; we have at our disposal

$$\forall \boldsymbol{v}_2[\psi(\mathbf{S}^{h_1(a)}\mathbf{0}, \mathbf{S}^{h_2(a)}\mathbf{0}, \boldsymbol{v}_2) \leftrightarrow \boldsymbol{v}_2 = \mathbf{S}^{f(a)}\mathbf{0}]. \tag{1}$$

$$\forall y_1[\theta_1(\mathbf{S}^{a}\mathbf{0}, y_1) \leftrightarrow y_1 = \mathbf{S}^{h_1(a)}\mathbf{0}]. \tag{2}$$

$$\forall y_2[\theta_2(\mathbf{S}^{a}\mathbf{0}, y_2) \leftrightarrow y_2 = \mathbf{S}^{h_2(a)}\mathbf{0}]. \tag{3}$$

And we want

$$\forall \boldsymbol{v}_2(\varphi(\mathbf{S}^{a}\mathbf{0}, \boldsymbol{v}_2) \leftrightarrow \boldsymbol{v}_2 = \mathbf{S}^{f(a)}\mathbf{0}), \tag{4}$$

i.e.,

$$\begin{aligned}\forall\ \boldsymbol{v}_2(\forall y_1 \forall y_2[\theta_1(\mathbf{S}^{a}\mathbf{0}, y_1) \to \theta_2(\mathbf{S}^{a}\mathbf{0}, y_2) \to \psi(y_1, y_2, \boldsymbol{v}_2)]\\ \leftrightarrow \boldsymbol{v}_2 = \mathbf{S}^{f(a)}\mathbf{0}).\end{aligned} \tag{4}$$

But (1), (2), and (3) imply (4), as the reader is asked to verify in Exercise 4.

More generally we have

Theorem 33L Let g be an n-place function, let $h_1, \ldots, h_n$ be m-place functions, and let f be defined by

$$f(a_1, \ldots, a_m) = g(h_1(a_1, \ldots, a_m), \ldots, h_n(a_1, \ldots, a_m)).$$

From formulas functionally representing g and $h_1, \ldots, h_n$ we can find a formula that functionally represents f.

In the above proof we have $m = 1$ and $n = 2$. But the general case is proved in exactly the same way.

In order to obtain a function such as

$$f(a, b) = g(h(a), b),$$

we note that

$$f(a, b) = g\big(h\big(I_1^2(a, b)\big), I_2^2(a, b)\big).$$

The above theorem then can be applied (twice) to show that f is representable (provided that g and h are).

To facilitate discussion of functions with an arbitrary number of variables, we will use vector notation. For example, the equation in the above theorem can be written

$$f(\vec{a}) = g(h_1(\vec{a}), \ldots, h_n(\vec{a})).$$

Another important closure property of the functions representable in Cn A_E is closure under the "least-zero" operator.

Theorem 33M Assume that the $(m + 1)$-place function g is representable and that for every $a_1, \ldots, a_m$ there is a b such that

$$g(a_1, \ldots, a_m, b) = 0.$$

Then we can find a formula that represents the m-place function f, where

$$f(a_1, \ldots, a_m) = \text{the least } b \text{ such that } g(a_1, \ldots, a_m, b) = 0.$$

(In vector notation we can rewrite this last equation:

$$f(\vec{a}) = \text{the least } b \text{ such that } g(\vec{a}, b) = 0.$$

The traditional notation for the least-zero operator is

$$f(\vec{a}) = \mu b[g(\vec{a}, b) = 0]$$

and the operator is often called "the μ-operator.")

Proof. To simplify the notation we take $m = 1$; thus

$$f(a) = b \quad \text{iff} \quad g(a, b) = 0 \quad \text{and for all} \quad c < b,\ g(a, c) \neq 0.$$

If ψ represents g, then we can obtain a formula representing f (as a relation) simply by formalizing the right side of this equivalence:

$$\psi(\mathbf{v}_1, \mathbf{v}_2, \mathbf{0}) \wedge \forall\, y(y < \mathbf{v}_2 \to \neg\, \psi(\mathbf{v}_1, y, \mathbf{0})).$$

This formula defines (the graph of) f and is numeralwise determined by A_E. ⊣

A Catalog

We now construct a repertoire of representable (in $\mathrm{Cn}\, A_E$) functions and relations, including in particular functions for encoding and decoding sequences.

0. As a consequence of Theorem 33I, any relation that has (in $\mathfrak{N}$) a quantifier-free definition is representable. And the class of representable relations is closed under unions, intersections, and complements. And if R is representable, then so are

$$\{\langle a_1, \ldots, a_m, b\rangle \mid \text{for all } c < b,\ \langle a_1, \ldots, a_m, c\rangle \in R\}$$

and

$$\{\langle a_1, \ldots, a_m, b\rangle \mid \text{for some } c < b,\ \langle a_1, \ldots, a_m, c\rangle \in R\}.$$

For example, any finite relation has a quantifier-free definition, as does the ordering relation.

1. A relation R is representable iff its characteristic function K_R is. (K_R is the function for which $K_R(\vec{a}) = 1$ when $\vec{a} \in R$, and $K_R(\vec{a}) = 0$ otherwise.)

Proof. (⇐) Say that R is a unary relation (a subset of $\mathbb{N}$) and that K_R is represented by $\psi(\boldsymbol{v}_1, \boldsymbol{v}_2)$. We claim that $\psi(\boldsymbol{v}_1, \mathbf{S0})$ represents R. For it defines R and is numeralwise determined by A_E.

(⇒) Say that $\varphi(\boldsymbol{v}_1)$ represents R. Then

$$(\varphi(\boldsymbol{v}_1) \wedge \boldsymbol{v}_2 = \mathbf{S0}) \vee (\neg\, \varphi(\boldsymbol{v}_1) \wedge \boldsymbol{v}_2 = \mathbf{0})$$

represents (the graph of) K_R, for the same reason as in the last paragraph. (Actually this formula even functionally represents K_R, as the reader can verify.) ⊣

2. If R is a representable binary relation and f, g are representable functions, then

$$\{\vec{a} \mid \langle f(\vec{a}), g(\vec{a})\rangle \in R\}$$

is representable. Similarly for an m-ary relation R and functions $f_1, \ldots, f_m$.

Proof. Its characteristic function at $\vec{a}$ has the value $K_R(f(\vec{a}), g(\vec{a}))$. Thus it is obtained from representable functions by composition. ⊣

For example, suppose that R is a representable ternary relation. Then

$$\{\langle x, y\rangle \mid \langle y, x, x\rangle \in R\}$$

is representable, being

$$\big\{\langle x, y\rangle \big| \big\langle I_2^2(x, y), I_1^2(x, y), I_1^2(x, y)\big\rangle \in R\big\}.$$

In this way we can rearrange and repeat variables in describing a representable relation.

3. If R is a representable binary relation, then so is

$$P = \{\langle a, b\rangle \mid \text{for some } c \leq b,\, \langle a, c\rangle \in R\}.$$

Proof. We have from catalog item 0 that if

$$Q = \{\langle a, b\rangle \mid \text{for some } c < b,\, \langle a, c\rangle \in R\},$$

then Q is representable. And

$$\begin{aligned}\langle a, b\rangle \in P &\Leftrightarrow \langle a, S(b)\rangle \in Q\\ &\Leftrightarrow \big\langle I_1^2(a, b), S\big(I_2^2(a, b)\big)\big\rangle \in Q.\end{aligned}$$

Hence by catalog item 2, P is representable. $\dashv$

More generally, if R is a representable $(m + 1)$-ary relation, then

$$\{\langle a_1, \ldots, a_m, b\rangle \mid \text{for some } c \leq b,\, \langle a_1, \ldots, a_m, c\rangle \in R\}$$

is also representable. In vector notation this relation becomes

$$\{\langle \vec{a}, b\rangle \mid \text{for some } c \leq b,\, \langle \vec{a}, c\rangle \in R\}.$$

Similarly

$$\{\langle \vec{a}, b\rangle \mid \text{for all } c \leq b,\, \langle \vec{a}, c\rangle \in R\}$$

is representable.

4. The divisibility relation

$$\{\langle a, b\rangle \mid a \text{ divides } b \text{ in } \mathbb{N}\}$$

is representable.

Proof. We have a divides b iff for some $q \leq b$, $a \cdot q = b$. We know that $\{\langle a, b, q\rangle \mid a \cdot q = b\}$ is representable, as it has a quantifier-free definition. Upon applying the above items, we get the divisibility relation. (In yet further detail, from catalog item 3 we get the representability of

$$R = \{\langle a, b, c\rangle \mid \text{for some } q \leq c,\ a \cdot q = b\}$$

and a divides b iff $\langle a, b, b\rangle \in R$.) $\dashv$

5. The set of primes is representable.
6. The set of pairs of adjacent primes is representable.

Proof. $\langle a, b\rangle$ is a pair of adjacent primes iff a is prime and b is prime and $a < b$ and there does not exist any $c < b$ such that

$a < c$ and c is prime. The right side of this equivalence is easily formalized by a numeralwise determined formula. ⊣

Note (for future use in Section 3.8) that we have not yet used the fact that exponentiation is representable.

Observe that as this catalog progresses, we are in effect building up a "language" $\mathcal{L}$ such that anything (any relation, any function) that is $\mathcal{L}$-definable (in $\mathfrak{N}$) will be certain to be representable in our theory. Thus, Theorem 33I tells us that (a) atomic formulas are allowed in $\mathcal{L}$, (b) all sentential connectives are permitted, and (c) bounded quantifiers can be used. (Unbounded quantifiers are not in general allowed.) Then our catalog gradually adds particular predicate symbols and function symbols; catalog item 6 adds a two-place predicate symbol for "adjacent primality"; and catalog item 7 will add a function symbol for the prime-listing function. Theorem 33L justifies using these function symbols inside expressions of $\mathcal{L}$.

7. The function whose value at a is p_a, the $(a + 1)$st prime, is representable. (Thus $p_0 = 2$, $p_1 = 3$, $p_2 = 5$, $p_3 = 7$, $p_4 = 11$, and so forth.)

Proof. $p_a = b$ iff b is prime and there exists some $c \leq b^{a^2}$, such that (i)–(iii) hold:

(i) 2 does not divide c.

(ii) For any $q < b$ and any $r \leq b$, if $\langle q, r \rangle$ is a pair of adjacent primes, then for all $j < c$,

$$q^j \text{ divides } c \iff r^{j+1} \text{ divides } c.$$

(iii) b^a divides c and b^{a+1} does not.

This equivalence is not obvious, but at least the relation defined by the right-hand side is representable. To verify the equivalence, first note that if $p_a = b$, then we can take

$$c = 2^0 \cdot 3^1 \cdot 5^2 \cdot \ldots \cdot p_a^a.$$

It is easy to check that this value of c meets all the conditions. Conversely, suppose c is a number meeting conditions (i)–(iii). We claim that c must be

$$2^0 \cdot 3^1 \cdot \ldots \cdot b^a \cdot \text{powers of larger primes.}$$

Certainly the exponent of 2 in c is 0, by (i). We can use (ii) to work our way across to the prime b. But by (iii) the exponent of b is a, so b must be the $(a + 1)$st prime, p_a. ⊣

This function will be very useful in encoding finite sequences of numbers into single numbers. Let

$$\begin{aligned}\langle a_0, \ldots, a_m\rangle &= p_0^{a_0+1} \cdot \cdots \cdot p_m^{a_m+1}\\ &= \prod_{i\le m} p_i^{a_i+1}.\end{aligned}$$

This holds also for $m = -1$; we define $\langle\ \rangle = 1$. For example,

$$\langle 2, 1\rangle = 2^3 \cdot 3^2 = 72.$$

The idea is that 72 safely encodes the pair $\langle 2, 1\rangle$.

There are other ways to encode pairs of numbers and finite sequences of numbers. In Section 3.8, we will make use of a pairing function

$$J(a, b) = \frac{1}{2}[(a + b)^2 + 3a + b]$$

that has the advantage of growing at a polynomial rate, unlike the growth rate of $2^{a+1}3^{b+1}$. Here is a very different way to encode, for example, the numbers 24, 117, 11 (in that order). First we convert to numerals in base 9: 26, 140, 12. Secondly, we concatenate these numerals, separated by 9's: 269140912. The triple is encoded by the number thereby designated (in base 10), that is, 269,140,912. This method may seem tricky, but it produces a result that is *much* smaller than $2^{25}3^{118}5^{12}$, which requires 73 digits in base 10.

8. For each m, the function whose value at $a_0, \ldots, a_m$ is $\langle a_0, \ldots, a_m\rangle$ is representable.

9. There is a representable function (whose value at $\langle a, b\rangle$ is written $(a)_b$) such that for $b \le m$,

$$(\langle a_0, \ldots, a_m\rangle)_b = a_b.$$

(This is our "decoding" function. For example, $(72)_0 = 2$ and $(72)_1 = 1$.)

Proof. We define $(a)_b$ to be the least n such that either $a = 0$ or p_b^{n+2} does not divide a. (There always *is* such an n.) Observe that $(0)_b = 0$, and for $a \ne 0$, $(a)_b$ is one less than the exponent of p_b in the prime factorization of a (but not less than 0). Hence for $b \le m$,

$$(\langle a_0, \ldots, a_m\rangle)_b = a_b.$$

To prove representability we use the least-zero operator. Let

$$R = \{\langle a, b, n\rangle \mid a = 0 \text{ or } p_b^{n+2} \text{ does not divide } a\}.$$

Then $(a)_b = \mu n[K_{\overline{R}}(a, b, n) = 0]$, where $\overline{R}$ is the complement of R. ⊣

Since the method used in the above proof will be useful elsewhere as well, we here state it separately:

Theorem 33N Assume that R is a representable relation such that for every $\vec{a}$ there is some n such that $\langle \vec{a}, n\rangle \in R$. Then the function f defined by

$$f(\vec{a}) = \text{the least } n \text{ such that } \langle \vec{a}, n\rangle \in R$$

is representable.

Proof. $f(\vec{a}) = \mu n[K_{\overline{R}}(\vec{a}, n) = 0]$. ⊣

We will later use the notation

$$f(\vec{a}) = \mu n[\langle \vec{a}, n\rangle \in R].$$

10. Say that b is a *sequence number* iff for some $m \geq -1$ and some $a_0, \ldots, a_m$,

$$b = \langle\!\langle a_0, \ldots, a_m\rangle\!\rangle.$$

(When $m = -1$ we get $\langle\!\langle\ \rangle\!\rangle = 1$.) Then the set of sequence numbers is representable.

Proof. Exercise 5. ⊣

11. There is a representable function lh such that

$$\text{lh}\langle\!\langle a_0, \ldots, a_m\rangle\!\rangle = m + 1.$$

(Here "lh" stands for "length." For example, lh 72 = 2.)

Proof. We define lh a to be the least n such that either $a = 0$ or p_n does not divide a. This works. ⊣

12. There is a representable function (whose value at $\langle a, b\rangle$ is called the *restriction* of a to b, written $a \upharpoonright b$) such that for any $b \leq m + 1$,

$$\langle\!\langle a_0, \ldots, a_m\rangle\!\rangle \upharpoonright b = \langle\!\langle a_0, \ldots, a_{b-1}\rangle\!\rangle.$$

Proof. Let $a \upharpoonright b$ be the least n such that either $a = 0$ or both $n \neq 0$ and for any $j < b$, any $k < a$

$$p_j^k \text{ divides } a \Rightarrow p_j^k \text{ divides } n.$$

This works. ⊣

13. (Primitive recursion) With a $(k + 1)$-place function f we associate another function $\overline{f}$ such that $\overline{f}(a, b_1, \ldots, b_k)$ encodes the values of $f(j, b_1, \ldots, b_k)$ for all $j < a$. Specifically, let

$$\overline{f}(a, \vec{b}) = \langle\!\langle f(0, \vec{b}), \ldots, f(a - 1, \vec{b})\rangle\!\rangle.$$

For example, $\overline{f}(0, \vec{b}) = \langle\!\langle\ \rangle\!\rangle = 1$, encoding the first zero values of f. $\overline{f}(1, \vec{b}) = \langle\!\langle f(0, \vec{b})\rangle\!\rangle$. In any case, $\overline{f}(a, \vec{b})$ is a sequence number of length a, encoding the first a values of f.

Now suppose we are given a $(k+2)$-place function g. There exists a unique function f satisfying

$$f(a, \vec{b}) = g(\overline{f}(a, \vec{b}), a, \vec{b}).$$

For example,

$$\begin{aligned} f(0, \vec{b}) &= g(\langle\ \rangle, 0, \vec{b}), \\ f(1, \vec{b}) &= g(\langle f(0, \vec{b})\rangle, 1, \vec{b}). \end{aligned}$$

(The existence and uniqueness of this f should be intuitively clear. For a proof, we can apply the recursion theorem of Section 1.4, obtaining first $\overline{f}$ and then extracting f.)

Theorem 33P Let g be a $(k+2)$-place function and let f be the unique $(k+1)$-place function such that for all a and (k-tuples) $\vec{b}$,

$$f(a, \vec{b}) = g(\overline{f}(a, \vec{b}), a, \vec{b}).$$

If g is representable, then so is f.

Proof. First we claim that $\overline{f}$ is representable.

This follows from the fact that

$\overline{f}(a, \vec{b}) =$ the least s such that s is a sequence number of length a and for i less than a, $(s)_i = g(s \restriction i, i, \vec{b})$.

It then follows that f is representable, since

$$f(a, \vec{b}) = g(\overline{f}(a, \vec{b}), a, \vec{b})$$

and the functions on the right are representable. ⊣

Actually the phrase "primitive recursion" is more commonly applied to a simpler version of this, given in Exercise 8.

14. For a representable function F, the function whose value at $a, \vec{b}$ is

$$\prod_{i<a} F(i, \vec{b})$$

is also representable. Similarly with Σ in place of Π. (For $a = 0$, we use the standard conventions: The empty product — the product of no numbers — is 1, and the empty sum is 0.)

Proof. Call this function G; then

$$\begin{aligned} G(0, \vec{b}) &= 1, \\ G(a+1, \vec{b}) &= F(a, \vec{b}) \cdot G(a, \vec{b}). \end{aligned}$$

Apply Exercise 8. ⊣

15. Define the *concatenation* of a and b, $a * b$, by

$$a * b = a \cdot \prod_{i<\mathrm{lh} b} p_{i+\mathrm{lh}\, a}^{(b)_i+1}.$$

This is a representable function of a and b, and

$$\langle a_1, \dots, a_m\rangle * \langle b_1, \dots, b_n\rangle = \langle a_1, \dots, a_m, b_1, \dots, b_n\rangle.$$

The concatenation operation has the further property of being associative on sequence numbers.

16. We will also want a "capital asterisk" operation. Let

$$*_{i<a} f(i) = f(0) * f(1) * \cdots * f(a-1).$$

For a representable function F, the function whose value at a, $\vec{b}$ is $*_{i<a} F(i, \vec{b})$ is representable.

Proof. $*_{i<0} F(i, \vec{b}) = \langle\,\rangle = 1$ and

$$*_{i<a+1} F(i, \vec{b}) = *_{i<a} F(i, \vec{b}) * F(a, \vec{b}).$$

So this is just like catalog item 14. ⊣

Exercises

1. Show that in the structure $(\mathbb{N}; \cdot, E)$ we can define the addition relation $\{\langle m, n, m+n\rangle \mid m, n \text{ in } \mathbb{N}\}$. Conclude that in this structure $\{0\}$, the ordering relation $<$, and the successor relation $\{\langle n, S(n)\rangle \mid n \in \mathbb{N}\}$ are definable. (*Remark*: This result can be strengthened by replacing the structure $(\mathbb{N}; \cdot, E)$ by simply $(\mathbb{N}; E)$. The multiplication relation is definable here, by exploiting one of the laws of exponents: $(d^a)^b = d^{ab}$.)

2. Prove Theorem 33C, stating that true (in $\mathfrak{N}$) quantifier-free sentences are theorems of A_E. (See the outline given there.)

3. A theory T (in a language with $\mathbf{0}$ and $\mathbf{S}$) is called *ω-complete* iff for any formula φ and variable x, if $\varphi^x_{\mathbf{S}^n\mathbf{0}}$ belongs to T for every natural number n, then $\forall x\varphi$ belongs to T. Show that if T is a consistent ω-complete theory in the language of $\mathfrak{N}$ and if $A_E \subseteq T$, then $T = \operatorname{Th}\mathfrak{N}$.

4. Show that in the proof preceding Theorem 33L, formula (4) is logically implied by the set consisting of formulas (1), (2), and (3).

5. Show that the set of sequence numbers is representable (catalog item 10).

6. Is 3 a sequence number? What is lh 3? Find $(1*3)*6$ and $1*(3*6)$.

7. Establish the following facts:
   (a) $a + 1 < p_a$.
   (b) $(b)_k \le b$; equality holds iff $b = 0$.
   (c) $\operatorname{lh} a \le a$; equality holds iff $a = 0$.
   (d) $a \upharpoonright i \le a$.
   (e) $\operatorname{lh}(a \upharpoonright i)$ is the smaller of i and $\operatorname{lh} a$.

8. Let g and h be representable functions, and assume that

$$f(0, b) = g(b),$$
$$f(a + 1, b) = h(f(a, b), a, b).$$

Show that f is representable.

9. Show that there is a representable function f such that for every $n, a_0, \ldots, a_n$,

$$f(\langle a_0, \ldots, a_n \rangle) = a_n.$$

(For example, $f(72) = 1$ and $f(750) = 2$.)

