Notes on Mathematical Logic

David W. Kueker

University of Maryland, College Park
E-mail address: dwk@math.umd.edu
URL: http://www-users.math.umd.edu/~dwk/

Contents

Part 3

Incompleteness and Undecidability

CHAPTER 7

An Informal Introduction to Decidability Problems

0. Introduction

The material in this part grows out of Gödel's famous Incompleteness Theorem of 1931. It states that the usual Peano axioms for $+$, $\cdot$ on the natural numbers are *not* sufficient to derive all true first-order sentences of arithmetic on the natural numbers. More generally it states that there is no way to "effectively" give a complete set of axioms for arithmetic on the natural numbers. As a consequence there is no way to "effectively" decide whether or not a first-order sentence is true on the natural numbers.

In this chapter we present an informal notion of "effectiveness" and use it in discussing problems concerning decidability. We then derive Gödel's Incompleteness Theorem *assuming* a major lemma on definability of decidable relations. An essential tool is the device of "Gödel-numbering," which enables us to treat formulas, derivations, etc., as natural numbers.

In the next chapter we give a formal treatment of effectiveness in terms of recursive functions. This enables us to *prove* the lemma mentioned above. We can then also derive various improvements, extensions, and generalizations of Gödel's original result, due to Rosser, Church, Tarski, et al.

1. Effective Procedures and Decidability

Recall from sentential logic that the method of truth tables gave an *effective procedure* to decide whether an arbitrary sentence of sentential logic is a tautology. In Section 2.2 we raised the natural question of whether there is such an effective procedure for deciding logical validity in first-order logic. Note that neither the definition of validity nor the Completeness Theorem provides such a procedure.

More generally, given any theory T of a language $\mathcal{L}$ one can ask whether or not there is an effective procedure which decides, for each sentence σ of $\mathcal{L}$, whether or not $T \models \sigma$ (equivalently, whether $\sigma \in T$). To make this question precise we need to define what we mean by an "effective procedure."

First of all, note that an "effective procedure" is supposed to define a *function* on some given domain D—in the example above of sentential logic, the domain is the set of all sentences of sentential logic and the function takes the values "yes" or "no" according to whehter the sentence is a tautology or not. Secondly, an "effective procedure" should *not* just define the values of a function but should give a means to *calculate* (at least in theory) the value of the function for any element in the domain. Thus, in the example, one could (in a finite number of steps) actually write down and check the truth table of any given sentence of sentential logic. On the other hand, one cannot check each of the infinitely many structures for a first-order language $\mathcal{L}$ to see whether or not a given sentence σ of $\mathcal{L}$ is true on the structure.

We are thus led to the following, still informal, definition:

Definition 1.1. Let f be a function defined on D. An *effective procedure* to calculate f is a finite list of instructions (in, say, English) such that given any $d \in D$ they can be applied to d and executed so that after finitely many discrete steps (each performable in a finite amount of time) the process halts and produces the value $f(d)$.

Finally, a function f is *(effectively) computable* if and only if *there is* some effective procedure to calculate it. Note that the same function can be defined in many different ways, not all of which give an effective procedure to calculate it.

Let us look at several examples:

(1) $D = \omega; f(n) = n+1$. Certainly this f must be effectively computable, but coming up with an effective procedure to calculate it requires us to decide what exactly the elements of ω *are*, and what exactly "adding one" does to them. The set-theoretic definitions just lead us into more difficulties (what *is* a set?), so we take the more formalistic point of view that the number n is a 0 followed by n 1s. Thus the effective procedure is to place one more 1 at the end of the sequence. Even this assumes some material about finite sequences, but this seems to be an unavoidable starting point. From now on we will not worry about what the elements of ω "really are," and also we will freely use elementary manipulations with finite sequences.

(2) $D = \omega \times \omega; f(\langle m, n\rangle) = m + n$. The effective procedure here says to start with m and add 1 n times.

(3) $D = \omega \times \omega; f(\langle m, n\rangle) = 0$ if $m \leq n; f(\langle m, n\rangle) = 1$ otherwise. We leave it to the reader to come up with an effective procedure.