10. Assume that R is a representable relation and that g and h are representable functions. Show that f is representable, where

$$f(\vec{a}) = \begin{cases} g(\vec{a}) & \text{if } \vec{a} \in R, \\ h(\vec{a}) & \text{if } \vec{a} \notin R. \end{cases}$$

11. (Monotone recursion) Assume that R is a representable binary relation on $\mathbb{N}$. Let C be the smallest subset of $\mathbb{N}$ (i.e., the intersection of all subsets) such that for all $n, a_0, \ldots, a_{n-1}, b$,

$$\langle \langle a_0, \ldots, a_{n-1} \rangle, b \rangle \in R \;\;\&\;\; a_i \in C \text{ (for all } i < n) \;\;\Rightarrow\;\; b \in C.$$

Further assume that (1) for all $n, a_0, \ldots, a_{n-1}, b$,

$$\langle \langle a_0, \ldots, a_{n-1} \rangle, b \rangle \in R \;\;\Rightarrow\;\; a_i < b \;\; \text{(for all } i < n),$$

and (2) there is a representable function f such that for all n, $a_0, \ldots, a_{n-1}, b$,

$$\langle \langle a_0, \ldots, a_{n-1} \rangle, b \rangle \in R \;\;\Rightarrow\;\; n < f(b)$$

Show that C is representable. (C is, in a sense, generated by R. $C \neq \varnothing$ in general because if $\langle \langle\, \rangle, b \rangle \in R$, then $b \in C$.)

SECTION 3.4

Arithmetization of Syntax

In this section we intend to develop two themes:

1. Certain assertions about wffs can be converted into assertions about natural numbers (by assigning numbers to expressions).

2. These (English) assertions about natural numbers can in many cases be translated into the formal language. And the theory Cn A_E is strong enough to prove many of the translations so obtained.

This will give us the ability to construct formulas that, by expressing facts about numbers, indirectly express facts about formulas (even about themselves!). Such an ability will be exploited in Section 3.5 to obtain results of undefinability and undecidability.

Gödel Numbers

We first want to assign numbers to expressions of the formal language. Recall that the symbols of our language are those listed in Table IX.

TABLE IX

Parameters	Logical symbols
0. $\forall$	1. (
2. $\mathbf{0}$	3.)
4. $\mathbf{S}$	5. $\neg$
6. $<$	7. $\rightarrow$
8. $+$	9. $=$
10. $\cdot$	11. $\boldsymbol{v}_1$
12. $\mathbf{E}$	13. $\boldsymbol{v}_2$, etc.

There is a function h assigning to each symbol the integer listed to its left. Thus $h(\forall) = 0$, $h(\mathbf{0}) = 2$, and $h(\boldsymbol{v}_i) = 9 + 2i$. In order to make our subsequent work more widely applicable, we will assume only that we have some language with $\mathbf{0}$ and $\mathbf{S}$ which is *recursively numbered.* By this we mean that we have a one-to-one function h from the parameters of that language into the even numbers such that the two relations

$$\{\langle k, m\rangle \mid k \text{ is the value of } h \text{ at some } m\text{-place predicate parameter}\}$$

and

$$\{\langle k, m\rangle \mid k \text{ is the value of } h \text{ at some } m\text{-place function symbol}\}$$

are both representable in $\operatorname{Cn} A_E$. Of course in the case of the language of $\mathfrak{N}$ these sets are even finite. The first set is $\{\langle 6, 2\rangle\}$ and the second is

$$\{\langle 2, 0\rangle, \langle 4, 1\rangle, \langle 8, 2\rangle, \langle 10, 2\rangle, \langle 12, 2\rangle\}.$$

We define h on the logical symbols as before; thus $h(s)$ is an odd number for each logical symbol s.

For an expression $\varepsilon = s_0 \cdots s_n$ of the language we define its Gödel number, $\sharp(\varepsilon)$, by

$$\sharp(s_0 \cdots s_n) = \langle h(s_0), \ldots, h(s_n)\rangle.$$

For example, using our original function h for the language of $\mathfrak{N}$, we obtain

$$\begin{aligned}
&\sharp(\exists\, \boldsymbol{v}_3\ \boldsymbol{v}_3 = \mathbf{0})\\
&\qquad = \sharp((\neg \forall\, \boldsymbol{v}_3 (\neg = \boldsymbol{v}_3 \mathbf{0})))\\
&\qquad = \langle 1, 5, 0, 15, 1, 5, 9, 15, 2, 3, 3\rangle\\
&\qquad = 2^2 \cdot 3^6 \cdot 5^1 \cdot 7^{16} \cdot 11^2 \cdot 13^6 \cdot 17^{10} \cdot 19^{16} \cdot 23^3 \cdot 29^4 \cdot 31^4.
\end{aligned}$$

This is a large number, being of the order of 1.3×10^{75}. To a set Φ of expressions we assign the set

$$\sharp\Phi = \{\sharp(\varepsilon) \mid \varepsilon \in \Phi\}$$

of Gödel numbers.

To a sequence $\langle\alpha_0, \dots, \alpha_n\rangle$ of expressions (such as a deduction), we assign the number

$$\mathcal{G}(\langle\alpha_0, \dots, \alpha_n\rangle) = \langle\sharp\alpha_0, \dots, \sharp\alpha_n\rangle.$$

We now proceed to show that various relations and functions having to do with Gödel numbers are representable in $\text{Cn}\, A_E$ (and hence are recursive). As in the preceding section, whenever we say that a relation or function is representable (without specifying a theory) we mean that it is representable in the theory $\text{Cn}\, A_E$.

We will make use of certain abbreviations in the language we use (i.e., English, although it is coming to differ more and more from what one ordinarily thinks of as English). For "there is a number a such that" we write "$\exists a$." In the same spirit, "$\exists a, b < c$" means "there are numbers a and b both of which are less than c such that." Similarly, we may employ "$\forall$." We would not have dared to employ such abbreviations in Chapter 2, for fear of creating confusion between the formal language and the meta-language (English). But by now we trust the reader to avoid such erroneous ways.

1. The set of Gödel numbers of variables is representable.

PROOF. It is $\{a \mid (\exists b < a) a = \langle 11 + 2b\rangle\}$. It follows from results of the preceding section that this is a representable set. ⊣

2. The set of Gödel numbers of terms is representable.

PROOF. The set of terms was defined inductively. And terms were built up from constituents with smaller Gödel numbers. We will treat this case in some detail, since it is typical of the argument used for inductively defined relations.

Let f be the characteristic function of the set of Gödel numbers of terms. From the definition of "term" we obtain

$$f(a) = \begin{cases} 1 & \text{if } a \text{ is the Gödel number of a variable,} \\ 1 & \text{if } (\exists i < \square, \exists k < a)\ [i \text{ is a sequence number} \\ & \quad \& \ (\forall j < \text{lh}\, i) f((i)_j) = 1 \ \& \ k \text{ is the value of} \\ & \quad h \text{ at some } (\text{lh}\, i)\text{-place function symbol } \& \\ & \quad a = \langle k\rangle * \ast_{j<\text{lh}\, i} (i)_j], \\ 0 & \text{otherwise.} \end{cases}$$

But what upper bound for i can we use in place of that "□" symbol? Before we can argue that f is representable, we will need an upper bound on i that depends in some representable way on a.

The claim is that we can take $i < a^{a\,\mathrm{lh}\,a}$. To see this, suppose that $a = \sharp s t_1 \cdots t_n$ (where s is an n-place function symbol and $t_1, \ldots, t_n$ are terms). Then we want to take $i = \langle \sharp t_1, \ldots, \sharp t_n \rangle$. How big could this be, in terms of a? We have the bounds:

$$\begin{aligned} i &= 2^{\sharp t_1 + 1} \cdots p_{n-1}^{\sharp t_n + 1} \\ &\leq 2^a \cdots p_{n-1}^a \\ &< 2^a \cdots p_{\mathrm{lh}\,a-1}^a \quad \text{because } n = \mathrm{lh}\,i < \mathrm{lh}\,a \\ &\leq a^a \cdots a^a \quad (\mathrm{lh}\,a \text{ times}) \quad \text{because } a = 2^{(a)_0+1} \cdots p_{\mathrm{lh}\,a-1}^{(a)_{\mathrm{lh}\,a-1}+1} \geq p_{\mathrm{lh}\,a-1} \\ &= (a^a)^{\mathrm{lh}\,a} = a^{a\,\mathrm{lh}\,a} \end{aligned}$$

So in the above equation for f, we replace $\square$ by $a^{a\,\mathrm{lh}\,a}$.

Although the right side of this equation refers to f, it refers only to $f((i)_j)$, where $(i)_j < a$. This feature permits us to apply primitive recursion. $f(a) = g(\overline{f}(a), a)$, where

$$g(s, a) = \begin{cases} 1 & \text{if } a \text{ is the Gödel number of a variable,} \\ 1 & \text{if } (\exists i < a^{a\,\mathrm{lh}\,a}, \exists k < a)\ [i \text{ is a sequence number} \\ & \quad \& \ (\forall j < \mathrm{lh}\,i)(s)_{(i)_j} = 1 \ \& \ k \text{ is the value of} \\ & \quad h \text{ at some } (\mathrm{lh}\,i)\text{-place function symbol } \& \\ & \quad a = \langle k \rangle * *_{j < \mathrm{lh}\,i}(i)_j], \\ 0 & \text{otherwise.} \end{cases}$$

For if in this equation we set s equal to $\overline{f}(a)$, then $(s)_{(i)_j} = f((i)_j)$ for $(i)_j < a$. Hence by Theorem 33P, f is representable provided that g is.

It remains to show that g is representable. But this is straightforward, by using results of the preceding section. Briefly, the graph of g is the union of three relations, corresponding to the three clauses in the above equation. Each of the three is obtained from equality and other representable relations by bounded quantification and the substitution of representable functions. ⊣

3. The set of Gödel numbers of atomic formulas is representable.

Proof. a is the Gödel number of an atomic formula iff $(\exists i < a^{a\,\mathrm{lh}\,a}, \exists k < a)$ [i is a sequence number & $(\forall j < \mathrm{lh}\,i)(i)_j$ is the Gödel number of a term & k is the value of h at some $(\mathrm{lh}\,i)$-place predicate symbol & $a = \langle k \rangle * *_{j < \mathrm{lh}\,i}(i)_j$]. ⊣

4. The set of Gödel numbers of wffs is representable.

Proof. The wffs were inductively defined. Let f be the characteristic function of the set, then

$$f(a) = \begin{cases} 1 & \text{if } a \text{ is the Gödel number of an atomic formula,} \\ 1 & \text{if } (\exists i < a)[a = \langle h((), h(\neg)\rangle * i * \langle h())\rangle \\ & \quad \& \ f(i) = 1], \\ 1 & \text{if } (\exists i, j < a)[a = \langle h(()\rangle * i * \langle h(\rightarrow)\rangle * j * \langle h())\rangle \\ & \quad \& \ f(i) = f(j) = 1], \\ 1 & \text{if } (\exists i, j < a)[a = \langle h(\forall)\rangle * i * j \ \& \ i \text{ is the Gödel} \\ & \quad \text{number of a variable and } f(j) = 1], \\ 0 & \text{otherwise.} \end{cases}$$

By the same argument used for the set of Gödel numbers of terms, we have the representability of f. ⊣

5. There is a representable function Sb such that for a term or formula α, variable x, and term t,

$$\mathrm{Sb}(\sharp\alpha, \sharp x, \sharp t) = \sharp\alpha^x_t.$$

Proof. We will need to define $\mathrm{Sb}(a, b, c)$ making use of values $\mathrm{Sb}(i, b, c)$ where $i < a$. As in the case of catalog item 2 (the characteristic function of the set of terms), it will then be possible to show that both $\overline{\mathrm{Sb}}$ and Sb are representable.

The function Sb is described by the following six clauses (i)–(vi):

(i) If a is the Gödel number of a variable and $a = b$ then

$$\mathrm{Sb}(a, b, c) = c.$$

(ii) If $(\exists i < a^{a\,\mathrm{lh}\,a}, \exists k < a)[i$ is a sequence number & $(\forall j < \mathrm{lh}\, i)(i)_j$ is the Gödel number of a term & k is the value of h at some $(\mathrm{lh}\, i)$-place function or predicate symbol & $a = \langle k\rangle * *_{j<\mathrm{lh}\, i}(i)_j]$ then

$$\mathrm{Sb}(a, b, c) = \langle k\rangle * *_{j<\mathrm{lh}\, i}\mathrm{Sb}((i)_j, b, c)$$

for that i and k.

(iii) If $(\exists i < a)[i$ is the Gödel number of a wff & $a = \langle h((), h(\neg)\rangle * i * \langle h())\rangle]$ then

$$\mathrm{Sb}(a, b, c) = \langle h((), h(\neg)\rangle * \mathrm{Sb}(i, b, c) * \langle h())\rangle$$

for that i.

(iv) If $(\exists i, j < a)[i$ and j are Gödel numbers of wffs & $a = \langle h(()\rangle * i * \langle h(\rightarrow)\rangle * j * \langle h())\rangle]$ then

$$\mathrm{Sb}(a, b, c) = \langle h(()\rangle * \mathrm{Sb}(i, b, c) * \langle h(\rightarrow)\rangle * \mathrm{Sb}(j, b, c) * \langle h())\rangle$$

for that i and j.

(v) If $(\exists i, j < a)[i$ is the Gödel number of a variable & $i \neq b$ & j is the Gödel number of a wff & $a = \langle h(\forall)\rangle * i * j]$ then

$$\mathrm{Sb}(a, b, c) = \langle h(\forall)\rangle * i * \mathrm{Sb}(j, b, c)$$

for that i and j.

(vi) If none of the above conditions on a and b are met (where we ignore the displayed equation for $\mathrm{Sb}(a, b, c)$) then

$$\mathrm{Sb}(a, b, c) = a.$$

Then the function Sb is obtained by primitive recursion

$$\mathrm{Sb}(a, b, c) = G(\overline{\mathrm{Sb}}(a, b, c), a, b, c)$$

where G is a 4-place function. The graph of G is the union of six 5-ary relations

$$G = R_1 \cup R_2 \cup R_3 \cup R_4 \cup R_5 \cup R_6$$

corresponding to the six clauses above.

The first of the six is

$$R_1 = \{\langle s, a, b, c, d\rangle \mid a \text{ is the Gödel number of a variable \& } a = b \;\&\; d = c\}.$$

The second one is

$$R_2 = \{\langle s, a, b, c, d\rangle \mid (\exists i < a^{a\,\mathrm{lh}\,a}, \exists k < a)[i \text{ is a sequence number } \& (\forall j < \mathrm{lh}\, i)(i)_j \text{ is the Gödel number of a term \& } k \text{ is the value of } h \text{ at some } (\mathrm{lh}\, i)\text{-place function or predicate symbol \& } a = \langle k\rangle * *_{j<\mathrm{lh}\, i}(i)_j \;\&\; d = \langle k\rangle * *_{j<\mathrm{lh}\, i}\,(s)_{(i)_j}]\}$$

and the others are similar translations of the corresponding clauses in the description of Sb.

It is necessary to note that G is indeed a function; it is single-valued. This is because no two clauses could apply to one number a. And if, for example, clause (ii) applies to a, then we know from Section 2.3 that the numbers i and k are uniquely determined.

Finally, we apply the usual methods to verify that R_1–R_6 are representable, so G is representable, so $\overline{\mathrm{Sb}}$ is representable, so Sb is representable. (Substitution is a complicated operation!) ⊣

6. The function whose value at n is $\sharp(\mathbf{S}^n\mathbf{0})$ is representable.

PROOF. Call this function f; then

$$f(0) = \langle h(\mathbf{0})\rangle,$$
$$f(n+1) = \langle h(\mathbf{S})\rangle * f(n).$$

Apply Exercise 8 of the preceding section. ⊣

7. There is a representable relation Fr such that for a term or formula α and a variable x,

$$\langle \sharp\alpha, \sharp x\rangle \in \mathrm{Fr} \Leftrightarrow x \text{ occurs free in } \alpha.$$

Proof. $\langle a, b\rangle \in \mathrm{Fr} \Leftrightarrow \mathrm{Sb}(a, b, \sharp\mathbf{0}) \neq a$. ⊣

8. The set of Gödel numbers of sentences is representable.

Proof. a is the Gödel number of a sentence iff a is the Gödel number of a formula and for any $b < a$, if b is the Gödel number of a variable then $\langle a, b\rangle \notin \mathrm{Fr}$. ⊣

9. There is a representable relation Sbl such that for a formula α, variable x, and term t, $\langle \sharp a, \sharp x, \sharp t\rangle \in \mathrm{Sbl}$ iff t is substitutable for x in α.

Proof. Exercise 1. ⊣

10. The relation Gen, where $\langle a, b\rangle \in \mathrm{Gen}$ iff a is the Gödel number of a formula and b is the Gödel number of a generalization of that formula, is representable.

Proof. $\langle a, b\rangle \in \mathrm{Gen}$ iff $a = b$ or $(\exists i, j < b)[i$ is the Gödel number of a variable and $\langle a, j\rangle \in \mathrm{Gen}$ and $b = \big\langle (h(\forall)\big\rangle * i * j]$. Apply the usual argument to the characteristic function of Gen. ⊣

11. The set of Gödel numbers of tautologies is representable.

The set of tautologies is informally decidable since we can use the method of truth tables. To obtain representability, we recast truth tables in terms of Gödel numbers. There are several preliminary steps:

11.1 The relation R, such that $\langle a, b\rangle \in R$ iff a is the Gödel number of a formula α and b is the Gödel number of a prime constituent of α, is representable.

Proof. $\langle a, b\rangle \in R \Leftrightarrow a$ is the Gödel number of a formula and one of the following:

(i) $a = b \;\&\; (a)_0 \neq h(()$.
(ii) $(\exists i < a)[a = \big\langle h((), h(\neg)\big\rangle * i * \big\langle (h())\big\rangle$ and $\langle i, b\rangle \in R]$.
(iii) The analogue to (ii) but with $\longrightarrow$.

Apply the usual argument to the characteristic function of R. ⊣

11.2 There is a representable function P such that for a formula α, $P(\sharp\alpha) = \big\langle \sharp\beta_1, \ldots, \sharp\beta_n\big\rangle$, the list of Gödel numbers of prime constituents of α, in numerical order.

Proof. First define a function g for locating the next prime constituent in $\natural a$ after $\natural y$ (where $\natural a$ is the formula α for which $a = \sharp\alpha$).

$$g(a, y) = \text{the least } n \text{ such that either } n = a + 1 \text{ or both}$$
$$y < n \text{ and } \langle a, n\rangle \in R.$$

Next define a function h such that $h(a, n)$ gives the $(n+1)$st prime constituent of $\natural a$ (if there are that many):

$$h(a, 0) = g(a, 0) \qquad h(a, n + 1) = g(a, h(a, n)).$$

Finally, let $P(a) = *_{i<k}\langle h(a, i)\rangle$ where k is the least number for which $h(a, k) > a$. ⊣

11.3 Say that the integer v *encodes a truth assignment* for α iff v is a sequence number and $\operatorname{lh} v = \operatorname{lh} P(\sharp\alpha)$ and $(\forall i < \operatorname{lh} v)(\exists e < 2)(v)_i = \langle (P(\sharp\alpha))_i, e\rangle$. This is a representable condition on v and $\sharp\alpha$.

For example, if $P(\sharp\alpha) = \langle \sharp\beta_0, \ldots, \sharp\beta_n\rangle$, then

$$v = \langle\langle \sharp\beta_0, e_0\rangle, \ldots, \langle \sharp\beta_n, e_n\rangle\rangle,$$

where each e_i is 0 or 1. We will later need an upper bound for v in terms of $\sharp\alpha$. The largest v is obtained when each e_i is 1. Also $\sharp\beta_i \leq \sharp\alpha$, so that

$$\begin{aligned} v &\leq \langle\langle \sharp\alpha, 1\rangle, \ldots, \langle \sharp\alpha, 1\rangle\rangle \\ &= *_{i<\operatorname{lh} P(\sharp\alpha)}\langle\langle \sharp\alpha, 1\rangle\rangle. \end{aligned}$$

11.4 There is a representable relation Tr such that for a formula α and a v which encodes a truth assignment for α (or more), $\langle \sharp\alpha, v\rangle \in \mathrm{Tr}$ iff that truth assignment satisfies α.

Proof. Exercise 2. ⊣

Finally, α is a tautology iff α is a formula and for every v encoding a truth assignment for α, $\langle \sharp\alpha, v\rangle \in \mathrm{Tr}$. The (English) quantifier on v can be bounded by a representable function of $\sharp\alpha$, as explained in 11.3.

12. The set of Gödel numbers of formulas of the form $\forall x\, \varphi \rightarrow \varphi^x_t$, where t is a term substitutable for the variable x in φ, is representable.

Proof. α is of this form iff $(\exists \text{ wff } \varphi < \alpha)(\exists \text{ variable } x < \alpha)(\exists \text{ term } t < \alpha)[t$ is substitutable for x in φ and $\alpha = \forall x\, \varphi \rightarrow \varphi^x_t]$. Here "$\varphi < \alpha$" means that $\sharp\varphi < \sharp\alpha$. This is easily rewritten in terms of Gödel numbers: a belongs to the set iff $(\exists f < a)(\exists x < a)(\exists t < a)$ $[f$ is the Gödel number of a formula & x is the Gödel number of a variable & t is the Gödel number of term and $\langle f, x, t\rangle \in \mathrm{Sbl}$ &

$$a = \langle h(\,(\,), h(\forall)\rangle * x * f * \langle h(\rightarrow)\rangle * \mathrm{Sb}(f, x, t) * \langle h())\rangle].$$

13. The set of Gödel numbers of formulas of the form $\forall x(\alpha \rightarrow \beta) \rightarrow \forall x\, \alpha \rightarrow \forall x\, \beta$ is representable.

Proof. γ is of this form iff $(\exists \text{ variable } x < \gamma)(\exists \text{ formulas } \alpha, \beta < \gamma)$ $[\gamma = \forall x(\alpha \rightarrow \beta) \rightarrow \forall x\, \alpha \rightarrow \forall x\, \beta]$. This is easily rewritten in terms of Gödel numbers, as in 12. ⊣

14. The set of Gödel numbers of formulas of the form $\alpha \rightarrow \forall x\, \alpha$, where x does not occur free in α, is representable.

Proof. Similar to 13. ⊣

15. The set of Gödel numbers of formulas of the form $x = x$ is representable.

Proof. Similar to 13. ⊣

16. The set of Gödel numbers of formulas of the form $x = y \to \alpha \to \alpha'$, where α is atomic and α' is obtained from α by replacing x at zero or more places by y, is representable.

Proof. This is similar to 13, except for the relation of "partial substitution." Let $\langle a, b, x, y\rangle \in \mathrm{Psb}$ iff x and y are Gödel numbers of variables, a is the Gödel number of an atomic formula, b is a sequence number of length $\mathrm{lh}\, a$, and for all $j < \mathrm{lh}\, a$, either $(a)_j = (b)_j$ or $(a)_j = x$ and $(b)_j = y$. This relation is representable. ⊣

17. The set of Gödel numbers of logical axioms is representable.

Proof. α is a logical axiom iff $\exists\beta \leq \alpha$ such that α is a generalization of β and β is in one of the sets in items 11–16. ⊣

18. For a finite set A of formulas,

$$\{\mathcal{G}(D) \mid D \text{ is a deduction from } A\}$$

is representable. In fact it is enough here for $\sharp A$ to be representable.

Proof. A number d belongs to this set iff d is a sequence number of positive length and for every i less than $\mathrm{lh}\, d$, either

1. $(d)_i \in \sharp A$,
2. $(d)_i$ is the Gödel number of a logical axiom, or
3. $(\exists j, k < i)[(d)_j = \langle h(()\rangle * (d)_k * \langle h(\to)\rangle * (d)_i * \langle h())\rangle]$.

This is representable whenever $\sharp A$ is, as is certainly the case for finite A. ⊣

19. Any recursive relation is representable in $\mathrm{Cn}\, A_E$.

Proof. Recall that the relation R is recursive iff there is *some* finite consistent set A of sentences such that some formula ρ represents R in $\mathrm{Cn}\, A$. (There is no loss of generality in assuming that the language has only finitely many parameters: those in the finite set A, those in ρ, and $\mathbf{0}$, $\mathbf{S}$, and $\forall$.) In the case of a unary relation R, we have that $a \in R$ iff the least D which is a deduction from A of either $\rho(\mathbf{S}^a\mathbf{0})$ or $\neg\rho(\mathbf{S}^a\mathbf{0})$ is, in fact, a deduction of the former.

More formally, $a \in R$ iff the last component of $f(a)$ is $\sharp\rho(\mathbf{S}^a\mathbf{0})$, where

$f(a) =$ the least d such that d is in the set of item 18 and the last component of d is either $\sharp\rho(\mathbf{S}^a\mathbf{0})$ or $\sharp\neg\rho(\mathbf{S}^a\mathbf{0})$.

For this (fixed) ρ, there always is such a d. ⊣

Since the converse to item 19 is immediate, we have

Theorem 34A A relation is recursive iff it is representable in the theory $\mathrm{Cn}\, A_E$.

Henceforth we will usually use the word “recursive” in preference to “representable.”

Corollary 34B Any recursive relation is definable in $\mathfrak{N}$.

20. Now suppose we have a set A of sentences such that $\sharp A$ is recursive. Then $\sharp\,\mathrm{Cn}\,A$ need *not* be recursive (as we will show in the next section). But we do have a way of defining $\mathrm{Cn}\,A$ from A:

$a \in \sharp\,\mathrm{Cn}\,A$ iff $\exists d[d$ is the number of a deduction from A
and the last component of d is a and a is the
Gödel number of a sentence$]$

The part in square brackets is recursive, by the proof to item 18. But we cannot in general put any bound on the number d. The best we can say is that $\sharp\,\mathrm{Cn}\,A$ is the domain of a recursive relation (or, as we will say later, is *recursively enumerable*).

Item 20 will play a key role our subsequent work. In particular, it will later be restated as Theorem 35I.

21. If $\sharp A$ is recursive and $\mathrm{Cn}\,A$ is a complete theory, then $\sharp\,\mathrm{Cn}\,A$ is recursive.

In other words, a complete recursively axiomatizable theory is recursive. This is the analogue to Corollary 25G, which asserts that a complete axiomatizable theory is decidable.

The proof is essentially unchanged. Let (in the consistent case)

$g(s) =$ the least d such that s is not the Gödel number of a sentence,
or d is in the set of item 18 and the last component of d is
either s or is $\langle h((), h(\neg)\rangle * s * \langle h())\rangle$.

Thus $g(\sharp\sigma)$ is $\mathcal{G}$ of the least deduction of σ or $(\neg\,\sigma)$ from A. And $s \in \sharp\,\mathrm{Cn}\,A$ iff $s > 0$ and the last component of $g(s)$ is s. ⊣

At this point we might reconsider the plausibility of Church’s thesis. Suppose that the relation R is decidable. Then there is a finite list of explicit instructions (a program) for the decision procedure. The procedure itself will presumably consist of certain atomic steps, which are then performed repeatedly. (The reader familiar with computer programming will know that a short program can still require much time for its execution, but some commands will be utilized over and over.) Any one atomic step is presumably very simple.

By devices akin to Gödel numbering, we can mirror the decision procedure in the integers. The characteristic function of R can then be put in the form

$K_R(\vec{a}) = U$[the least s such that

(i) $(s)_0$ encodes the input $\vec{a}$;

(ii) for all positive $i < \operatorname{lh} s$, $(s)_i$ is obtained from $(s)_{i-1}$ by performance of the applicable atomic step;

(iii) the last component of s describes a terminal situation, at which the computation is completed],

where U (the upshot function) is some simple function that extracts from the last component of s the answer (affirmative or negative). The recursiveness of R is now reduced to the recursiveness of U and of the relations indicated in (i), (ii), and (iii). In special cases, such as decision procedures provided by the register machines of Section 3.6, the recursiveness of these components is easily verified. It seems most improbable that any decision procedure will ever be regarded as effective and yet will have components that are nonrecursive. For example, in (ii), it seems that it ought to be possible to make each atomic step extremely simple, and in particular to make each one recursive.

Exercises

1. Supply a proof for item 9 of this section.

2. Supply a proof for item 11.4 of this section.

3. Use Exercise 11 of Section 3.3 to give a new proof that the set of Gödel numbers of terms is representable (item 2).

4. Let T be a consistent recursively axiomatizable theory (in a recursively numbered language with $\mathbf{0}$ and $\mathbf{S}$). Show that any relation representable in T must be recursive.

SECTION 3.5

Incompleteness and Undecidability

In this section we reap the rewards of our work in Sections 3.3 and 3.4. We have assigned Gödel numbers to expressions, and we have shown that certain intuitively decidable relations on $\mathbb{N}$ (related to syntactical notions about expressions) are representable in $\operatorname{Cn} A_E$.

Throughout this section we assume that the language in question is the language of $\mathfrak{N}$. (This affects the meaning of "Cn" and "theory.")

Fixed-Point Lemma Given any formula β in which only $\boldsymbol{v}_1$ occurs free, we can find a sentence σ such that

$$A_E \vdash [\sigma \leftrightarrow \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0})].$$

We can think of σ as indirectly saying, "β is true of me." Actually, of course, σ doesn't say anything; it's just a string of symbols. And even when translated into English according to the intended structure $\mathfrak{N}$, it then talks of numbers and their successors and products and so forth. It is only by virtue of our having associated numbers with expressions that we can think of σ as referring to a formula, in this case to σ itself.

PROOF. Let $\theta(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)$ functionally represent in $\operatorname{Cn} A_E$ a function whose value at $\langle \sharp\alpha, n\rangle$ is $\sharp(\alpha(\mathbf{S}^n\mathbf{0}))$. (See items 5 and 6 in Section 3.4.) First consider the formula

$$\forall\, \boldsymbol{v}_3[\theta(\boldsymbol{v}_1, \boldsymbol{v}_1, \boldsymbol{v}_3) \to \beta(\boldsymbol{v}_3)]. \tag{1}$$

(We may suppose $\boldsymbol{v}_3$ is substitutable for $\boldsymbol{v}_1$ in β. The above formula has only $\boldsymbol{v}_1$ free. It defines in $\mathfrak{N}$ a set to which $\sharp\alpha$ belongs iff $\sharp(\alpha(\mathbf{S}^{\sharp\alpha}\mathbf{0}))$ is in the set defined by β.) Let q be the Gödel number of (1). Let σ be

$$\forall\, \boldsymbol{v}_3[\theta(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \boldsymbol{v}_3) \to \beta(\boldsymbol{v}_3)].$$

Thus σ is obtained from (1) by replacing $\boldsymbol{v}_1$, by $\mathbf{S}^q\mathbf{0}$. Notice that σ does assert (under $\mathfrak{N}$) that $\sharp\sigma$ is in the set defined by β. But we must check that

$$\sigma \leftrightarrow \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0}) \tag{2}$$

is a consequence of A_E. Because θ functionally represents a function whose value at $\langle q, q\rangle$ is $\sharp\sigma$, we have

$$A_E \vdash \forall\, \boldsymbol{v}_3[\theta(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \boldsymbol{v}_3) \leftrightarrow \boldsymbol{v}_3 = \mathbf{S}^{\sharp\sigma}\mathbf{0}]. \tag{3}$$

We can obtain (2) as follows:

($\to$) It is clear (by looking at σ) that

$$\sigma \vdash \theta(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \mathbf{S}^{\sharp\sigma}) \to \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0}).$$

And, by (3),

$$A_E \vdash \theta(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \mathbf{S}^{\sharp\sigma}\mathbf{0}).$$

Hence

$$A_E; \sigma \vdash \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0}),$$

which gives half of (2).

($\leftarrow$) The sentence in (3) implies

$$\beta(\mathbf{S}^{\sharp\sigma}\mathbf{0}) \to [\forall\, \boldsymbol{v}_3(\theta(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \boldsymbol{v}_3) \to \beta(\boldsymbol{v}_3))].$$

But the part in square brackets is just σ. ⊣

(Sometimes the notation $\ulcorner\sigma\urcorner$ is used for $\mathbf{S}^{\sharp\sigma}\mathbf{0}$. In this notation, the fixed-point lemma states that $A_E \vdash (\sigma \leftrightarrow \beta(\ulcorner\sigma\urcorner))$.)

Our first application of this lemma does not concern the subtheory $\operatorname{Cn} A_E$, and requires only the weaker fact that

$$\models_{\mathfrak{N}} [\sigma \leftrightarrow \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0})].$$

Tarski Undefinability Theorem (1933) The set $\sharp \operatorname{Th} \mathfrak{N}$ is not definable in $\mathfrak{N}$.

Proof. Consider any formula β (which you suspect *might* define $\sharp \operatorname{Th} \mathfrak{N}$). By the fixed-point lemma (applied to $\neg \beta$) we have a sentence σ such that

$$\models_{\mathfrak{N}} [\sigma \leftrightarrow \neg \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0})].$$

(If β did define $\sharp \operatorname{Th} \mathfrak{N}$, then σ would indirectly say "I am false.") Then

$$\models_{\mathfrak{N}} \sigma \iff \not\models_{\mathfrak{N}} \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0}),$$

so either σ is true but (its Gödel number is) not in the set β defines, or it is false and in that set. Either way σ shows that β cannot define $\sharp \operatorname{Th} \mathfrak{N}$. ⊣

The above theorem gives at once the undecidability of the theory of $\mathfrak{N}$:

Corollary 35A $\sharp \operatorname{Th} \mathfrak{N}$ is not recursive.

Proof. Any recursive set is (by Corollary 34B) definable in $\mathfrak{N}$. ⊣

Gödel Incompleteness Theorem (1931) If $A \subseteq \operatorname{Th} \mathfrak{N}$ and $\sharp A$ is recursive, then $\operatorname{Cn} A$ is not a complete theory.

Thus there is no complete recursive axiomatization of $\operatorname{Th} \mathfrak{N}$.

Proof. Since $A \subseteq \operatorname{Th} \mathfrak{N}$, we have $\operatorname{Cn} A \subseteq \operatorname{Th} \mathfrak{N}$. If $\operatorname{Cn} A$ is a complete theory, then equality holds. But if $\operatorname{Cn} A$ is a complete theory, then $\sharp \operatorname{Cn} A$ is recursive (item 21 of the preceding section). And by the above corollary, $\sharp \operatorname{Th} \mathfrak{N}$ is not recursive. ⊣

In particular, $\operatorname{Cn} A_E$ is not a complete theory and so is not equal to $\operatorname{Th} \mathfrak{N}$. And the incompleteness would not be eliminated by the addition of any recursive set of true axioms. (By a recursive set of sentences we mean of course a set Σ for which $\sharp\Sigma$ is recursive.)

We can extract more information from the proof of Gödel's theorem. Suppose we have a particular recursive $A \subseteq \operatorname{Th} \mathfrak{N}$ in mind. Then by item 20 in Section 3.4 we can find a formula β that defines $\sharp \operatorname{Cn} A$ in $\mathfrak{N}$. The sentence σ produced by the proof of Tarski's theorem is (as we noted there) a true sentence *not* in $\operatorname{Cn} A$. This sentence asserts that $\sharp\sigma$ does not belong to the set defined by β, i.e., it indirectly says, "I am not a theorem of A." Thus $A \nvdash \sigma$, and of course $A \nvdash \neg\sigma$ as well. This way of viewing the proof is closer to Gödel's original proof, which did not involve a detour through Tarski's theorem. For that matter, Gödel's statement of the theorem did not involve $\operatorname{Th} \mathfrak{N}$; we have taken some liberties in the labeling of theorems.

Next we need a lemma which says (roughly) that one can add one new axiom (and hence finitely many new axioms) to a recursive theory without losing the property of recursiveness.

Lemma 35B If $\sharp \operatorname{Cn} \Sigma$ is recursive, then $\sharp \operatorname{Cn}(\Sigma; \tau)$ is recursive.

Proof. $\alpha \in \operatorname{Cn}(\Sigma; \tau) \Leftrightarrow (\tau \rightarrow \alpha) \in \operatorname{Cn} \Sigma$. Thus

$$\begin{aligned} a \in \sharp \operatorname{Cn}(\Sigma; \tau) \iff & a \text{ is the Gödel number of a sentence} \\ & \text{and } \langle h(()\rangle * \sharp\tau * \langle h(\rightarrow)\rangle * a * \langle h())\rangle \\ & \text{is in } \sharp \operatorname{Cn} \Sigma. \end{aligned}$$

This is recursive by the results of the preceding sections. ⊣

Theorem 35C (Strong Undecidability of Cn A_E) Let T be a theory such that $T \cup A_E$ is consistent. Then $\sharp T$ is not recursive.

(Notice that because throughout this section the language in question is the language of $\mathfrak{N}$, the word "theory" here means "theory in the language of $\mathfrak{N}$.")

Proof. Let T' be the theory $\operatorname{Cn}(T \cup A_E)$. If $\sharp T$ is recursive, then since A_E is finite we can conclude by the above lemma that $\sharp T'$ is also recursive.

Suppose, then, that $\sharp T'$ is recursive and so is represented in $\operatorname{Cn} A_E$ by some formula β. From the fixed-point lemma we get a sentence σ such that

$$A_E \vdash [\sigma \leftrightarrow \neg \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0})]. \qquad (*)$$

(Indirectly σ asserts, "I am not in T'.")

$$\begin{aligned} \sigma \notin T' &\Rightarrow \sharp\sigma \notin \sharp T' \\ &\Rightarrow A_E \vdash \neg \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0}) \\ &\Rightarrow A_E \vdash \sigma \qquad \text{by } (*) \\ &\Rightarrow \sigma \in T'. \end{aligned}$$

So we get $\sigma \in T'$. But this, too, is untenable:

$$\begin{aligned} \sigma \in T' &\Rightarrow \sharp\sigma \in \sharp T' \\ &\Rightarrow A_E \vdash \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0}) \\ &\Rightarrow A_E \vdash \neg \sigma \qquad \text{by } (*) \\ &\Rightarrow (\neg \sigma) \in T', \end{aligned}$$

which contradicts the consistency of T'. ⊣

Corollary 35D Assume that $\sharp\Sigma$ is recursive and $\Sigma \cup A_E$ is consistent. Then $\operatorname{Cn} \Sigma$ is not a complete theory.

Proof. A complete recursively axiomatizable theory is recursive (item 21 of Section 3.4). But $\sharp \operatorname{Cn} \Sigma$ is not recursive, by the above theorem. ⊣

This corollary is Gödel's incompleteness theorem again, but with truth in $\mathfrak{N}$ replaced by consistency with A_E.

Church's Theorem (1936) The set of Gödel numbers of valid sentences (in the language of $\mathfrak{N}$) is not recursive.

Proof. In the strong undecidability of $\operatorname{Cn} A_E$, take T to be the smallest theory in the language, the set of valid sentences. $\dashv$

The set of Gödel numbers of valid wffs is not recursive either, lest the set of valid sentences be recursive.

This proof applies to the language of $\mathfrak{N}$. For a language with more parameters, the set of valid sentences is still nonrecursive (lest its intersection with the language of $\mathfrak{N}$ be recursive). Actually it is enough for the language to contain at least one two-place predicate symbol. (See Corollary 37G.) On the other hand, *some* lower bound on the language is needed. If the language has $\forall$ as its only parameter (the language of equality), then the set of valid formulas is decidable. (See Exercise 6.) More generally, it is known that if the only parameters are $\forall$ and one-place predicate symbols, then the set of valid formulas is decidable.

Recursive Enumerability

A relation on the natural numbers is said to be *recursively enumerable* iff it is of the form

$$\{\vec{a} \mid \exists b \, \langle \vec{a}, b\rangle \in Q\}$$

with Q recursive. Recursively enumerable relations play an important role in logic. They constitute the formal counterpart to the effectively enumerable relations (as will be explained presently).

(The standard abbreviation for "recursively enumerable" is "r.e." When the term "computable" is used instead of "recursive," then one speaks of *computably enumerable* — abbreviated c.e. — relations.)

Recursively enumerable relations are — like the recursive relations — definable in $\mathfrak{N}$. If $\varphi(\mathbf{v}_1, \mathbf{v}_2)$ defines in $\mathfrak{N}$ a binary relation Q, then $\exists\, \mathbf{v}_2\, \varphi(\mathbf{v}_1, \mathbf{v}_2)$ defines $\{a \mid \exists b \, \langle a, b\rangle \in Q\}$.

Theorem 35E The following conditions on an m-ary relation R are equivalent:

1. R is recursively enumerable.
2. R is the domain of some recursive relation Q.
3. For some recursive $(m+1)$-ary relation Q,

$$R = \{\langle a_1, \ldots, a_m\rangle \mid \exists b \, \langle a_1, \ldots, a_m, b\rangle \in Q\}.$$

4. For some recursive $(m+n)$-ary relation Q,

$$R = \{\langle a_1, \ldots, a_m\rangle \mid \exists b_1, \ldots, b_n \, \langle a_1, \ldots, a_m, b_1, \ldots, b_n\rangle \in Q\}.$$

Proof. By definition 1 and 3 are equivalent. Also 2 and 3 are equivalent by our definition (in Chapter 0) of domain and $(m+1)$-tuple.

Clearly 3 implies 4. So we have only to show that 4 implies 3. This is because

$$\begin{aligned}&\exists b_1, \ldots, b_n \, \langle a_1, \ldots, a_m, b_1, \ldots, b_n\rangle \in Q\\ &\text{iff } \exists c \, \langle a_1, \ldots, a_m, (c)_0, \ldots, (c)_{n-1}\rangle \in Q\end{aligned}$$

and

$$\{\langle a_1 \ldots, a_m, c\rangle \mid \langle a_1, \ldots, a_m, (c)_0, \ldots, (c)_{n-1}\rangle \in Q\}$$

is recursive whenever Q is recursive. (Here we have used our sequence decoding function to collapse a string of quantifiers into a single one.) ⊣

By part 4 of this theorem, R is recursively enumerable iff it is definable in $\mathfrak{N}$ by a formula $\exists x_1 \cdots \exists x_n\varphi$, where φ is numeralwise determined by A_E. In fact, we can require here that φ be quantifier-free; this result was proved in 1961 (with exponentiation) and in 1970 (without exponentiation). The proofs involve some number theory; we will omit them here.

Notice that any recursive relation is also recursively enumerable. For if R is recursive, then it is defined in $\mathfrak{N}$ by a formula $\exists x_1 \cdots \exists x_n\varphi$, where φ is numeralwise determined by A_E and $x_1, \ldots, x_n$ do not occur in φ.

Theorem 35F A relation is recursive iff both it and its complement are recursively enumerable.

This is the formal counterpart to the fact (cf. Theorem 17F) that a relation is decidable iff both it and its complement can be effectively enumerated.

Proof. If a relation is recursive, then so is its complement, whence both are recursively enumerable.

Conversely, suppose that both P and its complement are recursively enumerable; thus for any $\vec{a}$,

$$\begin{aligned}\vec{a} \in P &\Leftrightarrow \exists b \, \langle \vec{a}, b\rangle \in Q\\ \vec{a} \notin P &\Leftrightarrow \exists b \, \langle \vec{a}, b\rangle \in R\end{aligned}$$

for some recursive Q and R. Let

$f(\vec{a})$ = the least b such that either $\langle \vec{a}, b\rangle \in Q$ or $\langle \vec{a}, b\rangle \in R$.

Such a number b always exists, and f is recursive. Finally,

$$\vec{a} \in P \Leftrightarrow \langle \vec{a}, f(\vec{a})\rangle \in Q,$$

so P is recursive. ⊣

The recursively enumerable relations constitute the formal counterpart of the effectively enumerable relations. For we have the following informal result, paralleling a characterization of recursive enumerability given by Theorem 35E.

***Lemma 35G** A relation is effectively enumerable iff it is the domain of a decidable relation.

Proof. Assume that Q is effectively enumerated by some procedure. Then $\vec{a} \in Q$ iff $\exists n[\vec{a}$ appears in the enumeration in n steps]. The relation defined in square brackets is decidable and has domain Q.