(4) $D = \omega \times \omega; f(\langle m, n\rangle) = 0$ if $m \mid n; f(\langle m, n\rangle) = 1$ otherwise. Here too we leave the formulation of an effective procedure to the reader.

(5) $D = \omega; f(n) = 0$ if n is prime; $f(n) = 1$ otherwise. In outline, the effective procedure here is to check each $m < n, m \neq 1$, to see whether or not $m \mid n$. If such an m is found, $f(n) = 0$; otherwise $f(n) = 1$.

(6) $D = \omega; f(n) = 0$ if $n \leq 2$; $f(n) = 1$ otherwise. Certainly this f is effectively computable using the result in example (3).

(7) $D = \omega; f(n) = 0$ if and only if there are positive integers a, b, c such that $a^n + b^n = c^n$; $f(n) = 1$ otherwise. It is not obvious whether this function is effectively computable—certainly this definition does not give an effective procedure to calculate it. But this f *may* still be effectively computable—in fact it *may* be the same function as the function in example (6).

We will continue with some examples involving expressions of a first-order language $\mathcal{L}$. We assume we have come to some agreement as to what the symbols of $\mathcal{L}$ really "are" (like v_ns). More importantly, we assume that for each symbol of $\mathcal{L}$ we can effectively decide what sort of a symbol it is, as "a function symbol of 2 places," etc. Under these basic assumptions, we can consider the following examples:

(8) $D =$ all expressions of $\mathcal{L}$; $f(d) = 0$ if d is a term; $f(d) = 1$ otherwise. In outline, an effective procedure to compute f involves trying to "read" d as a term—since there are only finitely many ways this could happen, you can check each one in turn and see if it does indeed work for d.

(9) D = all expressions of $\mathcal{L}$; $f(d) = 0$ if d is a formula of $\mathcal{L}$; $f(d) = 1$ otherwise. This function is similarly seen to be effectively computable.

(10) D = all formulas of $\mathcal{L}$; $f(d) = 0$ if d is a logical axiom; $f(d) = 1$ otherwise. The reader can verify that this is also effectively computable.

(11) D = all finite sequences of formulas of $\mathcal{L}$; $f(d) = 0$ if d is a formal deduction (from Λ); $f(d) = 1$ otherwise. Here the definition of formal deduction gives us an effective procedure to compute f, given the result of example (10).

(12) $D = \mathrm{Fm}_{\mathcal{L}}; f(\phi) = 0$ if $\vdash \phi; f(\phi) = 1$ otherwise. There is no obvious effective procedure to compute f, since there are infinitely many formal deductions which would have to be checked.

We emphasize that an effective procedure does not have to be *practical* in all cases—the finite number of steps involved might be too large to be actually performable. For example, one could actually be given a sentence of sentential logic involving 100 different sentence letters; the truth table for this sentence would have 2^{100} lines—which is way beyond our capacity to check. Thus, once one knows that a function is effectively computable there still remains the question of how *efficient* a procedure there is to compute it.

Concepts of computability apply to sets by considering characteristic functions.

DEFINITION 1.2. Given D and $X \subseteq D$, the *characteristic function* of X is the function $K_X \colon D \to \{0, 1\}$ defined by

$$K_X(d) = \begin{cases} 0 & \text{if } d \in X \\ 1 & \text{if } d \in D - X \end{cases}$$

Actually, K_X depends also on D, which is usually clear from the context.

DEFINITION 1.3. Fix D, and let $X \subseteq D$.

(a) X is *decidable* (as a subset of D) if and only if K_X is effectively compuatable.

(b) $X \neq \varnothing$ is *listable* if and only if there is an effectively computable function $f \colon \omega \to X$ which maps *onto* X, i.e., such that $X = \{f(n) \colon n \in \omega\}$.

The preceding examples then show, for example, that the set of prime numbers is decidable (as a subset of ω), the set of tautologies is decidable (as a subset of the set of all sentences of sentential logic), and the set of formulas of $\mathcal{L}$ is decidable (as a subset of the set of all expressions of $\mathcal{L}$).