Conversely, to enumerate $\{\langle a, b\rangle \mid \exists n\langle a, b, n\rangle \in R\}$ for decidable R, we check to see if $\langle (m)_0, (m)_1, (m)_2\rangle \in R$ for $m = 0, 1, 2, \ldots$. Whenever the answer is affirmative, we place $\langle (m)_0, (m)_1\rangle$ on the output list. $\dashv$

***Corollary 35H (Church's Thesis, Second Form)** A relation is effectively enumerable iff it is recursively enumerable.

Proof. By identifying the class of decidable relations with the class of recursive relations, we automatically identify the domains of decidable relations with domains of recursive relations. $\dashv$

The second form of Church's thesis is, in fact, equivalent to the first form. To prove the first form from the second, we use Theorems 35F and 17F.

We have already shown that a recursively axiomatizable theory is recursively enumerable, but using different words. We restate the result here, as it indicates the role recursive enumerability plays in logic.

Theorem 35I If A is a set of sentences such that $\sharp A$ is recursive, then $\sharp \operatorname{Cn} A$ is recursively enumerable.

Proof. Item 20 of Section 3.4. $\dashv$

In particular, $\sharp \operatorname{Cn} A_E$ is recursively enumerable, but (by Theorem 35C) it is not recursive. In the next section, we will see other examples of recursively enumerable sets that are not recursive.

This theorem is the precise counterpart of the informal fact that a theory with a decidable set of axioms is effectively enumerable (Corollaries 25F and 26I). It indicates the gap between what is *provable* in an axiomatic theory and what is *true* in the intended structure. With a recursive set of axioms, all we can possibly obtain is a recursively enumerable set of consequences. But by Tarski's theorem, Th $\mathfrak{N}$ is not even definable in $\mathfrak{N}$, much less recursively enumerable.

Even if we expand the language or add new axioms, the same phenomenon is present. As long as we can recursively distinguish deductions from nondeductions, the set of theorems can be only recursively enumerable. For example, the set of sentences of number theory provable in your favorite system of axiomatic set theory is recursively enumerable. Furthermore, this set includes A_E and is consistent (unless you have very strange favorites). It follows that this set theory is nonrecursive and incomplete. (This topic is discussed more carefully in Section 3.7.)

Weak Representability

Consider a recursively enumerable set Q, where

$$a \in Q \Leftrightarrow \exists b\, \langle a, b \rangle \in R$$

for recursive R. We know there is a formula ρ that represents R in $\mathrm{Cn}\, A_E$. Consequently, the formula $\exists \boldsymbol{v}_2 \rho$ defines Q in $\mathfrak{N}$. This formula cannot represent Q in $\mathrm{Cn}\, A_E$ unless Q is recursive. But it can come halfway.

$$\begin{array}{rlll} a \in Q & \Rightarrow & \langle a, b \rangle \in R & \text{for some } b \\ & \Rightarrow & A_E \vdash \rho(\mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}) & \text{for some } b \\ & \Rightarrow & A_E \vdash \exists\, \boldsymbol{v}_2 \rho(\mathbf{S}^a\mathbf{0}, \boldsymbol{v}_2) & \\ a \notin Q & \Rightarrow & \langle a, b \rangle \notin R & \text{for all } b \\ & \Rightarrow & A_E \vdash \neg\, \rho(\mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}) & \text{for all } b \\ & \Rightarrow & A_E \nvdash \exists\, \boldsymbol{v}_2 \rho(\mathbf{S}^a\mathbf{0}, \boldsymbol{v}_2) & \end{array}$$

The last step is justified by the fact that if $A_E \vdash \neg\, \varphi(\mathbf{S}^b\mathbf{0})$ for all b, then $A_E \nvdash \exists\, x\, \varphi(x)$. (The term *ω-consistency* is given to this property.) For it is impossible for $\exists\, x\, \varphi(x)$, $\neg\, \varphi(\mathbf{S}^0\mathbf{0})$, $\neg\, \varphi(\mathbf{S}^1\mathbf{0})$, ... all to be true in $\mathfrak{N}$.

Thus we have

$$a \in Q \;\Leftrightarrow\; A_E \vdash \exists\; \boldsymbol{v}_2 \rho(\mathbf{S}^a\mathbf{0}, \boldsymbol{v}_2).$$

It will be convenient to formulate a definition of this half of representability.

Definition. Let Q be an n-ary relation on $\mathbb{N}$, ψ a formula in which only $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ occur free. Then ψ *weakly represents* Q in a theory T iff for every $a_1, \ldots, a_n$ in $\mathfrak{N}$,

$$\langle a_1, \ldots, a_n \rangle \in Q \;\Leftrightarrow\; \psi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}) \in T.$$

Observe that if Q is representable in a consistent theory T, then Q is also weakly representable in T.

Theorem 35J A relation is weakly representable in $\mathrm{Cn}\, A_E$ iff it is recursively enumerable.

Proof. We just showed that a recursively enumerable unary relation Q is weakly representable in $\mathrm{Cn}\, A_E$; the same proof applies to n-ary Q with only notational changes. Conversely, let Q be weakly represented by ψ in $\mathrm{Cn}\, A_E$. Then

$$\begin{aligned}\langle a_1, \ldots, a_n\rangle \in Q \;&\Leftrightarrow\; \exists D\,[D \text{ is a deduction of } \psi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_n}\mathbf{0}) \\ &\qquad \text{from the axioms } A_E] \\ &\Leftrightarrow\; \exists d\, \langle d, f(a_1, \ldots, a_n)\rangle \in P\end{aligned}$$

for a certain recursive function f and recursive relation P. ⊣

Arithmetical Hierarchy

Define a relation on the natural numbers to be *arithmetical* iff it is definable in $\mathfrak{N}$. But some arithmetical relations are, in a sense, more definable than others. We can organize the arithmetical relations into a hierarchy according to how definable the relations are.

Let Σ_1 be the class of recursively enumerable relations; these relations are "one quantifier away" from recursiveness. Extending this idea, we define the class of Σ_k relations and the class of Π_k relations. For example, the first few classes consist of relations of the form shown in the second column:

$$\begin{array}{lll}\Sigma_1: & \{\vec{a} \mid \exists b\, \langle \vec{a}, b\rangle \in R\}, & R \text{ recursive.}\\ \Pi_1: & \{\vec{a} \mid \forall b\, \langle \vec{a}, b\rangle \in R\}, & R \text{ recursive.}\\ \Sigma_2: & \{\vec{a} \mid \exists c \forall b\, \langle \vec{a}, b, c\rangle \in R\}, & R \text{ recursive.}\\ \Pi_2: & \{\vec{a} \mid \forall c \exists b\, \langle \vec{a}, b, c\rangle \in R\}, & R \text{ recursive.}\end{array}$$

In general, a relation Q is in Π_k iff it is of the form

$$\{\vec{a} \mid \forall b_1 \exists b_2 \cdots \Box b_k\, \langle \vec{a}, \vec{b}\rangle \in R\}$$

for a recursive relation R. Here "□" is to be replaced by "∀" if k is odd and by "∃" if k is even. Similarly, Q is in Σ_k iff it has the form

$$\{\vec{a} \mid \exists b_1 \forall b_2 \cdots \Box b_k\, \langle \vec{a}, \vec{b}\rangle \in R\}$$

for recursive R, where now "□" is replaced by "∃" if k is odd and by "∀" if k is even.

The classes Σ_k and Π_k can also be defined by recursion on k. Σ_1 is the class of recursively enumerable relations. Next, a relation belongs to Π_k iff its complement is in Σ_k. And a relation is in Σ_{k+1} iff it is the domain of a relation in Π_k. (We can even start from $k = 0$, by letting Σ_0 be the class of recursive relations.)

Example. The set of Gödel numbers of formulas numeralwise determined by A_E is in Π_2.

Proof. a belongs to this set iff [a is the Gödel number of a formula α] and $\forall b \exists d[d$ is $\mathcal{G}$ of a deduction from A_E either of $\alpha(\mathbf{S}^{(b)_0}\mathbf{0}, \mathbf{S}^{(b)_1}\mathbf{0}, \ldots)$ or of the negation of this sentence]. By the technique of Section 3.4, we can show that the phrases in square brackets define recursive relations. By using the English counterpart to prenex form, we obtain the desired form,

$$\{a \mid \forall b \exists d\, \langle a, b, d\rangle \in R\},$$

with R recursive. ⊣

One more bit of notation: Let Δ_1 be the class of recursive relations. Then our earlier result (Theorem 35F) stating that a relation is recursive iff both it and its complement are recursively enumerable can now be stated by the equation

$$\Delta_1 = \Sigma_1 \cap \Pi_1.$$

Since this equation holds, we proceed to define Δ_n for $n > 1$ by the analogous equation,

$$\Delta_n = \Sigma_n \cap \Pi_n.$$

The following inclusions hold:

$$\begin{tikzcd}
 & \Sigma_1 \arrow[dr, phantom, sloped, "\subseteq"] & & \Sigma_2 \arrow[dr, phantom, sloped, "\subseteq"] & & \Sigma_3 & \\
\Delta_1 \arrow[ur, phantom, sloped, "\subseteq"] \arrow[dr, phantom, sloped, "\subseteq"] & & \Delta_2 \arrow[ur, phantom, sloped, "\subseteq"] \arrow[dr, phantom, sloped, "\subseteq"] & & \Delta_3 \arrow[ur, phantom, sloped, "\subseteq"] \arrow[dr, phantom, sloped, "\subseteq"] & & \cdots \\
 & \Pi_1 \arrow[ur, phantom, sloped, "\subseteq"] & & \Pi_2 \arrow[ur, phantom, sloped, "\subseteq"] & & \Pi_3 &
\end{tikzcd}$$

The case $\Delta_1 \subseteq \Sigma_1$ was mentioned previously (cf. Theorem 35F); its proof hinged on the possibility of "vacuous quantification." The proofs of the other cases are conceptually the same. If x does not occur in φ, then φ, $\forall x\, \varphi$, and $\exists x\, \varphi$ are all equivalent. For example, a relation in Σ_1 is defined by a formula $\exists y\, \varphi$, where φ is numeralwise determined by A_E. But the same relation is defined by $\exists y \forall x\, \varphi$ and $\forall x \exists y\, \varphi$ (where x does not occur in φ). Hence the relation is also in Σ_2 and Π_2.

It is also true that all the inclusions shown are proper inclusions, i.e., equality does not hold. But we will not prove this fact here. The inclusions are shown pictorially in Fig. 10.

The class of arithmetical relations equals $\bigcup_k \Sigma_k$ and also $\bigcup_k \Pi_k$. For example, any relation in Σ_2 is arithmetical, being defined in $\mathfrak{N}$ by a formula $\exists x \forall y\, \varphi$, where φ is numeralwise determined by A_E. Conversely, any arithmetical relation is defined in $\mathfrak{N}$ by some prenex formula. The quantifier-free part of this prenex formula defines a recursive relation (since quantifier-free formulas are numeralwise determined by A_E). Consequently, the defined relation falls somewhere in the hierarchy. The technique of "collapsing" $\exists\exists \cdots \exists$ quantifiers used in the proof of Theorem 35E (and its dual technique for $\forall\forall \cdots \forall$) can be used to good advantage here.

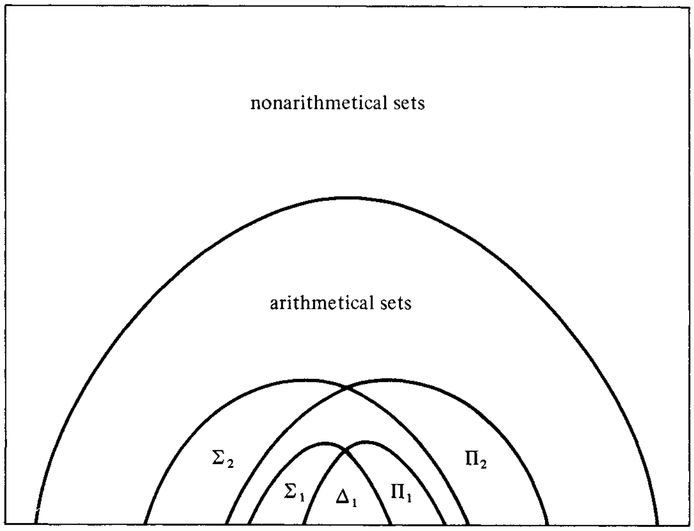


Figure 10. Picture of $\mathcal{P}\mathbb{N}$.

Thus we have the following result, which relates definability in $\mathfrak{N}$ to the hierarchy we have just built up from the recursive relations:

Theorem 35K A relation on the natural numbers is arithmetical (i.e., definable in $\mathfrak{N}$) iff it is in Σ_k for some k, and this property in turn is equivalent to being in Π_l for some l.

In particular, any recursively enumerable relation is arithmetical, as noted previously.

There are certain tricks which are useful in locating specific arithmetical relations in the hierarchy. For example, let A be the set of Gödel numbers of formulas α such that for some n,

$$A_E \vdash \alpha(\mathbf{S}^n\mathbf{0}) \quad \text{and} \quad (\forall i < n)\; A_E \vdash \neg\alpha(\mathbf{S}^i\mathbf{0}).$$

Then $a \in A$ iff [a is the Gödel of a wff α] and $\exists n \exists D[D$ is a deduction from A_E of $\alpha(\mathbf{S}^n\mathbf{0})]$ and $(\forall i < n)(\exists D_i)[D_i$ is a deduction from A_E of $\neg\alpha(\mathbf{S}^i\mathbf{0})]$. The parts in square brackets are recursive so we count the remaining quantifiers. The bounded quantifier "$\forall i < n$" need not be counted. For we have

$$(\forall i < n)(\exists d)\langle d, i\rangle \in P \Leftrightarrow (\exists d)(\forall i < n)\langle (d)_i, i\rangle \in P.$$

Use of this fact lets us push the bounded quantifier inward until it merges with the recursive part. Consequently, $A \in \Sigma_1$.

The following theorem generalizes Theorem 35I.

Theorem 35L Let A be a set of sentences such that $\sharp A$ is in Σ_k, where $k > 0$. Then $\sharp \operatorname{Cn} A$ is also in Σ_k.

Proof. Return to the proofs of items 18 and 20 of Section 3.4. We had there:

$a \in \sharp \operatorname{Cn} A \Leftrightarrow a$ is the Gödel number of a sentence and $\exists d[d$ is a sequence number and the last component of d is a and for every i less than $\operatorname{lh} d$, either (1) $(d)_i \in \sharp A$, (2) $(d)_i$ is the Gödel number of a logical axiom, or (3) for some j and l less than i, $(d)_j = \langle h(()\rangle * (d)_l * \langle h(\rightarrow)\rangle * (d)_i * \langle h())\rangle]$.

Since $\sharp A \in \Sigma_k$ in (1) we must replace "$(d)_i \in \sharp A$" by something of the form

$$\exists b_1 \forall b_2 \cdots \Box b_k \langle (d)_i, \vec{b}\rangle \in Q$$

for recursive Q. It remains to convert the result into an English prenex expression in Σ_k form. We suggest that the reader set $k = 2$ and write out this expression; the device used in the preceding example will help. ⊣

Exercises

1. Show that there is no recursive set R such that $\sharp \operatorname{Cn} A_E \subseteq R$ and $\sharp\{\sigma \mid (\neg \sigma) \in \operatorname{Cn} A_E\} \subseteq \overline{R}$, the complement of R. (This result can be stated: The theorems of A_E cannot be recursively separated from the refutable sentences.) *Suggestion*: Make a sentence σ saying "My Gödel number is *not* in R." Look to see where $\sharp\sigma$ is.

2. Let A be a recursive set of sentences in a recursively numbered language with $\mathbf{0}$ and $\mathbf{S}$. Assume that every recursive relation is representable in the theory $\operatorname{Cn} A$. Further assume that A is ω-consistent; i.e., there is no formula φ such that $A \vdash \exists x\, \varphi(x)$ and for all $a \in \mathbb{N}$, $A \vdash \neg \varphi(\mathbf{S}^a\mathbf{0})$. Construct a sentence σ indirectly asserting that it is not a theorem of A, and show that neither $A \vdash \sigma$ nor $A \vdash \neg \sigma$. *Suggestion*: See Section 3.0.

   Remark: This is a version of the incompleteness theorem that is closer to Gödel's original 1931 argument. Note that there is no requirement that the axioms A be *true* in $\mathfrak{N}$. Nor is it required that A include A_E; the fixed-point argument can still be applied.

3. Let T be a theory in a recursively numbered language (with $\mathbf{0}$ and $\mathbf{S}$). Assume that all recursive subsets of $\mathbb{N}$ are weakly representable in T. Show that $\sharp T$ is not recursive. *Suggestion*: Construct a binary relation P such that any weakly representable subset of $\mathbb{N}$ equals $\{b \mid \langle a, b\rangle \in P\}$ for some a, and such that P is recursive if $\sharp T$ is.

Consider the set $H = \{b \mid \langle b, b\rangle \notin P\}$. See Section 3.0. The argument given there for the "diagonal approach" in the special case $T = \text{Th}\,\mathfrak{N}$ can be adapted here.

Remark: This exercise gives a version of the result, "Any sufficiently strong theory is undecidable."

4. Show that there exist $2^{\aleph_0}$ nonisomorphic countable models of Th $\mathfrak{N}$. *Suggestion*: For each set A of primes, make a model having an element divisible by exactly the primes in A.

5. (Lindenbaum) Let T be a decidable consistent theory (in a reasonable language). Show that T can be extended to a complete decidable consistent theory T'. *Suggestion*: Examine in turn each sentence σ; add either σ or $\neg\sigma$ to T. But take care to maintain decidability.

6. Consider the language of equality, having $\forall$ as its only parameter. Let λ_n be the translation of "There are at least n things," cf. the proof of Theorem 26A. Call a formula *simple* iff it can be built up from atomic formulas and the λ_n's by use of connective symbols (but no quantifiers). Show how, given any formula in the language of equality, we can find a logically equivalent simple formula. *Suggestion*: View this as an elimination-of-quantifiers result (where the quantifiers in λ_n do not count). Use Theorem 31F.

7. (a) Assume that A and B are subsets of $\mathbb{N}$ belonging to Σ_k (or Π_k). Show that $A \cup B$ and $A \cap B$ also belong to Σ_k (or Π_k, respectively).

(b) Assume that A is in Σ_k (or Π_k) and that the functions $f_1, \ldots, f_m$ are recursive. Show that

$$\{\vec{a} : \langle f_1(\vec{a}), \ldots, f_m(\vec{a})\rangle \in A\}$$

is also in Σ_k (or Π_k, respectively). *Suggestion*: First do this for Σ_1. Then observe that the argument used can be generalized.

8. Let T be a theory in a recursively numbered language (with $\mathbf{0}$ and $\mathbf{S}$). Let n be fixed, $n \geq 0$. Assume that all subsets of $\mathbb{N}$ in Σ_n are weakly representable in T. Show that $\sharp T$ is not in Π_n. (Observe that Exercise 3 is a special case of this, wherein $n = 0$. The suggestions given there carry over to the present case.)

9. Show that

$$\{\sharp\sigma \mid A_E; \sigma \text{ is } \omega\text{-consistent}\}$$

(see Exercise 2) is a Π_3 set.

10. The theory Cn A_E has many complete extensions, of which Th $\mathfrak{N}$ is but one. How many? That is, what is the cardinality of the set of complete theories (in the language) that extend A_E?

SECTION 3.6

Recursive Functions

We have used recursive functions (i.e., functions that, when viewed as relations, are recursive) to obtain theorems of incompleteness and undecidability of theories. But the class of recursive functions is also an interesting class in its own right, and in this section we will indicate a few of its properties.

Recall that by Church's thesis, a function is recursive iff it is computable by an effective procedure (page 210). This fact is responsible for much of the interest in recursive functions. At the same time, this fact makes possible an intuitive understanding of recursiveness, which greatly facilitates the study of the subject. Suppose, for example, that you are suddenly asked whether or not the inverse of a recursive permutation of $\mathbb{N}$ is recursive. Before trying to prove this, you should first ask yourself the intuitive counterpart: Is the inverse of a computable permutation f also computable? You then — one hopes — perceive that the answer is affirmative. To compute $f^{-1}(3)$, you can compute $f(0), f(1), \ldots$ until for some k it is found that $f(k) = 3$. Then $f^{-1}(3) = k$. Having done this, you have gained two advantages. For one, you feel certain that the answer to the question regarding recursive permutations must also be affirmative. And secondly, you have a good outline of how to prove this; the proof is found by making rigorous the intuitive proof. This strategy for approaching problems involving recursiveness will be very useful in this section.

Before proceeding, it might be wise to summarize here some of the facts about recursive functions we have already established. We know that a function f is recursive iff it (as a relation) is representable in $\mathrm{Cn}\,A_E$, by Theorem 34A. Consequently, every recursive function is weakly representable in this theory.

In Section 3.3 a repertoire of recursive functions was developed. In addition, it was shown that the class of recursive functions is closed under certain operations, such as composition (Theorem 33L) and the "least-zero" operator (Theorem 33M).

We also know of a few functions that are not recursive. There are uncountably many (to be exact, $2^{\aleph_0}$) functions from $\mathbb{N}^m$ into $\mathbb{N}$ altogether, but only countably many of them can be recursive. So an abundance of nonrecursive functions exists, despite the fact that the most commonly met functions (such as the polynomials) were shown in Section 3.3 to be recursive. By catalog item 1 of Section 3.3, the characteristic function of a nonrecursive set is nonrecursive. For example, if $f(a) = 1$ whenever a is the Gödel number of a member of $\mathrm{Cn}\,A_E$ and $f(a) = 0$ otherwise, then f is not recursive.

Normal Form

For any computable function, such as the polynomial function a^2+3a+5, one can in principle design a digital computer into which one feeds a and out of which comes a^2+3a+5 (Fig. 11). But if you then want a different function, you must build a different computer. (Or change the wiring in the one you have.) It was recognized long ago that it is usually more desirable to build a single general-purpose stored-program computer. Into this you feed both a and the program for computing your polynomial (Fig. 12). This "universal" computer requires two inputs, and it will compute any one-place computable function (if supplied with enough memory space), provided that the right program is fed into it. Of course, there are some programs that do not produce any function on $\mathbb{N}$, as many a programmer has, to his sorrow, discovered. (Such programs produce malfunctions instead!)

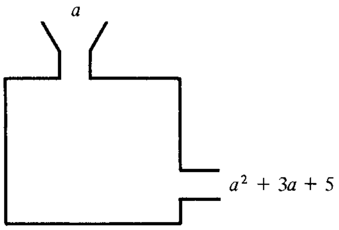


Figure 11. Special-purpose computer.

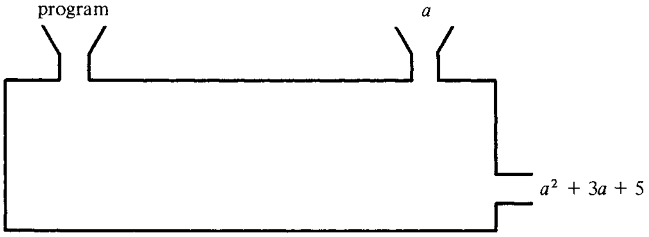


Figure 12. General-purpose computer.

In this subsection and the next, we will repeat what has just been said, but with recursive functions and with proofs. For our universal computer we will have a recursive relation T_1 and a recursive function U. Then for any recursive $f : \mathbb{N} \to \mathbb{N}$ there will exist an e (analogous to the program) such that

$$\begin{aligned} f(a) &= U(\text{the least } k \text{ such that } \langle e, a, k\rangle \in T_1) \\ &= U(\mu k\,\langle e, a, k\rangle \in T_1), \end{aligned}$$

where the second equation is to be understood as being an abbreviation for the first. Actually e will here be the Gödel number of a formula

φ that represents (or at least weakly represents) f in $\operatorname{Cn} A_E$. And the numbers k for which $\langle e, a, k\rangle \in T_1$ will encode both $f(a)$ and $\mathcal{G}$ of a deduction from A_E of $\varphi(\mathbf{S}^a\mathbf{0}, \mathbf{S}^{f(a)}\mathbf{0})$.

Definition. For each positive integer m, let T_m be the $(m+2)$-ary relation to which an $(m+2)$-tuple $\langle e, a_1, \ldots, a_m, k\rangle$ belongs iff

(i) e is the Gödel number of a formula φ in which only $\mathbf{v}_1, \ldots, \mathbf{v}_m, \mathbf{v}_{m+1}$ occur free;

(ii) k is a sequence number of length 2, and $(k)_0$ is $\mathcal{G}$ of a deduction from A_E of $\varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}, \mathbf{S}^{(k)_1}\mathbf{0})$.

The idea here is that for any one-place recursive function f we can first of all take e to be the Gödel number of a formula φ weakly representing f (as a relation). Then we know that for any a and b,

$$A_E \vdash \varphi(\mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}) \quad \text{iff} \quad b = f(a).$$

So any number k meeting clause (ii) of the definition must equal $\langle (k)_0, f(a)\rangle$, where $(k)_0$ is $\mathcal{G}$ of a deduction of $\varphi(\mathbf{S}^a\mathbf{0}, \mathbf{S}^{f(a)}\mathbf{0})$ from A_E. (We have departed from the usual definition of T_m here by not requiring that k be as small as possible.)

Take for the "upshot" function U the function

$$U(k) = (k)_1.$$

This U is recursive and in the situation described in the preceding paragraph we have $U(k) = f(a)$.

Lemma 36A For each m, the relation T_m is recursive.

Proof, for $m = 2$. $\langle e, a_1, a_2, k\rangle \in T_2$ iff e is the Gödel number of a formula, $\sharp(\forall\, \mathbf{v}_1\, \forall\, \mathbf{v}_2\, \forall\, \mathbf{v}_3) * e$ is the Gödel number of a sentence, k is a sequence number of length 2, and $(k)_0$ is $\mathcal{G}$ of a deduction from A_E of

$$\mathrm{Sb}(\mathrm{Sb}(\mathrm{Sb}(e, \sharp\mathbf{v}_1, g(a_1)), \sharp\mathbf{v}_2, g(a_2)), \sharp\mathbf{v}_3, g((k)_1)),$$

where $g(n) = \sharp\mathbf{S}^n\mathbf{0}$. From Section 3.4 we know all this to be recursive. $\dashv$

Theorem 36B (a) For any recursive function $f : \mathbb{N}^m \to \mathbb{N}$, there is an e such that for all $a_1, \ldots, a_m$,

$$f(a_1, \ldots, a_m) = U(\mu k\, \langle e, a_1, \ldots, a_m, k\rangle \in T_m).$$

(In particular, such a number k exists.)

(b) Conversely, for any e such that $\forall a_1 \cdots a_m \exists k\, \langle e, a_1, \ldots, a_m, k\rangle \in T_m$, the function whose value at $a_1, \ldots, a_m$ is $U(\mu k \langle e, a_1, \ldots, a_m, k\rangle \in T_m)$ is recursive.

PROOF. Part (b) follows immediately from the fact that U and T_m are recursive. As for part (a), we take for e the Gödel number of a formula φ weakly representing f in $\mathrm{Cn}\,A_E$. Given any $\vec{a}$, we know that $A_E \vdash \varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}, \mathbf{S}^{f(\vec{a})}\mathbf{0})$. If we let d be $\mathcal{G}$ of a deduction from A_E of this sentence, then $\langle e, \vec{a}, \langle d, f(\vec{a})\rangle\rangle \in T_m$. Hence there is some k for which $\langle e, \vec{a}, k\rangle \in T_m$. And for any such k, we know that $A_E \vdash \varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}, \mathbf{S}^{(k)_1}\mathbf{0})$, since $(k)_0$ is $\mathcal{G}$ of a deduction. Consequently, $U(k) = (k)_1 = f(\vec{a})$ by our choice of φ. Thus we have $U(\mu k\,\langle e, \vec{a}, k\rangle \in T_m) = f(\vec{a})$. ⊣

This theorem, due to Kleene in 1936, shows that every recursive function is representable in the normal form

$$f(\vec{a}) = U(\mu k\,\langle e, \vec{a}, k\rangle \in T_m).$$

Thus a computing machine able to calculate U and the characteristic function of T_1 is a "universal" computer for one-place recursive functions. The input e corresponds to the program, and it must be chosen with care if any output is to result (i.e., if there is to be any k such that $\langle e, a, k\rangle \in T_1$).

Recursive Partial Functions

The theory of recursive functions becomes more natural if we consider the broader context of partial functions.

DEFINITION. An m-place *partial function* is a function f with $\mathrm{dom}\, f \subseteq \mathbb{N}^m$ and $\mathrm{ran}\, f \subseteq \mathbb{N}$. If $\vec{a} \notin \mathrm{dom}\, f$, then $f(\vec{a})$ is said to be *undefined*. If $\mathrm{dom}\, f = \mathbb{N}^m$, then f is said to be *total*.

The reader is hereby cautioned against reading too much into our choice of the words "partial" and "total" (or the word "undefined," for that matter). A partial function f may or may not be total; the words "partial" and "total" are not antonyms.

We will begin by looking at those partial functions that are informally computable.

*DEFINITION. An m-place partial function f is *computable* iff there is an effective procedure such that (a) given an m-tuple $\vec{a}$ in $\mathrm{dom}\, f$, the procedure produces $f(\vec{a})$; and (b) given an m-tuple $\vec{a}$ not in $\mathrm{dom}\, f$, the procedure produces no output at all.

This definition extends the one previously given for total functions. At that time we proved a result (Theorem 33H), part of which generalizes to partial functions.

***THEOREM 36C** An m-place partial function f is computable iff f (as an $(m + 1)$-ary relation) is effectively enumerable.

Proof. The proof is reminiscent of the proof of another result, Theorem 17E. First suppose we have a way of effectively enumerating f. Given an m-tuple $\vec{a}$, we examine the listing of the relation as the procedure churns it out. If and when an $(m+1)$-tuple beginning with $\vec{a}$ appears, we print out its last component as $f(\vec{a})$.

Conversely, assume that f is computable, and first suppose that f is a one-place partial function. We can enumerate f as a relation by the following procedure:

1. Spend one minute calculating $f(0)$.
2. Spend two minutes calculating $f(0)$, then two minutes calculating $f(1)$.
3. Spend three minutes calculating $f(0)$, three minutes calculating $f(1)$, and three minutes calculating $f(2)$.

And so forth. Of course, whenever one of these calculations produces any output, we place the corresponding pair on the list of members of the relation f.

For a computable m-place partial function, instead of calculating the value of f at $0, 1, 2, \ldots$ we calculate its value at $\langle (0)_0, \ldots, (0)_{m-1}\rangle$, $\langle (1)_0, \ldots, (1)_{m-1}\rangle$, $\langle (2)_0, \ldots, (2)_{m-1}\rangle$, etc. ⊣

In the case of a computable total function f, we were also able to conclude that f was a decidable relation. But this may fail for a nontotal f. For example, let

$$f(a) = \begin{cases} 0 & \text{if } a \in \sharp\text{Cn}\, A_E, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Then f is computable. (We compute $f(a)$ by enumerating $\sharp\text{Cn}\, A_E$ and looking for a.) But f is not a decidable relation, lest $\sharp\text{Cn}\, A_E$ be decidable. On the basis of this example and the foregoing theorem, we select our definition for the precise counterpart of the concept of computable partial function.

Definition. A *recursive partial function* is a partial function that, as a relation, is recursively enumerable.

The reader should be warned that "recursive partial function" is an indivisible phrase; a recursive partial function need *not* be (as relation) recursive. But at least for a total function our terminology is consistent with past practice.

Theorem 36D Let $f : \mathbb{N}^m \to \mathbb{N}$ be a total function. Then f is a recursive partial function iff f is recursive (as a relation).

Proof. If f is recursive (as a relation), then *a fortiori* f is recursively enumerable. Conversely, suppose that f is recursively enumerable. Since f is total,

$$f(\vec{a}) \neq b \iff \exists c[f(\vec{a}) = c \,\&\, b \neq c].$$

The form of the right-hand side shows that the complement of f is also recursively enumerable. Thus by Theorem 35F, f is recursive. ⊣

In first discussing normal form results, we pictured a two-input device (Fig. 13). For any computable partial function, there is some program that computes it. But now the converse holds: Any program will produce some computable *partial* function. Of course many programs will produce the empty function, but that is a computable partial function.

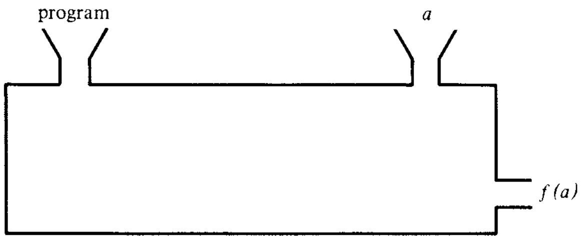


Figure 13. Computer with program for f.

For the recursive partial functions the same considerations apply. Define, for each $e \in \mathbb{N}$, the m-place partial function $[\![e]\!]_m$ by

$$[\![e]\!]_m(a_1, \ldots, a_m) = U(\mu k \,\langle e, a_1, \ldots, a_m, k\rangle \in T_m).$$

The right-hand side is to be understood as undefined if there is no such k. In other words,

$$\vec{a} \in \operatorname{dom}[\![e]\!]_m \quad \text{iff} \quad \exists k \,\langle e, a_1, \ldots, a_m, k\rangle \in T_m,$$

in which case the value $[\![e]\!]_m(\vec{a})$ is given by the above equation.

The following theorem is an improved version of Theorem 36B:

Normal Form Theorem (Kleene, 1943) (a) The $(m + 1)$-place partial function whose value at $\langle e, a_1, \ldots, a_m\rangle$ is $[\![e]\!]_m(a_1, \ldots, a_m)$ is a recursive partial function.

(b) For each $e \geq 0$, $[\![e]\!]_m$ is an m-place recursive partial function.

(c) Any m-place recursive partial function equals $[\![e]\!]_m$ for some e.

Proof. (a) We have

$$[\![e]\!]_m(\vec{a}) = b \Leftrightarrow \exists k[\langle e, \vec{a}, k\rangle \in T_m \,\&\, U(k) = b \,\&\, (\forall k' < k)\langle e, \vec{a}, k'\rangle \notin T_m].$$

The part in square brackets is recursive, so the function (as a relation) is recursively enumerable.

(b) The above proof still applies, e now being held fixed.

(c) Let f be an m-place recursive partial function, so that $\{\langle \vec{a}, b\rangle \mid f(\vec{a}) = b\}$ is recursively enumerable. Hence there is a formula φ that weakly represents this relation in Cn A_E. We claim that $f = [\![\sharp\varphi]\!]_m$. For if $f(\vec{a}) = b$, then $A_E \vdash \varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}, \mathbf{S}^{b}\mathbf{0})$. Hence there is a k such that $\langle \sharp\varphi, \vec{a}, k\rangle \in T_m$. For any such k, $U(k) = b$, since $A_E \nvdash \varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}, \mathbf{S}^{c}\mathbf{0})$ for $c \neq b$. Similarly, if $f(\vec{a})$ is undefined, then $A_E \nvdash \varphi(\mathbf{S}^{a_1}\mathbf{0}, \ldots, \mathbf{S}^{a_m}\mathbf{0}, \mathbf{S}^{c}\mathbf{0})$ for any c, whence $[\![\sharp\varphi]\!]_m$ is undefined here also. $\dashv$

Part (a) of the normal form theorem (in the case $m = 1$) tells us that the function Φ defined by the equation

$$\Phi(e, a) = [\![e]\!]_1(a) = U(\mu k\, \langle e, a, k\rangle \in T_1)$$

is a recursive partial function. And part (c) tells us that Φ is "universal" in the sense that we can get any one-place recursive partial function from Φ by holding the first variable fixed at a suitable value.

The informal counterpart of the universal function Φ is the computer operating system. The operating system takes two inputs, the program e and the data a. And it runs the program on that data. But the operating system itself is *computable* as a two-place partial function.

The proof of normal form theorem gives us a way to compute the values of our "operating system" Φ, albeit in an extremely inefficient way. The straightforward idea of "looking at the program e and doing what it says to the data a" has been obscured, to say the least.

The function $[\![e]\!]_m$ is said to be the m-place recursive partial function with *index* e. Part (c) of the normal form theorem tells us that every recursive partial function has an index. The proof shows that the Gödel number of a formula weakly representing a function is always an index of the function.

We now have a convenient indexing $[\![0]\!]_1, [\![1]\!]_1, \ldots$ of all the one-place recursive partial functions. Function $[\![e]\!]_1$ is produced by the "instructions" encoded by e. Of course, that function will be empty unless e is the Gödel number of a formula and certain other conditions are met.

All the recursive total functions are included in our enumeration of recursive partial functions. But we cannot tell effectively by looking at a number e whether or not it is the index of a total function:

Theorem 36E $\{e \mid [\![e]\!]_1 \text{ is total}\}$ is not recursive.

Proof. Call this set A. Consider the function defined by

$$f(a) = \begin{cases} [\![a]\!]_1(a) + 1 & \text{if } a \in A, \\ 0 & \text{if } a \notin A. \end{cases}$$

Then f, by its construction, is total. Is it recursive? We have

$$f(a) = b \iff [(a \notin A \;\&\; b = 0) \text{ or } (a \in A \;\&\; \exists k(\langle a, a, k\rangle \in T_1 \\ \&\; b = U(k) + 1 \;\&\; (\forall j < k)\langle a, a, j\rangle \notin T_1))].$$

Thus if A is recursive, then f (as a relation) is recursively enumerable. But then f is a total recursive function, and so equals $[\![e]\!]_1$ for some $e \in A$. But $f(e) = [\![e]\!]_1(e) + 1$, so we cannot have $f = [\![e]\!]_1$. This contradiction shows that A cannot be recursive. ⊣

It is not hard to show that A is in Π_2. This classification is the best possible, as it can be shown that A is not in Σ_2.

Theorem 36F The set

$$K = \{a \mid [\![a]\!]_1(a) \text{ is defined}\}$$

is recursively enumerable but not recursive.

Proof. K is recursively enumerable, since $a \in K \Leftrightarrow \exists k\, \langle a, a, k\rangle \in T_1$. To see that K cannot be recursive, consider the function defined by

$$g(a) = \begin{cases} [\![a]\!]_1(a) + 1 & \text{if } a \in K, \\ 0 & \text{if } a \notin K. \end{cases}$$

This is a total function. Exactly as in the preceding theorem, we have that K cannot be recursive. ⊣

Corollary 36G (Unsolvability of the Halting Problem) The relation

$$\{\langle e, a\rangle \mid [\![e]\!]_1(a) \text{ is defined}\}$$

is not recursive.

Proof. We have $a \in K$ iff the pair $\langle a, a\rangle$ belongs to this relation. (Thus the problem of membership in K is "reducible" to the halting problem.) If this relation were recursive, then K would be, which is not the case. ⊣

This corollary tells us that there is no effective way to tell, given a program e for a recursive partial function and an input a, whether or not the function $[\![e]\!]_1$ is defined at a.

We can obtain an indexing of the recursively enumerable relations by using the following characterization.

Theorem 36H A relation on $\mathbb{N}$ is recursively enumerable iff it is the domain of some recursive partial function.

Proof. The domain of any recursively enumerable relation is also recursively enumerable; cf. part 4 of Theorem 35E. In particular, the domain of any recursive partial function is recursively enumerable.

Conversely, let Q be any recursively enumerable relation, where

$$\vec{a} = Q \Leftrightarrow \exists b\, \langle \vec{a}, b\rangle \in R$$

with R recursive. Let

$$f(\vec{a}) = \mu b\, \langle \vec{a}, b\rangle \in R;$$

i.e.,

$$f(\vec{a}) = b \iff \langle \vec{a}, b\rangle \in R \;\&\; (\forall c < b)\langle \vec{a}, c\rangle \notin R.$$

Then f, as a relation, is recursive. Hence f is a recursive partial function. Clearly its domain is Q. $\dashv$

Thus our indexing of the recursive partial functions induces an indexing of the recursively enumerable relations. Define

$$W_e = \operatorname{dom}[\![e]\!]_1.$$

Then $W_0, W_1, W_2, \ldots$ is a list of all recursively enumerable subsets of $\mathbb{N}$. In Theorem 36E we showed that $\{e \mid W_e = \mathbb{N}\}$ is not recursive. Similarly, Theorem 36F asserts that $\{e \mid e \in W_e\}$ is not recursive. Define a relation Q by

$$Q = \{\langle e, a\rangle \mid a \in W_e\}.$$

Then Q is recursively enumerable, since $\langle e, a\rangle \in Q \Leftrightarrow \exists k\, \langle e, a, k\rangle \in T_1$. Furthermore, Q is universal for recursively enumerable sets, in the sense that for any recursively enumerable $A \subseteq \mathbb{N}$ there is some e such that $A = \{a \mid \langle e, a\rangle \in Q\}$. The unsolvability of the halting problem can be stated: Q is not recursive.

We can apply the classical diagonal argument to "diagonalize out" of the list $W_0, W_1, W_2, \ldots$ of recursively enumerable sets. The set $\{a \mid a \notin W_a\}$ cannot coincide with any W_q. In fact this set is exactly $\overline{K}$, the complement of the set K in Theorem 36F. Because

$$q \in \overline{K} \iff q \notin W_q,$$

the set $\overline{K}$ cannot equal any W_q; the number q witnesses the inequality of the two sets $\overline{K}$ and W_q.

And there is more: Whenever W_q is a recursively enumerable *subset* of $\overline{K}$, that is, $W_q \subseteq \overline{K}$, then we can produce a number in $\overline{K}$ that is not in W_q. Such a number is q itself. To see this, observe that in the line displayed in the preceding paragraph we cannot have both sides false ($q \in K$ and $q \in W_q$) because $W_q \subseteq \overline{K}$. So both sides are true.

Theorem 36F asserts that K, although recursively enumerable, is not recursive. To show non-recursiveness, it suffices to show that its complement $\overline{K}$ is not recursively enumerable. The preceding paragraph does this in a particularly strong way, thereby giving us a second proof of Theorem 36F.

At this point, let us reconsider the Gödel incompleteness theorem, from the computability point of view.

The set K is recursively enumerable (i.e., Σ_1). It follows (cf. Theorem 35K) that K is arithmetical; that is, K is definable in the structure $\mathfrak{N}$.

So there is a formula $\varkappa(\boldsymbol{v}_1)$ with just $\boldsymbol{v}_1$ free that defines K in $\mathfrak{N}$. And so the set $\overline{K}$ is defined in $\mathfrak{N}$ by the formula $\neg\,\varkappa(\boldsymbol{v}_1)$. Thus we have

$$a \in \overline{K} \iff (\neg\,\varkappa(\mathbf{S}^a\mathbf{0})) \in \operatorname{Th}\mathfrak{N}.$$

This fact tells how we can "reduce" questions about membership in the set $\overline{K}$ to questions about $\operatorname{Th}\mathfrak{N}$. Imagine that we are given a number a, and we want to know whether or not $a \in \overline{K}$. We can *compute* the number $\sharp(\neg\,\varkappa(\mathbf{S}^a\mathbf{0}))$. (Informally, it is clear that we can *effectively* compute this number. Formally, we apply item 5 from Section 3.4 to make sure we can *recursively* compute the number.) If we somehow had an oracle for $\sharp\operatorname{Th}\mathfrak{N}$ (i.e., a magic device that, given a number, would tell us whether or not that number was in $\sharp\operatorname{Th}\mathfrak{N}$), then we could answer the question "Is $a \in \overline{K}$?"