One can show (see the next section) that $\{n \in \omega \colon a^n + b^n = c^n$ for some positive $a, b, c \in \omega\}$ is listable, but this does not tell us whether or not this set is decidable. The same holds for $\{\phi \in \mathrm{Fm}_{\mathcal{L}} \colon \vdash \phi\}$.

We will normally suppress reference to the domain D when talking about the decidability of X—usually it will be clear what D is, and some ambiguity will be harmless due to the following easy lemma.

LEMMA 1.1. *Let $X \subseteq D_1 \subseteq D_2$. Assume that D_1 is decidable (as a subset of D_2). Then X is decidable as a subset of D_1 if and only if X is decidable as a subset of D_2.*

Thus, we can say simply that "the set of logical axioms of $\mathcal{L}$ is decidable," and the meaning is independent of whether we mean "as a subset of $\mathrm{Fm}_{\mathcal{L}}$" or "as a subset of the set of expressions of $\mathcal{L}$."

Roughly speaking, decidable implies listable. More precisely, we have:

PROPOSITION 1.2. *Assume D is listable and $X \subseteq D$ is decidable (as a subset of D). Then X is listable.*

PROOF. The idea is to list the elements of D but only retain the ones belonging to X. We may suppose $X \neq \varnothing$. Let $f\colon \omega \to D$ be effectively computable and list D. We define $g\colon \omega \to X$ to list X as follows: Let n_0 be the first $n \in \omega$ such that $K_X(f(n)) = 0$ (i.e., such that $f(n) \in X$); define $g(0) = f(n_0)$. Next define

$$g(1) = \begin{cases} f(n_0+1) & \text{if } f(n_0+1) \in X \\ f(n_0) & \text{otherwise} \end{cases}$$

Similarly for $g(2), g(3), \dots$. Then g is effectively computable and lists X. □

One other general fact of immense importance should be mentioned here.

PROPOSITION 1.3. *Fix D. There are only countably many effectively computable functions whose domain is D. Hence D has only countably many decidable (or listable) subsets.*

PROOF. There are just countably many finite sequences which could possibly be instructions in English for effectively computing a function of domain D (or ω). □

Thus for any infinite D, "most" subsets of D are not decidable, or even listable.

In our new terminology, the question we began with is whether or not $\mathrm{Cn}_{\mathcal{L}}(\varnothing) = \{\sigma \in \mathrm{Sn}_{\mathcal{L}}\colon \models \sigma\}$ is decidable. More generally, for a theory T of $\mathcal{L}$, is $T = \mathrm{Cn}_{\mathcal{L}}(T)$ decidable? Since there are uncountably many different theories T (even of the language $\mathcal{L}$ of pure identity theory), we know that "most" theories T are not decidable.

The question, then, is which are decidable (particularly among specific, mathematically natural, theories)?

The question answered by Gödel's Incompleteness Theorem is not quite of this nature, but it can be precisely formulated in terms of decidability.

The question Gödel answered is the following: is there a decidable set Σ of axioms for $T = Th((\omega, +, \cdot, <, 0, s))$? Of course, if T were decidable we could take $\Sigma = T$. Gödel showed that T has no decidable set of axioms, in particular, then, that T is not decidable. (*Remember*, to say a set of sentences is decidable means *as a subset of* $\mathrm{Sn}_{\mathcal{L}}$, or $\mathrm{Fm}_{\mathcal{L}}$, or expressions of $\mathcal{L}$.)

2. Gödel Numbers

The decidable and listable sets in which we are interested may be sets of finite sequences of natural numbers, or sets of expressions of $\mathcal{L}$, or sets of finite sequences of expressions of $\mathcal{L}$, etc. It is important to know that we can replace any such set by a set of natural numbers, and that this replacement is "effective." Thus, our domain D could always be taken to be ω, in contexts of computability.

DEFINITION 2.1. Let $n_0, \dots, n_k \in \omega$. The *sequence number* of the sequence $(n_0, \dots, n_k)$ is the number $\langle n_0, \dots, n_k\rangle = 2^{n_0+1} \cdot 3^{n_1+1} \cdot 5^{n_2+1} \cdot \ \cdots \ \cdot p_k^{n_k+1}$, where $p_0 = 2$ and $p_i =$ the i^{th} odd prime ($i > 0$).