Now let us eliminate the magic. For sets A and B of natural numbers, we say that A is *many-one reducible* to B (in symbols, $A \leq_{\mathrm{m}} B$) iff there exists a total recursive function f such that for every number a,

$$a \in A \iff f(a) \in B.$$

The earlier example tells us that $\overline{K} \leq_{\mathrm{m}} \sharp\operatorname{Th}\mathfrak{N}$. More generally, the argument shows that any arithmetical set is many-one reducible to $\sharp\operatorname{Th}\mathfrak{N}$.

Lemma 36I Assume that A and B are sets of natural numbers with $A \leq_{\mathrm{m}} B$.

(a) If B is recursive, then A is also recursive.

(b) If B is recursively enumerable, the A is also recursively enumerable.

(c) If B is Σ_n for some n, the A is also Σ_n for that n.

Proof. Part (a) is already familiar; it was, in different terminology, catalog item 2 in Section 3.3.

Part (b) is essentially part (a) "plus a quantifier." That is, because B is recursively enumerable, we know that for some recursive binary relation Q,

$$c \in B \iff \exists b\, Q(c, b).$$

If f is the total recursive function that many-one reduces A to B, then every number a,

$$a \in A \iff f(a) \in B \iff \exists b[Q(f(a), b)].$$

The part in square brackets is recursive (i.e., $\{\langle a, b\rangle \mid Q(f(a), b)\}$ is recursive), as in part (a) of the lemma. So we have A in the required form to be recursively enumerable.

Part (c) is essentially part (a) "plus n quantifiers" and is proved like part (b). $\dashv$

Our reason for examining the particular set $\overline{K}$ is that it gives us the following consequence:

Gödel Incompleteness Theorem Th $\mathfrak{N}$ is not recursively axiomatizable.

Proof. Th $\mathfrak{N}$ cannot be recursively enumerable, lest $\overline{K}$ be recursively enumerable, by the preceding lemma. But any recursively axiomatizable theory would be recursively enumerable (item 20 of Section 3.4; also Theorem 35I). $\dashv$

In starkest terms, the situation is this: Any recursively axiomatizable theory is recursively enumerable. But Th $\mathfrak{N}$ is not recursively enumerable. So any recursively axiomatizable subtheory must be incomplete.

It will be worth while to go over this proof again, but replacing negative statements (such-and-such a set does *not* have a particular property) by positive statements.

Assume that T is any recursively axiomatizable subtheory of Th $\mathfrak{N}$. (So by the above theorem, T is incomplete.) We want to lay our hands on a sentence demonstrating the incompleteness.

We have made a total recursive function f that many-one reduces $\overline{K}$ to $\sharp$ Th $\mathfrak{N}$, namely $f(a) = \sharp(\neg\, \varkappa(\mathbf{S}^a\mathbf{0}))$; then for every a,

$$a \in \overline{K} \iff f(a) \in \sharp\,\mathrm{Th}\,\mathfrak{N}.$$

And $f(a)$ is (the Gödel number of) the sentence saying "$a \notin K$."

Consider the set J of numbers defined by the condition

$$a \in J \iff f(a) \in \sharp T.$$

Thus J is the set of numbers that T "knows" are not in K. There are two observations to be made concerning J:

First, J is recursively enumerable. It is many-one reduced by f to the recursively enumerable set $\sharp T$; apply Lemma 36I(b).

Secondly, $J \subseteq \overline{K}$. We have $T \subseteq$ Th $\mathfrak{N}$, so if T knows that $a \notin K$, then really $a \notin K$:

$$a \in J \iff f(a) \in \sharp T \Longrightarrow f(a) \in \sharp\,\mathrm{Th}\,\mathfrak{N} \iff a \in \overline{K}.$$

So J is a recursively enumerable subset of $\overline{K}$. It is a *proper* subset, because $\overline{K}$ is not recursively enumerable. That is, there is some number q with $q \in \overline{K}$ and $q \notin J$. Consequently, $f(q) \in \sharp$ Th $\mathfrak{N}$ but $f(q) \notin \sharp T$. That is, the sentence $(\neg\, \varkappa(\mathbf{S}^q\mathbf{0}))$ is true (in $\mathfrak{N}$) but fails to be in T, thereby demonstrating the incompleteness of T.

And what does this sentence "say"? For q, we can take any number for which $W_q = J$. Then $q \in \overline{K}$ and $q \notin J$.

Here then is the situation:

$$\begin{array}{lll} (\neg \varkappa(\mathbf{S}^q\mathbf{0})) & \text{says } q \notin K & \\ & \text{i.e., } q \notin W_q & \\ & \text{i.e., } q \notin J & \text{since } W_q = J \\ & \text{i.e., } f(q) \notin \sharp T & \text{by definition of } J \\ & \text{i.e., } T \nvdash (\neg \varkappa(\mathbf{S}^q\mathbf{0})). & \end{array}$$

The sentence we made to witness the incompleteness of T asserts its own unprovability in the axiomatizable theory T!

The computability approach and the self-reference approach to Gödel's incompleteness theorem are not so different after all. Moreover, the computability approach is close to the diagonalization approach (of Section 3.0), but with the diagonal argument moved to a different context.

Reduction of Decision Problems[1]

Suppose we have a two-place recursive partial function f. Then we claim that, for example, the function g defined by

$$g(a) = f(3, a)$$

is also a recursive partial function. On the basis of informal computability this is clear; one computes g by plugging in 3 for the first variable and then following the instructions for f. A proof can be found by formalizing this argument. There is some formula $\varphi = \varphi(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ that weakly represents f (as a relation) in $\operatorname{Cn} A_E$. Then g is weakly represented by $\varphi(\mathbf{S}^3\mathbf{0}, \mathbf{v}_1, \mathbf{v}_2)$, provided that $\mathbf{v}_1$ and $\mathbf{v}_2$ are substitutable in φ for $\mathbf{v}_2$ and $\mathbf{v}_3$. (If not, we can always use an alphabetic variant of φ.)

Now all this is not very deep. But by standing back and looking at what was said, we perceive a more subtle fact. We were able to transform effectively the instructions for f into instructions for g. So there should be a recursive function that, given an index for f and the number 3, will produce an index for g. The following formulation of this fact is sometimes known by the cryptic name of "the S-m-n theorem."

Parameter Theorem For each $m \geq 1$ and $n \geq 1$ there is a recursive function ρ such that for any $e, \vec{a}, \vec{b}$,

$$[\![e]\!]_{m+n}(a_1, \ldots, a_m, b_1, \ldots, b_n) = [\![\rho(e, a_1, \ldots, a_m)]\!]_n(b_1, \ldots, b_n).$$

(Equality here means of course that if one side is defined, then so also is the other side, and the values coincide. Sometimes a special symbol "$\simeq$" is used for this role.)

[1] The remainder of this section can be skipped on a first reading.

On the left side of the equation $\vec{a}$ consists of arguments for the function $[\![e]\!]_{m+n}$; on the right side $\vec{a}$ consists of parameters upon which the function $[\![\rho(e, \vec{a})]\!]_n$ depends. In the example we had $m = n = 1$ and $a_1 = 3$. Since ρ depends on m and n, the notation "ρ_n^m" would be logically preferable. But, in fact, we will use simply "ρ."

Proof, for $m = n = 1$. It is possible to give a proof along the lines indicated by the discussion that preceded the theorem. But to avoid having to cope with alphabetic variants, we will adopt a slightly different strategy.

We know from the normal form theorem that the three-place partial function h defined by

$$h(e, a, b) = [\![e]\!]_2(a, b)$$

is a recursive partial function. Hence there is a formula ψ that weakly represents h (as a relation). We may suppose that in ψ the variables $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are not quantified. We can then take

$$\begin{aligned}\rho(e, a) &= \sharp\psi(\mathbf{S}^e\mathbf{0}, \mathbf{S}^a\mathbf{0}, \boldsymbol{v}_1, \boldsymbol{v}_2)\\ &= \mathrm{Sb}(\mathrm{Sb}(\mathrm{Sb}(\mathrm{Sb}(\sharp\psi, \sharp\boldsymbol{v}_1, \sharp\mathbf{S}^e\mathbf{0}), \sharp\boldsymbol{v}_2, \sharp\mathbf{S}^a\mathbf{0}), \sharp\boldsymbol{v}_3, \sharp\boldsymbol{v}_1), \sharp\boldsymbol{v}_4, \sharp\boldsymbol{v}_2).\end{aligned}$$

Then $\rho(e, a)$ is the Gödel number of a formula weakly representing the function $g(b) = [\![e]\!]_2(a, b)$. Hence it is an index of g.

⊣

We will utilize the parameter theorem to show that certain sets are *not* recursive. We already know that $K = \{a \mid [\![a]\!]_1(a) \text{ is defined}\}$ is not recursive. For a given nonrecursive set A we can sometimes find a (total) recursive function g such that

$$a \in K \Leftrightarrow g(a) \in A$$

or a (total) recursive function g' such that

$$a \notin K \Leftrightarrow g'(a) \in A.$$

In either case it then follows at once that A cannot be recursive lest K be. In the former case we have $K \leq_{\mathrm{m}} A$ and A is not Π_1 (by Lemma 36I); in the latter case $\overline{K} \leq_{\mathrm{m}} A$ and A is not Σ_1. In either case, A is not recursive. The function g or g' can often be obtained from the parameter theorem.

Example. $\{a \mid W_a = \varnothing\}$ is not recursive.

Proof. Call this set A. First, note that $A \in \Pi_1$, since $W_a = \varnothing$ iff $\forall b \forall k\, \langle a, b, k\rangle \notin T_1$. Consequently, K cannot be many-one reducible to A, but it is reasonable to hope that $\overline{K}$ might be. That is, we want a total recursive function g such that

$$[\![a]\!]_1(a) \text{ is undefined} \;\Leftrightarrow\; \mathrm{dom}[\![g(a)]\!]_1 = \varnothing.$$

This will hold if for all b, $[\![g(a)]\!]_1(b) = [\![a]\!]_1(a)$. So start with the recursive partial function

$$f(a, b) = [\![a]\!]_1(a)$$

and let $g(a) = \rho(\hat{f}, a)$, where $\hat{f}$ is an index for f. Then

$$[\![g(a)]\!]_1(b) = [\![\rho(\hat{f}, a)]\!]_1(b) = f(a, b) = [\![a]\!]_1(a).$$

Thus this g shows that $\overline{K}$ is many-one reducible to A. ⊣

Theorem 36J (Rice, 1953) Let $\mathcal{C}$ be a set of one-place recursive partial functions. Then the set $\{e \mid [\![e]\!]_1 \in \mathcal{C}\}$ of indices of members of $\mathcal{C}$ is recursive iff either $\mathcal{C}$ is empty or $\mathcal{C}$ contains all one-place recursive partial functions.

Proof. Only one direction requires proof. Let $I_\mathcal{C} = \{e \mid [\![e]\!]_1 \in \mathcal{C}\}$ be the set of indices of members of $\mathcal{C}$.

Case I: The empty function $\varnothing$ is not in $\mathcal{C}$. If nothing at all is in $\mathcal{C}$ we are done, but suppose some function ψ is in $\mathcal{C}$. We can show that K is many-one reducible to $I_\mathcal{C}$ if we have a recursive total function g such that

$$[\![g(a)]\!]_1 = \begin{cases} \psi & \text{if } a \in K, \\ \varnothing & \text{if } a \notin K. \end{cases}$$

For then $a \in K \Leftrightarrow [\![g(a)]\!]_1 \in \mathcal{C} \Leftrightarrow g(a) \in I_\mathcal{C}$.

We can obtain g from the parameter theorem by defining

$$g(a) = \rho(e, a),$$

where

$$[\![e]\!]_2(a, b) = \begin{cases} \psi(b) & \text{if } a \in K, \\ \text{undefined} & \text{if } a \notin K. \end{cases}$$

The above *is* a recursive partial function, since

$$[\![e]\!]_2(a, b) = c \Leftrightarrow a \in K \ \&\ \psi(b) = c$$

and the right-hand side is recursively enumerable.

Case II: $\varnothing \in \mathcal{C}$. Then apply case I to the complement $\overline{\mathcal{C}}$ of $\mathcal{C}$. We can then conclude that $I_{\overline{\mathcal{C}}}$ is not recursive. But $I_{\overline{\mathcal{C}}}$ is the complement of $I_\mathcal{C}$, so $I_\mathcal{C}$ cannot be recursive.

Thus in either case, $I_\mathcal{C}$ is not recursive. ⊣

Examples. For any fixed e, the set $\{a \mid W_a = W_e\}$ is not recursive, as a consequence of Rice's theorem. In particular, $\{a \mid W_a = \varnothing\}$ is not recursive, a result proved in an earlier example. For two other applications of Rice's theorem, we can say that $\{a \mid W_a \text{ is infinite}\}$ and $\{a \mid W_a \text{ is recursive}\}$ are not recursive.

Register Machines

There are many equivalent definitions of the class of recursive functions. Several of these definitions employ idealized computing devices. These computing devices are like digital computers but are free of any limitation on memory space. The first definition of this type was published by Alan Turing in 1936; similar work was done by Emil Post at roughly the same time. We will give here a variation on this theme, due to Shepherdson and Sturgis (1963).

A *register machine* will have a finite number of registers, numbered $1, 2, \ldots, K$. Each register is capable of storing a natural number of any magnitude. The operation of the machine will be determined by a *program*. A program is a finite sequence of *instructions*, drawn from the following list:

I r (where $1 \leq r \leq K$). "Increment r." The effect of this instruction is to increase the contents of register r by 1. The machine then proceeds to the next instruction in the program.

D r (where $1 \leq r \leq K$). "Decrement r." The effect of this instruction depends on the contents of register r. If that number is nonzero, it is decreased by 1 and the machine then proceeds, not to the next instruction, but to the following one. But if the number in register r is zero, the machine just proceeds to the next instruction. In summary: The machine tries to decrement register r and skips an instruction if it is successful.

T q (where q is an integer–positive, negative, or zero). "Transfer q." All registers are left unchanged. The machine takes as its next instruction the qth instruction following this one in the program (if $q \geq 0$), or the $|q|$th instruction preceding this one (if $q < 0$). The machine halts if there is no such instruction in the program. An instruction of T 0 results in a loop, with the machine executing this one instruction over and over again.

Examples

1. Program to clear register 7.

```
D 7          Try to decrement 7.
T 2
T −2         Go back and repeat.
             Halt.
```

2. Program to move a number from register r to register s.

```
Clear register s    (Use the program of the first example.)
D r                 Take 1 from r.
T 3                 Halt when zero.
I s                 Add 1 to s.
T −3                Repeat.
```

This program has seven instructions altogether. It leaves a zero in register r.

3. Program to add register 1 to registers 2 and 3.

```
D  1
T  4
I  2
I  3
T  −4
```

4. (Addition) Say that a and b are in registers 1 and 2. We want $a + b$ in register 3, and we want to leave a and b still in registers 1 and 2 at the end.

	Register contents			
Clear register 3.	a	b	0	
Move number from register 1 to register 4.	0	b	0	a
Add register 4 to registers 1 and 3.	a	b	a	0
Move number from register 2 to register 4.	a	0	a	b
Add register 4 to registers 2 and 3.	a	b	$a + b$	0

This program has 27 instructions as it is written, but three of them are unnecessary. (In the fourth line we begin by clearing register 4, which is already clear.) At the end we have the number a back in register 1. But during the program register 1 must be cleared; this is the only way of determining the number a.

5. (Subtraction) Let $a \dot{-} b = \max(a - b, 0)$. We leave this program to the reader (Exercise 11).

Now suppose f is an n-place partial function on $\mathbb{N}$. Possibly there will be a program P such that if we start a register machine (having all the registers to which P refers) with $a_1, \ldots, a_n$ in registers $1, \ldots, n$ and apply program P, then the following conditions hold:

(i) If $f(a_1, \ldots, a_n)$ is defined, then the calculation eventually terminates with $f(a_1, \ldots, a_n)$ in register $n + 1$. Furthermore, the calculation terminates by seeking a $(p+1)$st instruction, where p is the length of P.

(ii) If $f(a_1, \ldots, a_n)$ is undefined, then the calculation never terminates.

If there is such a program P, we say that P *calculates* f.

Theorem 36K Let f be a partial function. Then there is a program that calculates f iff f is a recursive partial function.

Thus by using register machines we arrive at exactly the class of recursive partial functions, a class we originally defined in terms of representability in consistent finitely axiomatizable theories. The fact that such different approaches produce the same class of partial functions is evidence that this is a significant class.

Outline of Proof. To show that the functions calculable by register machines are recursive partial functions, one "arithmetizes calculations" in the same spirit as we arithmetized deductions in Section 3.4. That is, one assigns Gödel numbers to programs and to sequences of memory configurations. One then verifies that the relevant concepts, translated into numerical relations by the Gödel numbering, are all recursive. (After going through this, one perceives that, from a sufficiently general point of view, deductions and calculations are really about the same sort of thing.)

Conversely, to show that the recursive partial functions are calculable by register machines, one can work through Sections 3.3 and 3.4 again, but where functions were previously shown to be representable in Cn A_E, they must now be shown to be calculable by register machines. This is not as hard as it might sound, since after the first few pages, the proofs are all the same as the ones used before. There is a reason for this similarity. It can be shown that the class of all recursive functions is generated from a certain handful of recursive functions by the operation of composition (in the sense of Theorem 33L) and the "least-zero" operator (Theorem 33M). Much of the work in Sections 3.3 and 3.4 amounts to a verification of this fact. Thus once one has shown that each function in this initial handful is calculable by a register machine and that the class of functions calculable by register machines is closed under composition and the least-zero operator, then the earlier work can be carried over, yielding the calculability of all recursive functions. ⊣

Exercises

1. Define functions f and g by

$$f(n) = \begin{cases} 0 & \text{if Goldbach's conjecture is true,} \\ 1 & \text{otherwise;} \end{cases}$$

$$g(n) = \begin{cases} 0 & \text{if in the decimal expansion of } \pi \text{ there} \\ & \text{is a run of at least } n \text{ consecutive 7's,} \\ 1 & \text{otherwise.} \end{cases}$$

Is f recursive? Is g recursive? (Goldbach's conjecture says that every even integer greater than 2 is the sum of two primes. The first edition of this book used Fermat's last theorem here.)

2. Define the "diagonal" function $d(a) = [\![a]\!]_1(a) + 1$.
   (a) Show that d is a recursive partial function.
   (b) By part (a), we have $d = [\![e]\!]_1$ for a certain number e. So on the one hand, $d(e) = [\![e]\!]_1(e)$ and on the other hand $d(e) = [\![e]\!]_1(e)+1$. Can we cancel, to conclude that $0 = 1$? *Suggestion*: Use the special symbol "$\simeq$" to mean that either both sides of the equation are undefined, or both sides are defined and equal. Restate the argument in this notation.
3. (a) Show that the range of any recursive partial function is recursively enumerable.
   (b) Show that the range of a strictly increasing (i.e., $f(n) < f(n+1)$) total recursive function f is recursive.
   (c) Show that the range of a nondecreasing (i.e., $f(n) \leq f(n+1)$) total recursive function f is recursive.
4. (a) Let A be a nonempty recursively enumerable subset of $\mathbb{N}$. Show that A is the range of some total recursive function.
   (b) Show that any infinite recursively enumerable subset of $\mathbb{N}$ includes an infinite recursive subset.
5. Show that every recursive partial function has infinitely many indices.
6. Give an example of a function f and a number e such that for all a,

$$f(a) = U(\mu k\, \langle e, a, k\rangle \in T_1)$$

   but e is not the Gödel number of a formula weakly representing in Cn A_E.
7. Show that the parameter theorem can be strengthened by requiring ρ to be one-to-one.
8. Recall that the union of two recursively enumerable sets is recursively enumerable (Exercise 7 of Section 3.5). Show that there is a total recursive function g such that $W_{g(a,b)} = W_a \cup W_b$.
9. Show that $\{a \mid W_a$ has two or more members$\}$ is in Σ_1 but not in Π_1.
10. Show that there is no recursively enumerable set A such that $\{[\![a]\!]_1 \mid a \in A\}$ equals the class of total recursive functions on $\mathbb{N}$.
11. Give register machine programs that calculate the following functions:
    (a) Subtraction, $a \dot{-} b = \max(a - b, 0)$.
    (b) Multiplication, $a \cdot b$.
    (c) $\max(a, b)$.
12. Assume that there is a register machine program that calculates the n-place partial function f. Show that given any positive

integers $r_1, \ldots, r_n$ (all distinct), p, and k, we can find a program Q such that whenever we start a register machine (having all the registers to which Q refers) with $a_1, \ldots, a_n$ in registers $r_1, \ldots, r_n$ and apply program Q, then (i) if $f(a_1, \ldots, a_n)$ is defined, then the calculation eventually terminates with $f(a_1, \ldots, a_n)$ in register p, with the contents of registers $1, 2, \ldots, k$ (except for register p) the same as their initial contents, and furthermore the calculation terminates by seeking a $(q + 1)$st instruction, where q is the length of Q; (ii) if $f(a_1, \ldots, a_n)$ is undefined, then the calculation never terminates.

13. Let $g : \mathbb{N}^{n+1} \to \mathbb{N}$ be a (total) function that is calculated by some register machine program. Let $f(a_1, \ldots, a_n) = \mu b[g(a_1, \ldots, a_n, b) = 0]$, where the right-hand side is undefined if no such b exists. Show that the partial function f can be calculated by some register machine program.

14. Show that the following sets have the given location in the arithmetical hierarchy. (In each case, the given location is the best possible, but we will not prove that fact.)
    (a) $\{e \mid [\![e]\!]_1 \text{ is total}\}$ is Π_2.
    (b) $\{e \mid W_e \text{ is finite}\}$ is Σ_2.
    (c) $\{e \mid W_e \text{ is cofinite}\}$ is Σ_3.
    (d) $\{e \mid W_e \text{ is recursive}\}$ is Σ_3.

15. Let $\mathit{Tot} = \{e \mid [\![e]\!]_1 \text{ is total}\}$. Clearly $\mathit{Tot} \subset K$. Show that there is no recursive set A with

$$\mathit{Tot} \subseteq A \subseteq K.$$

Remark: This result includes Theorems 36E and 36F; the proofs used there can be adapted here.

16. (a) Show that each Π_2 set of natural numbers is, for some number e, the set

$$\{a \mid \forall b \exists c\, T_2(e, a, b, c)\}.$$

(b) Show that the set $\{a \mid \text{not } \forall b \exists c\, T_2(a, a, b, c)\}$ is Σ_2 but not Π_2.

*(c) Generalize parts (a) and (b) to show that for each n, there is a set that is Σ_n but not Π_n.

17. Assume that A is a set of natural numbers that is arithmetical but is not Π_m. Use the argument of page 256 to show that $\sharp \operatorname{Th} \mathfrak{N}$ is not Σ_m.

Remark: Exercises 16 and 17 give a proof of Tarski's theorem (that $\sharp \operatorname{Th} \mathfrak{N}$ is not arithmetical) from computability theory.

SECTION 3.7
Second Incompleteness Theorem

Let us return once again to item 20 in Section 3.4. Suppose that we have a recursively axiomatizable theory T given by a recursive set A of axioms (i.e., $\sharp A$ is recursive). Then as in item 20

$$\begin{aligned} a \in \sharp T \iff \exists d\,[&d \text{ is the number of a deduction from } A \\ &\text{and the last component of } d \text{ is } a \text{ and } a \\ &\text{is the Gödel number of a sentence}]. \end{aligned}$$

The set of pairs $\langle a, d\rangle$ meeting the condition in brackets is recursive; let $\pi(\mathbf{v}_1, \mathbf{v}_2)$ be a formula — chosen in some natural way — that numeralwise represents that binary relation in A_E.

For any sentence σ, we can express "$T \vdash \sigma$" by the sentence $\exists\, \mathbf{v}_2\, \pi(\mathbf{S}^{\sharp\sigma}\mathbf{0}, \mathbf{v}_2)$. Let us give that sentence a name; define

$$\operatorname{Prb}_T \sigma = \exists\, \mathbf{v}_2 \pi(\mathbf{S}^{\sharp\sigma}\mathbf{0}, \mathbf{v}_2).$$

(Here Prb abbreviates "provable." The subscript should perhaps be "A" instead of "T"; in constructing the sentence we utilize the recursiveness of the set A of axioms.)

Lemma 37A Let T be a recursively axiomatizable theory as above.

(a) Whenever $T \vdash \sigma$ then $A_E \vdash \operatorname{Prb}_T \sigma$.

(b) If in addition T includes A_E, then T has the "reflection" property:

$$T \vdash \sigma \Longrightarrow T \vdash \operatorname{Prb}_T \sigma.$$

Proof. If $T \vdash \sigma$ then we can let d be the number of a deduction of σ from the axioms A for T. We have $A_E \vdash \pi(\mathbf{S}^{\sharp\sigma}\mathbf{0}, \mathbf{S}^d\mathbf{0})$, and hence $A_E \vdash \operatorname{Prb}_T \sigma$. This gives part (a), from which part (b) follows immediately. ⊣

Thus under modest assumptions, whenever T proves a sentence, it *knows* that it proves the sentence. Note that part (b) does *not* say that $T \vdash (\sigma \rightarrow \operatorname{Prb}_T \sigma)$. For example, if σ is true (in $\mathfrak{N}$) but unprovable from A_E, then the sentence $(\sigma \rightarrow \operatorname{Prb}_{A_E} \sigma)$ is *not* provable from A_E, and in fact is false in $\mathfrak{N}$.

Returning now to the proof of the Gödel incompleteness theorem (in the self-reference approach), we can apply the fixed-point lemma to obtain a sentence σ asserting its own unprovability in T:

$$A_E \vdash (\sigma \leftrightarrow \neg \operatorname{Prb}_T \sigma).$$

The following lemma provides part of the incompleteness theorem (the other part being Exercise 2 in Section 3.5):

Lemma 37B Let T be a recursively axiomatizable theory including A_E and let σ be obtained from the fixed-point lemma as above. If T is consistent, then $T \nvdash \sigma$.

Proof

$$\begin{aligned} T \vdash \sigma &\Rightarrow T \vdash \mathrm{Prb}_T\, \sigma && \text{by reflection} \\ &\Rightarrow T \vdash \neg\, \sigma && \text{by choice of } \sigma \end{aligned}$$

whence T is inconsistent. ⊣

So far, this lemma merely reflects ideas employed in Section 3.5, and the proof of Lemma 37B was not very complex. And that is exactly the point: The proof is *not* very complex, so perhaps it can be carried out *within* the theory T, if T is "sufficiently strong." That, we can hope that the steps

$$\begin{aligned} \mathrm{Prb}_T\, \sigma &\rightarrow \mathrm{Prb}_T\, \mathrm{Prb}_T\, \sigma \\ &\rightarrow \mathrm{Prb}_T\, \neg\, \sigma \\ &\rightarrow \mathrm{Prb}_T\, \mathbf{0} = \mathbf{S0} \end{aligned}$$

can be carried out in a sufficiently strong extension T of A_E.

If so, we get a remarkable conclusion. Let Cons T be the sentence $\neg\, \mathrm{Prb}_T\, \mathbf{0} = \mathbf{S0}$, which we think of as saying "T is consistent." (Here $\mathbf{0} = \mathbf{S0}$ is chosen simply as a convenient sentence refutable from A_E.) If T lets us carry out the steps in the preceding paragraph, then we can conclude:

$$T \nvdash \mathrm{Cons}\, T, \qquad \text{unless } T \text{ is } \textit{in}\text{consistent}$$

(Of course, an inconsistent theory contains every sentence, including sentences asserting — falsely — the theory's consistency. The situation we are finding here is that, under suitable assumptions, this is the *only* way that a theory can prove its own consistency.) Let's check the details: Suppose $T \vdash \mathrm{Cons}\, T$. Then by the preceding paragraph, $T \vdash \neg\, \mathrm{Prb}_T\, \sigma$. By choice of σ, we then get $T \vdash \sigma$. Lemma 37B then applies.

To make matters less vague, call the theory T *sufficiently strong* if it meets the following three "derivability" conditions.

1. $A_E \subseteq T$. This implies by Lemma 37A that T has the reflection property, $T \vdash \sigma \Rightarrow T \vdash \mathrm{Prb}_T\, \sigma$.

2. For any sentence σ, $T \vdash (\mathrm{Prb}_T\, \sigma \rightarrow \mathrm{Prb}_T\, \mathrm{Prb}_T\, \sigma)$. This is the reflection property, formalized within T.

3. For any sentences ρ and σ, $T \vdash (\mathrm{Prb}_T\, (\rho \rightarrow \sigma) \rightarrow (\mathrm{Prb}_T\, \rho \rightarrow \mathrm{Prb}_T\, \sigma))$. This is modus ponens, formalized within T.

Formalized Lemma 37B Assume that T is a sufficiently strong recursively axiomatizable theory, and let σ be a sentence such that

$$A_E \vdash (\sigma \leftrightarrow \neg\, \mathrm{Prb}_T\, \sigma).$$

Then $T \vdash (\mathrm{Cons}\, T \rightarrow \neg\, \mathrm{Prb}_T\, \sigma)$.

Proof. We put the pieces together carefully. By the choice of σ we get

$$T \vdash (\sigma \longrightarrow (\mathrm{Prb}_T\, \sigma \longrightarrow \mathbf{0} = \mathbf{S0})).$$

Applying first reflection and then formalized modus ponens to this formula yields

$$T \vdash (\mathrm{Prb}_T\, \sigma \longrightarrow \mathrm{Prb}_T\, (\mathrm{Prb}_T\, \sigma \longrightarrow \mathbf{0} = \mathbf{S0}))$$

after which another application of formalized modus ponens yields

$$T \vdash (\mathrm{Prb}_T\, \sigma \longrightarrow (\mathrm{Prb}_T\, \mathrm{Prb}_T\, \sigma \longrightarrow \neg\, \mathrm{Cons}\, T)).$$

The formula displayed above (to the right of the turnstile), together with $\mathrm{Prb}_T\, \sigma \longrightarrow \mathrm{Prb}_T\, \mathrm{Prb}_T\, \sigma$ (formalized reflection) imply by sentential logic $\mathrm{Prb}_T\, \sigma \longrightarrow \neg\, \mathrm{Cons}\, T$. ⊣

Gödel's Second Incompleteness Theorem (1931) Assume that T is a sufficiently strong recursively axiomatizable theory. Then $T \vdash \mathrm{Cons}\, T$ if and only if T is *in*consistent.

Proof. If $T \vdash \mathrm{Cons}\, T$ then by Formalized Lemma 37B we have $T \vdash \neg\, \mathrm{Prb}_T\, \sigma$ whence by our choice of σ, we have $T \vdash \sigma$. We conclude from the (unformalized) Lemma 37B that T is inconsistent. ⊣

We can squeeze a bit more out of these ideas. Lemma 37B can be regarded as a special case (where τ is $\mathbf{0} = \mathbf{S0}$) of the following:

Lemma 37C Let T be a recursively axiomatizable theory including A_E, let τ be a sentence, and let σ be obtained from the fixed-point lemma so that

$$A_E \vdash (\sigma \leftrightarrow (\mathrm{Prb}_T\, \sigma \longrightarrow \tau)).$$

If $T \vdash \sigma$, then $T \vdash \tau$.

Proof. We can think of σ as saying, “If I am provable, then τ.” If $T \vdash \sigma$ then by reflection $T \vdash \mathrm{Prb}_T\, \sigma$. By the choice of σ, we have $T \vdash \tau$. ⊣

Actually we are not interested in this lemma, but in its formalization:

Formalized Lemma 37C Assume that T is a sufficiently strong recursively axiomatizable theory. Let τ be a sentence, and let σ be a sentence such that

$$A_E \vdash (\sigma \leftrightarrow (\mathrm{Prb}_T\, \sigma \longrightarrow \tau)).$$

Then $T \vdash (\,\mathrm{Prb}_T\, \sigma \longrightarrow \mathrm{Prb}_T\, \tau)$.

PROOF. We proceed as before. By the choice of σ we get

$$T \vdash (\sigma \longrightarrow (\mathrm{Prb}_T\, \sigma \longrightarrow \tau)).$$

Applying first reflection and then formalized modus ponens to this formula yields

$$T \vdash (\mathrm{Prb}_T\, \sigma \longrightarrow \mathrm{Prb}_T\, (\mathrm{Prb}_T\, \sigma \longrightarrow \tau))$$

after which another application of formalized modus ponens yields

$$T \vdash (\mathrm{Prb}_T\, \sigma \longrightarrow (\mathrm{Prb}_T\, \mathrm{Prb}_T\, \sigma \longrightarrow \mathrm{Prb}_T\, \tau)).$$

The formula displayed above (to the right of the turnstile), together with $\mathrm{Prb}_T\, \sigma \longrightarrow \mathrm{Prb}_T\, \mathrm{Prb}_T\, \sigma$ (formalized reflection) imply by sentential logic $\mathrm{Prb}_T\, \sigma \longrightarrow \mathrm{Prb}_T\, \tau$. ⊣

LÖB'S THEOREM (1955) Assume that T is a sufficiently strong recursively axiomatizable theory. If τ is any sentence for which $T \vdash (\mathrm{Prb}_T\, \tau \longrightarrow \tau)$, then $T \vdash \tau$.

Clearly if $T \vdash \tau$, then $T \vdash (\rho \longrightarrow \tau)$ for any sentence ρ. So the conclusion to Löb's theorem can be stated

$$T \vdash (\mathrm{Prb}_T\, \tau \longrightarrow \tau) \iff T \vdash \tau.$$

PROOF. Given the sentence τ, we construct σ to say, "If I am provable then τ," as above. Suppose that $T \vdash (\mathrm{Prb}_T\, \tau \longrightarrow \tau)$. By Formalized Lemma 37C we have $T \vdash (\mathrm{Prb}_T\, \sigma \longrightarrow \mathrm{Prb}_T\, \tau)$. By our choice of σ, we conclude that $T \vdash \sigma$. So by the (unformalized) Lemma 37C, we have $T \vdash \tau$. ⊣

Löb's theorem was originally devised in order to solve the problem given in Exercise 1. But it implies (and in a sense is equivalent to) Gödel's second incompleteness theorem. Assume that T is a sufficiently strong axiomatizable theory. Applying Löb's theorem and taking τ to be $\mathbf{0} = \mathbf{S0}$, we have

$$T \vdash (\mathrm{Prb}_T\, (\mathbf{0} = \mathbf{S0}) \longrightarrow \mathbf{0} = \mathbf{S0}) \Rightarrow T \vdash \mathbf{0} = \mathbf{S0},$$

that is,

$$T \vdash \mathrm{Cons}\, T \Rightarrow T \text{ is inconsistent.}$$

Thus we obtain a proof of the second incompleteness theorem.

But there is an issue not yet examined: What theories *are* sufficiently strong? Are there any at all (apart from the trivial case of the inconsistent theory)?

Yes, and here are two. The first is called "Peano arithmetic" (PA). Its axioms consist of the A_E axioms, plus all the "induction axioms." These are the universal closures of formulas having the form

$$\varphi(0) \land \forall x(\varphi(x) \longrightarrow \varphi(\mathbf{S}x)) \longrightarrow \forall x\, \varphi(x)$$

for a wff φ. The induction axioms — which state the ordinary principle of mathematical induction — enable us to carry out many arguments about the natural numbers (e.g., the commutative law of addition) within Peano arithmetic. But to be sure that formalized reflection and formalized modus ponens can be derived with Peano arithmetic, one must carry out the details, which we will not go through here.

We know that Peano arithmetic is consistent, because it is true in $\mathfrak{N}$. But by the second incompleteness theorem, PA cannot prove its own consistency. We "know" that PA is consistent by means of an argument we carry out either in informal mathematics, or — if we want — in set theory. So set theory has a higher "consistency strength" than PA: It proves the consistency of PA and PA does not.

A second sufficiently strong theory is axiomatic set theory. Or to be more careful, it is the set of sentences in the language of number theory that are provable in axiomatic set theory. The next subsection deals with this situation. This theory has the advantage that it is quite believable — on an informal level — that formalized reflection and formalized modus ponens are derivable. But what are our grounds for thinking that set theory is consistent? We know that PA is consistent because it is true in the "standard model" $\mathfrak{N}$ of number theory. It is not at all clear that we can meaningfully speak of a "standard model of set theory"!

Applications to Set Theory

We know that in the language of number theory, Cn A_E is incomplete and nonrecursive, as is any compatible recursively axiomatizable theory in the language.

But now suppose we leave arithmetic for a while and look at set theory. Here we have a language (with the parameters $\forall$ and $\in$) and a set of axioms. In all presently accepted cases the set of axioms is recursive. Or more precisely, the set of Gödel numbers of the axioms is recursive. And so the theory (set theory) obtained is recursively enumerable. We claim that this theory, if consistent, is not recursive and hence not complete. We can already sketch the argument in rough form. We can, in a very real sense, embed the language of number theory in set theory. We can then look at that fragment of set theory which deals with the natural numbers and their arithmetic (the shaded area in Fig. 14). That is a theory compatible with A_E. And so it is nonrecursive. Now if set theory were recursive, then its arithmetical part would also be recursive, which it is not. As a bonus, we will come across the second incompleteness theorem for the case of set theory.

Henceforth by set theory (ST) we mean that theory (in the language with equality having the two parameters $\forall$ and $\in$) which is the set of consequences of the reader's favorite set-theoretic axioms. (The standard Zermelo–Fraenkel axioms will do nicely, if the reader has no favorite.

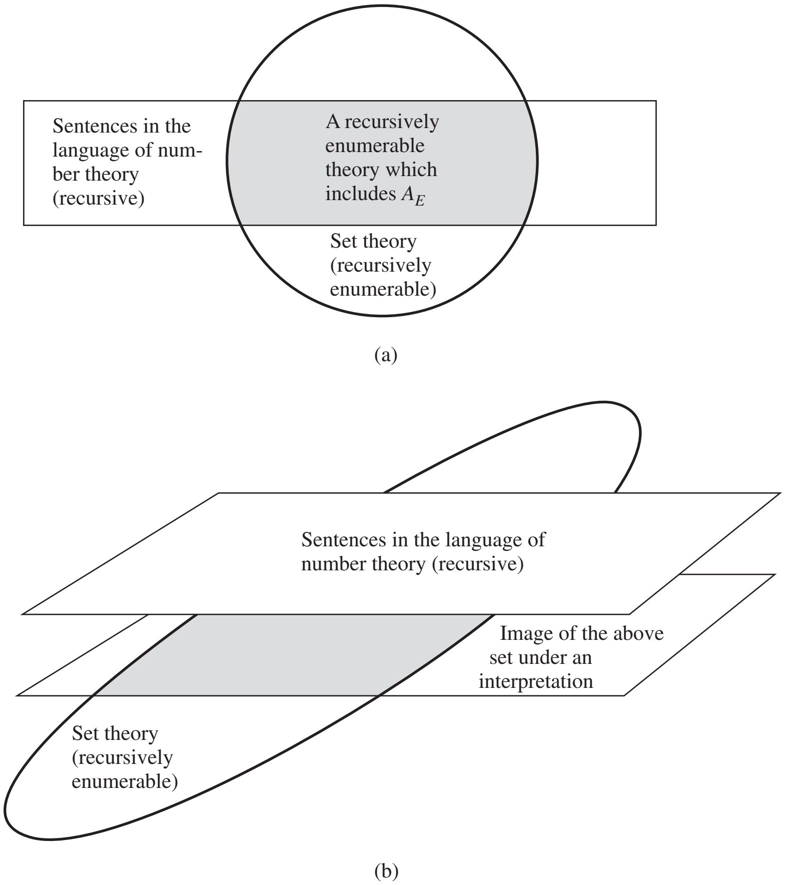

Sentences in the language of number theory (recursive)
A recursively enumerable theory which includes A_E
Set theory (recursively enumerable)
(a)
Sentences in the language of number theory (recursive)
Image of the above set under an interpretation
Set theory (recursively enumerable)
(b)

Figure 14. Set theory and number theory. (a) Flat picture. (b) A more accurate picture.

We ask only that the set of axioms be recursive, and that it be strong enough to yield certain everyday facts about sets.) We need an interpretation π of Cn A_E into ST. (The remainder of this section assumes a familiarity with Section 2.7.) But the existence of such a π is a standard result of set theory, although it is not usually stated in these words. We need formulas of the language of ST that adequately express the concept of being a natural number, being the sum of two given numbers, and so forth. To find these formulas, we turn to the way in which the arithmetic of natural numbers can be "embedded" in set theory. That is, on the one hand, natural numbers such as 2 or 7 do not *appear* to be sets. On the other hand, we can, when we choose, select sets to *represent* numbers. The standard approach is to take 0 to be the set $\varnothing$ and $n + 1$ to be the set $n; n$. This has the fringe benefit that each number is the set of all smaller numbers (e.g., $3 \in 7$). Let ω be the collection of all these sets (these "number-sets"); thus ω is the set representing $\mathbb{N}$.

The formula $\pi_\forall$ is the result of eliminating the defined symbol ω from the formula $\boldsymbol{v}_1 \in \omega$. The formula $\pi_{\mathbf{0}}$ is similarly obtained from the set-theoretic formula $\boldsymbol{v}_1 = \varnothing$, and the formula $\pi_{\mathbf{S}}$ is obtained from $\boldsymbol{v}_2 = \boldsymbol{v}_1 \cup \{\boldsymbol{v}_1\}$. The formula $\pi_{<}$ is simply $\boldsymbol{v}_1 \in \boldsymbol{v}_2$. For π_{+} we use the translation into the language of ST of

For any f, if $f : \omega \times \omega \to \omega$ and for all a and b
in ω we have $f(a, \varnothing) = a$
and $f(a, b \cup \{b\}) = f(a, b) \cup \{f(a, b)\}$,
then $f(\boldsymbol{v}_1, \boldsymbol{v}_2) = \boldsymbol{v}_3$.

(The manner of translation is partially indicated in Chapter 0.) The formulas $\pi_{\cdot}$ and $\pi_{\mathbf{E}}$ are obtained in much the same fashion.

The claim that this π is an interpretation of Cn A_E into ST makes a number (and the number is 17) of demands on ST.

(i) $\exists\, \boldsymbol{v}_1 \pi_\forall$ must be in ST. It is, since we can prove in set theory that ω is nonempty.

(ii) For each of the five function symbols f in the language of A_E, ST must contain a sentence asserting, roughly, that π_f defines a function on the set defined by $\pi_\forall$. (The exact sentence is set forth in the definition of interpretation in Section 2.7.) In the case of $\mathbf{0}$, we have in ST the result that there is a unique empty set and that it belongs to ω. The case for $\mathbf{S}$ is simple, since $\pi_{\mathbf{S}}$ defines a unary operation on the universe of all sets, and ω is closed under this operation. For $+$ we must use the recursion theorem on ω. That is, we can prove in ST (as sketched in Section 1.4) that there is a unique $f : \omega \times \omega \to \omega$ such that $f(a, \varnothing) = a$ and $f(a, b \cup \{b\}) = f(a, b) \cup \{f(a, b)\}$ for a, b in ω. The required property of π_{+} then follows. Similar arguments apply to $\cdot$ and $\mathbf{E}$.

(iii) For each of the 11 sentences σ in A_E, the sentence σ^π must be in ST. For example, in the case of L3, we have in ST the fact that for any m and n in ω, either $m \in n$, $m = n$, or $n \in m$.

Since these demands are finite in number, there is a finite $\Phi \subseteq$ ST such that π is also an interpretation of Cn A_E into Cn Φ.

Theorem 37D (Strong Undecidability of Set Theory) Let T be a theory in the language of set theory such that $T \cup$ ST (or at least $T \cup \Phi$) is consistent. Then $\sharp T$ is not recursive.

Proof. Let Δ be the consistent theory Cn$(T \cup \Phi)$. Let Δ_0 be the corresponding theory $\pi^{-1}[\Delta]$ in the language of number theory. From Section 2.7 we know that Δ_0 is a consistent theory (since Δ is). Also $A_E \subseteq \Delta_0$, since if $\sigma \in A_E$, then $\sigma^\pi \in$ Cn $\Phi \subseteq \Delta$.

Hence by the strong undecidability of Cn A_E (Theorem 35C), $\sharp\Delta_0$ is not recursive.

Now we must derive the nonrecursiveness of T from that of Δ_0. We have

$$\sigma \in \Delta_0 \quad \text{iff} \quad \sigma^\pi \in \Delta$$

and by the lemma below, $\sharp\sigma^\pi$ depends recursively on $\sharp\sigma$. That is, $\sharp\Delta_0 \leq_{\mathrm{m}} \sharp\Delta$. Hence $\sharp\Delta$ cannot be recursive, lest $\sharp\Delta_0$ be. Similarly, we have

$$\tau \in \Delta \quad \text{iff} \quad (\varphi \longrightarrow \tau) \in T,$$

where φ is the conjunction of the members of Φ. Since $\sharp(\varphi \longrightarrow \tau)$ depends recursively on $\sharp\tau$, we have $\sharp\Delta \leq_{\mathrm{m}} \sharp T$ so that $\sharp T$ cannot be recursive lest $\sharp\Delta$ be. ⊣

Lemma 37E There is a recursive function p such that for any formula α of the language of number theory, $p(\sharp\alpha) = \sharp(\alpha^\pi)$.

Proof. In Section 2.7 we gave explicit instructions for constructing α^π. The construction in some cases utilized formulas β^π for formulas β simpler than α. The methods of Sections 3.3 and 3.4 can be applied to the Gödel numbers of these formulas to show that p is recursive. But the details are not particularly attractive, and we omit them here. ⊣

Corollary 37F If set theory is consistent, then it is not complete.

Proof. Set theory has a recursive set of axioms. If complete, the theory is then recursive (by item 21 of Section 3.4). By the foregoing theorem, this cannot happen if ST is consistent. ⊣

Corollary 37G In the language with equality and a two-place predicate symbol, the set of (Gödel numbers of) valid sentences is not recursive.

Partial Proof. In the foregoing theorem take $T = \mathrm{Cn}\varnothing$, the set of valid sentences. The theorem then assures us that $\sharp T$ is nonrecursive, provided that Φ is consistent. We have not given the finite set Φ explicitly. But we assure the reader that Φ can be chosen in such a way as to be provably consistent. ⊣

It should be noted that π is *not* an interpretation of Th $\mathfrak{N}$ into ST (unless ST is inconsistent). For $\pi^{-1}[\mathrm{ST}]$ is a recursively enumerable theory in the language of $\mathfrak{N}$, as a consequence of Lemma 37E. Hence it cannot coincide with Th $\mathfrak{N}$, and it can include the complete theory Th $\mathfrak{N}$ only if it is inconsistent.

Gödel's Second Incompleteness Theorem for Set Theory

We can employ our usual tricks to find a sentence a of number theory which indirectly asserts that its own interpretation σ^π is not a theorem of set theory. For let D be the ternary relation on $\mathbb{N}$ such that

$\langle a, b, c\rangle \in D$ iff a is the Gödel number of a formula α of number theory and c is the Gödel number of a deduction from the axioms of ST of $\alpha(\mathbf{S}^b\mathbf{0})^\pi$.

The relation D is recursive (by the usual arguments); let $\delta(v_1, v_2, v_3)$ represent D in Cn A_E. Let r be the Gödel number of

$$\forall\, v_3 \neg\, \delta(v_1, v_1, v_3)$$

and let σ be

$$\forall\, v_3 \neg\, \delta(\mathbf{S}^r\mathbf{0}, \mathbf{S}^r\mathbf{0}, v_3).$$

Observe that σ *does* indirectly assert that $\sigma^\pi \notin$ ST. We will now prove that the assertion is correct:

Lemma 37H If ST is consistent, then $\sigma^\pi \notin$ ST.

Proof. Suppose to the contrary that σ^π is deducible from the axioms of ST; let k be $\mathcal{G}$ of such a deduction. Then $\langle r, r, k\rangle \in D$.

$$\therefore A_E \vdash \delta(\mathbf{S}^r\mathbf{0}, \mathbf{S}^r\mathbf{0}, \mathbf{S}^k\mathbf{0});$$
$$\therefore A_E \vdash \exists\, v_3\delta(\mathbf{S}^r\mathbf{0}, \mathbf{S}^r\mathbf{0}, v_3);$$

i.e.,

$$A_E \vdash \neg\,\sigma.$$

Applying our interpretation π, we conclude that $\neg\,\sigma^\pi$ is in ST, whence ST is inconsistent. Thus

$$\text{ST is consistent} \Rightarrow \sigma^\pi \notin \text{ST}. \qquad \dashv$$

Now the above proof, like all those in this book, is carried out in informal mathematics. But all of our work in the book could have been carried out within ST. Indeed it is common knowledge that essentially all work in mathematics can be carried out in ST. Imagine actually doing so. Then instead of a proof of an English sentence, "ST is consistent $\Rightarrow \sigma^\pi \notin$ ST," we have a deduction from the axioms of ST of a certain sentence in the formal language of set theory:

$$(\text{Cons(ST)} \rightarrow \Box).$$

Here Cons(ST) is the result of translating (in a nice way) "ST is consistent" into the language of set theory. Similarly, $\Box$ is the result of translating "$\sigma^\pi \notin$ ST." But we already *have* a sentence in the language of set theory asserting that $\sigma^\pi \notin \mathrm{ST}$. It is σ^π. This strongly suggests that $\Box$ is (or is provably equivalent in ST to) σ^π, from which we get

$$(\mathrm{Cons(ST)} \longrightarrow \sigma^\pi)$$

as a theorem of ST.

Now this *can* actually be carried out in such a way as to have $\Box$ be σ^π. We have given above an argument, which we hope will convince the reader that this is at least probable. And from it we now have the result:

Gödel's Second Incompleteness Theorem for Set Theory The sentence Cons(ST) is not a theorem of ST, unless ST is inconsistent.

Proof. By the above (plausibility) argument

$$(\mathrm{Cons(ST)} \longrightarrow \sigma^\pi)$$

is a theorem of ST. So if Cons(ST) is also a theorem of ST, then σ^π is, too. But by Lemma 37H, if $\sigma^\pi \in \mathrm{ST}$, then ST is inconsistent. ⊣

Of course if ST is inconsistent, then every sentence is a theorem, including Cons(ST). Because of this, a proof of Cons(ST) within ST would not convince people that ST was consistent. (And by Gödel's second theorem, it would convince them of the opposite.) But prior to Gödel's work it was possible to hope that Cons(ST) might be provable from assumptions weaker than the axioms of set theory, ideally assumptions already known to be consistent. But we now see that Cons(ST) is not in any subtheory of ST, unless of course ST is inconsistent.

We are left with the conclusion that any recursively axiomatizable theory of sets (provided it meets the desirable conditions of being consistent and strong enough to prove everyday facts) is an incomplete theory. This raises a challenge: to find additional axioms to add to the theory. On the one hand, we want the additional axioms to strengthen the theory in useful ways. On the other hand, we want the additional axioms to reflect accurately our informal ideas about what sets really are and how they really behave.

Exercises

1. Let σ be a sentence such that

$$A_E \vdash (\sigma \leftrightarrow \mathrm{Prb}_{A_E}\sigma).$$

(Thus σ says "I am provable," in contrast to the sentence "I am unprovable" that has been found to have such interesting properties.) Does $A_E \vdash \sigma$?

2. Let T be a theory in a recursively numbered language, and assume that there is an interpretation of $\mathrm{Cn}\, A_E$ into T. Show that T is strongly undecidable; i.e., whenever T' is a theory in the language for which $T \cup T'$ is consistent, then $\sharp T'$ is not recursive.

SECTION 3.8

Representing Exponentiation[1]

In Sections 3.1 and 3.2 we studied the theory of certain reducts of $\mathfrak{N}$ and found them to be decidable. Then in Section 3.3 we added *both* multiplication and exponentiation. The resulting theory was found (in Section 3.5) to be undecidable. Actually it would have been enough to add only multiplication (and forego exponentiation); we would still have undecidability.

Let $\mathfrak{N}_M$ be the reduct of $\mathfrak{N}$ obtained by dropping exponentiation:

$$\mathfrak{N}_M = (\mathbb{N}; 0, S, <, +, \cdot).$$

Thus the symbol $\mathbf{E}$ does not appear in the language of $\mathfrak{N}_M$. Let A_M be the set obtained from A_E by dropping E1 and E2. The purpose of this section is to show that all the theorems of Sections 3.3–3.5 continue to hold when "A_E" and "$\mathfrak{N}$" are replaced by "A_M" and "$\mathfrak{N}_M$." The key fact needed to establish this claim is that exponentiation is representable in $\mathrm{Cn}\, A_M$. That is, there is a formula ε in the language of $\mathfrak{N}_M$ such that for any a and b,

$$A_M \vdash \forall z[\varepsilon(\mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}, z) \leftrightarrow z = \mathbf{S}^{(a^b)}\mathbf{0}].$$

Thus $\varepsilon(x, y, z)$ can be used to simulate the formula $x\mathbf{E}y = z$ without actual use of the symbol $\mathbf{E}$.

If we look to see what relations and functions are representable in $\mathrm{Cn}\, A_M$, we find at first that everything (except for exponentiation itself) that was shown to be representable in $\mathrm{Cn}\, A_E$ is (by the same proof) representable in $\mathrm{Cn}\, A_M$. Until, that is, we reach item 7 in the catalog listing of Section 3.3. To go further, we must show that exponentiation itself is representable in $\mathrm{Cn}\, A_M$.

We know that exponentiation can be characterized by the recursion equations

$$a^0 = 1,$$
$$a^{b+1} = a^b \cdot a.$$

[1] This section may be omitted without loss of continuity.

From what we know about primitive recursion (catalog item 13 in Section 3.3 plus Exercise 8 there), we might think of defining

$$\begin{aligned} E^*(a,b) = {} & \text{the least } s \text{ such that } [(s)_0 = 1 \text{ and} \\ & \text{for all } i < b,\ (s)_{i+1} = (s)_i \cdot a]. \end{aligned}$$

For then $a^b = (E^*(a,b))_b$. This fails to yield a proof of representability, because we do not yet know that the decomposition function $(a)_b$ is representable in Cn A_M. But we do not really need that particular decomposition function (which corresponded to a particular way of encoding sequences). All we need is *some* function δ that acts like a decomposition function; the properties we need are summarized in the following lemma.

Lemma 38A There is a function δ representable in Cn A_M such that for every $n, a_0, \ldots, a_n$, there is an s for which $\delta(s,i) = a_i$ for all $i \le n$.

Once the lemma has been established, we can define

$$\begin{aligned} E^{**}(a,b) = {} & \text{the least } s \text{ such that } [\delta(s,0) = 1 \text{ and} \\ & \text{for all } i < b,\ \delta(s,i+1) = \delta(s,i) \cdot a]. \end{aligned}$$

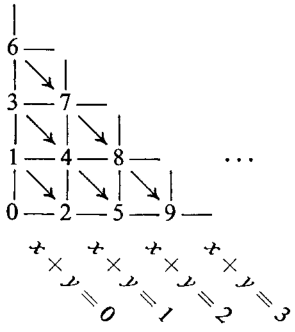
The lemma assures us that such an s exists. E^{**} is then representable in Cn A_M, as is exponentiation, since

$$a^b = \delta(E^{**}(a,b), b).$$

A function δ that establishes the lemma will be provided by some facts of number theory.

A Pairing Function

As a first step toward proving the foregoing lemma, we will construct functions for encoding and decoding pairs of numbers. It is well known that there exist functions mapping $\mathbb{N} \times \mathbb{N}$ one-to-one onto $\mathbb{N}$. In particular, the function J does this, where in the diagram shown, $J(a,b)$ has been written at the point with coordinates $\langle a, b \rangle$.



For example, $J(2,1) = 8$ and $J(0,2) = 3$. To obtain an equation for $J(a,b)$, we note that along the line $x + y = n$ there are $n + 1$ points (with coordinates in $\mathbb{N}$). Thus

$$\begin{aligned} J(a,b) &= \text{the number of points in the plane to which } J \text{ assigns smaller values} \\ &= [\text{the number of points on lines } x + y = n \text{ for } n = 0, 1, \ldots, (a + b - 1)] + [\text{the number of points on the line } x + y = a + b \text{ for which } x < a] \\ &= [1 + 2 + \cdots + (a + b)] + a \\ &= \tfrac{1}{2}(a + b)(a + b + 1) + a \\ &= \tfrac{1}{2}[(a + b)^2 + 3a + b]. \end{aligned}$$

Let K and L be the corresponding projection functions onto the axes, i.e., the unique functions such that

$$K(J(a,b)) = a, \quad L(J(a,b)) = b.$$

For example, $K(7) = 1$, the x-coordinate of the point $\langle 1, 2\rangle$ in the plane to which J assigned the number 7. Similarly, $L(7) = 2$, the y-coordinate of that point.

We claim that J, K, and L are representable in Cn A_M. The function

$$H(a) = \text{the least } b \text{ such that } a \leq 2b$$

has the property that $H(a) = \frac{1}{2}a$ for even a. Then we can write

$$\begin{aligned} J(a,b) &= H((a + b) \cdot (a + b + 1)) + a, \\ K(p) &= \text{the least } a \text{ such that [for some } b \leq p,\ J(a,b) = p], \\ L(p) &= \text{the least } b \text{ such that [for some } a \leq p,\ J(a,b) = p]. \end{aligned}$$

From the form of the four preceding equations we conclude that H, J, K, and L are representable in Cn A_M.

The Gödel β-function

Let β be the function defined as follows:

$$\begin{aligned} \beta(c,d,i) &= \text{the remainder in } c \div [1 + (i + 1) \cdot d] \\ &= \text{the least } r \text{ such that for some } q \leq c, \\ &\quad\ c = q \cdot [1 + (i + 1) \cdot d] + r. \end{aligned}$$

This unlikely-looking function produces a satisfactory decomposition function for Lemma 38A. Let

$$\delta(s,i) = \beta(K(s), L(s), i).$$

It is clear that δ is representable in Cn A_M. What is not so obvious is that it meets the conditions of Lemma 38A. We want to show:

$$\begin{array}{l} \text{For any } n \text{ and any } a_0, \ldots, a_n, \text{ there are numbers} \\ c \text{ and } d \text{ such that } \beta(c,d,i) = a_i \text{ for all } i \leq n. \end{array} \qquad (*)$$

For then it follows that $\delta(J(c,d), i) = \beta(c,d,i) = a_i$ for $i \leq n$.

Now $(*)$ is a statement of number theory, not logic. The proof of $(*)$ is based on the Chinese remainder theorem. Numbers $d_0, \ldots, d_n$ are said to be *relatively prime in pairs* iff no prime divides both d_i and d_j for $i \neq j$.

Chinese Remainder Theorem Let $d_0, \ldots, d_n$, be relatively prime in pairs; let $a_0, \ldots, a_n$ be natural numbers with each $a_i < d_i$. Then we can find a number c such that for all $i \leq n$,

$$a_i = \text{the remainder in } c \div d_i.$$

Proof. Let $p = \Pi_{i \leq n} d_i$, and for any c let $F(c)$ be the $(n+1)$-tuple of remainders when c is divided by $d_0, \ldots, d_n$. Notice that there are p possible values for this $(n+1)$-tuple.

We claim that F is one-to-one on $\{k \mid 0 \leq k < p\}$. For suppose that $F(c_1) = F(c_2)$. Then each d_i divides $|c_1 - c_2|$. Since the d_i's are relatively prime, p must divide $|c_1 - c_2|$. For c_1, c_2 less than p, this implies that $c_1 = c_2$.

Hence the restriction of F to $\{k \mid 0 \leq k < p\}$ takes on all p possible values. In particular, it assumes (at some point c) the value $\langle a_0, \ldots, a_n \rangle$. And that is the c we want. ⊣

Lemma 38B For any $s \geq 0$, the $s+1$ numbers

$$1 + 1 \cdot s!, \quad 1 + 2 \cdot s!, \quad \ldots, \quad 1 + (s+1) \cdot s!$$

are relatively prime in pairs.

Proof. All these numbers have the property that any prime factor q cannot divide $s!$, whence $q > s$. If the prime q divides both $1 + j \cdot s!$ and $1 + k \cdot s!$, then it divides their difference, $|j-k| \cdot s!$. Since q does not divide $s!$, it divides $|j-k|$. But $|j-k| \leq s < q$. This is possible only if $|j-k| = 0$. ⊣

Proof of $(*)$. Assume we are given $a_0, \ldots, a_n$; we need numbers c and d such that the remainder when c is divided by $1 + (i+1) \cdot d$ is a_i, for $i \leq n$.

Let s be the largest of $\{n, a_0, \ldots, a_n\}$ and let $d = s!$. Then by Lemma 38B, the numbers $1 + (i+1) \cdot d$ are relatively prime in pairs for $i \leq n$. So by the Chinese remainder theorem there is a c such that the remainder in $c \div [1 + (i+1) \cdot d]$ is a_i for $i \leq n$. ⊣

This completes the proof of Lemma 38A. And by the argument that followed that lemma, we can conclude:

Theorem 38C Exponentiation is representable in Cn A_M.

Armed with this theorem, we can now return to catalog item 7 of Section 3.3. The proof given there now establishes that the function in question (whose value at n is p_n) is representable in Cn A_M. For it was

TABLE X

Structure	Theory	Models of the theory	Definable sets	Comments
$(\mathbb{N})$	Decidable. Not finitely axiomatizable. Admits elimination of quantifiers.	Any infinite set.	$\varnothing$ and $\mathbb{N}$. $\{0\}$ is not definable.	
$(\mathbb{N}; 0)$	As above.	Any infinite set with distinguished element.	$\varnothing$, $\{0\}$, $\mathbb{N} - \{0\}$, $\mathbb{N}$. S is not definable.	
$(\mathbb{N}; 0, S)$	As above.	Standard part plus any number of Z-chains.	Finite and cofinite sets. $<$ is not definable.	$\{0\}$ is definable in $(\mathbb{N}; S)$.
$(\mathbb{N}; 0, S, <)$	Decidable. Finitely axiomatizable. Admits elimination of quantifiers.	As above, with any ordering of the Z-chains.	Finite and cofinite sets. $+$ is not definable.	$\{0\}$ and S are definable in $(\mathbb{N}; <)$.
$(\mathbb{N}; 0, S, <, +)$	Decidable (Presburger).	The Z-chains are densely ordered without endpoints. Also there is a suitable addition operation.	Eventually periodic sets. $\cdot$ is not definable.	$\{0\}$, S, and $<$ are definable in $(\mathbb{N}; +)$.
$(\mathbb{N}; 0, S, <, +, \cdot)$	Not arithmetical. $\therefore$ not recursively axiomatizable.	As above, but with a suitable multiplication operation.	All arithmetical relations are definable.	The arithmetical relations are definable in $(\mathbb{N}; S, \cdot)$, $(\mathbb{N}; +, \cdot)$, and $(\mathbb{N}; <, D)$, where $D(x, y) = (x)_y$.

formed by allowable methods from relations and functions (including exponentiation) known to be representable in Cn A_M.

The same phenomenon persists throughout Sections 3.3 and 3.4. The representability proofs given there now establish representability in Cn A_M. Thus any recursive relation is representable in Cn A_M, and if the relation happens to be a function, then it is functionally representable. The proofs given in Section 3.5 then apply to $\mathfrak{N}_M$ and A_M as well as to $\mathfrak{N}$ and A_E. In particular, we have the strong undecidability of Cn A_M: Any theory T in the language of $\mathfrak{N}_M$ for which $T \cup A_M$ is consistent cannot be recursive.

Notice that any relation definable in $\mathfrak{N}$ (i.e., any arithmetical relation) is also definable in $\mathfrak{N}_M$. For exponentiation, being representable in a subtheory of Th $\mathfrak{N}_M$, is *a fortiori* definable in $\mathfrak{N}_M$. By the new version of Tarski's theorem, $\sharp$Th$\mathfrak{N}_M$ is not definable in $\mathfrak{N}_M$, and consequently $\sharp$Th$\mathfrak{N}_M$ cannot be arithmetical.

In the terminology of Section 2.7, we can say that there is a faithful interpretation of Th $\mathfrak{N}$ into Th $\mathfrak{N}_M$. It equals the identity interpretation on all parameters except **E**, and to **E** it assigns a formula defining exponentiation in $\mathfrak{N}_M$.

In Table X we summarize some of the results of Chapter 3 on number theory and its reducts.

Exercises

1. Let $D(a, b) = (a)_b$. Show that any arithmetical relation is definable in the structure $(\mathbb{N}; <, D)$. *Remark*: One may well ask why Th $\mathfrak{N}_A$, arithmetic with addition, is decidable (as shown in Section 3.2), while Th $\mathfrak{N}_M$, the theory of arithmetic with addition and multiplication, is undecidable. One answer is that, as this section shows, multiplication lets us do a certain amount of sequence coding and decoding. The point of this exercise is to show that once we have the decoding function D and ordering, we have the full complexity of arithmetic with addition, multiplication, and exponentiation.
2. Show that the addition relation $\{\langle a, b, c\rangle \mid a + b = c\}$ is definable in the structure $(\mathbb{N}; S, \cdot)$. *Suggestion*: Under what conditions does the equation $S(ac) \cdot S(bc) = S(c \cdot c \cdot S(ab))$ hold?
3. (a) Show that Th$(\mathbb{Z}; +, \cdot)$ is strongly undecidable. (See Exercise 2 of Section 3.7.)

   (b) (This part assumes a background in algebra.) Show that the theory of rings is undecidable and that the theory of commutative rings is undecidable.