By the uniqueness of prime power decomposition we see that if $\langle n_0, \dots, n_k\rangle = \langle n'_0, \dots, n'_{k'}\rangle$ then $k = k'$ and $n_i = n'_i$ for all $i = 0, \dots, k$, hence $(n_0, \dots, n_k) =$

$(n'_0, \ldots, n'_k)$. Furthermore the function taking finite sequences $(n_0, \ldots, n_k)$ of natural numbers to their sequence numbers is effectively computable, as is its "inverse," the decoding function defined as:

$$d(m) = \begin{cases} (m_0, \ldots, m_k) & \text{if } m = \langle m_0, \ldots, m_k \rangle \\ 0 & \text{otherwise} \end{cases}$$

From now on, $\langle n_0, \ldots, n_k \rangle$ is always the sequence *number*, while $(n_0, \ldots, n_k)$ is the sequence. We will also occasionally use the *empty sequence* $(\,)$, whose length is 0 and whose sequence number is $\langle\,\rangle = 1$.

We see that the set of all finite sequences of natural numbers is (effectively) listable, and every decidable set of finite sequences of natural numbers is listable.

Most importantly, as Gödel was the first to appreciate, if you assign natural numbers to the symbols of a language $\mathcal{L}$ then all expressions of $\mathcal{L}$ (= finite sequences of symbols of $\mathcal{L}$) are assigned natural numbers via sequence numbers. Further all finite sequences of expressions of $\mathcal{L}$ (such as formal deductions) are assigned numbers. Thus, properties of formulas, etc., can be (effectively) translated into properties of natural numbers. This process is the so-called "arithmetization of syntax."

The original assignment of numbers to symbols must be effective, which may not be possible in some languages with infinitely many non-logical symbols.

To begin with, we assume $\mathcal{L}$ is a first-order language with just finitely many non-logical symbols $s_1, \ldots, s_n$. We define the Gödel-numbering function g mapping the symbols of $\mathcal{L}$ one-to-one into ω as follows:

$$\begin{aligned}
&g(v_n) = 2n \text{ for all } n \in \omega \\
&g(\neg) = 1 \\
&g(\to) = 3 \\
&g(\forall) = 5 \\
&g(\,(\,) = 7 \\
&g(\,)\,) = 9 \\
&g(\equiv) = 11 \\
&g(s_1) = 13 \\
&\vdots \\
&g(s_n) = 11 + 2n
\end{aligned}$$

(Obviously, this depends on fixing the order of the non-logical symbols, but precisely how they are ordered is not significant.)

Given an expression $\epsilon_0 \ldots \epsilon_n$ of $\mathcal{L}$ (thus each ϵ_i is a symbol of $\mathcal{L}$) we define the *Gödel number* of $\epsilon_0 \ldots \epsilon_n$ as $\ulcorner \epsilon_0 \ldots \epsilon_n \urcorner = \langle g(\epsilon_0), \ldots, g(\epsilon_n) \rangle$. If $(\alpha_0, \alpha_1 \ldots, \alpha_n)$ is a sequence of expressions of $\mathcal{L}$ then the Gödel number of the sequence is $\ulcorner \alpha_0, \ldots, \alpha_n \urcorner = \langle \ulcorner \alpha_0 \urcorner, \ldots, \ulcorner \alpha_n \urcorner \rangle$.

This assignment of Gödel numbers to expressions and to sequences of expressions is effective; further any natural number is the Gödel number of at most one expression and at most one sequence of expressions; finally the "decoding" functions are also effectively computable.

For example, $\ulcorner \equiv v_0 v_1 \urcorner = 2^{12} \cdot 3 \cdot 5^3$, and $\ulcorner (\equiv v_0 v_1 \to \equiv v_1 v_0) \urcorner = 2^8 \cdot 3^{12} \cdot 5 \cdot 7^3 \cdot 11^4 \cdot 13^{12} \cdot 17^3 \cdot 19 \cdot 23^{10}$ and $\ulcorner \equiv v_0 v_1, (\equiv v_0 v_1 \to \equiv v_1 v_0) \urcorner = 2^{\ulcorner \equiv v_0 v_1 \urcorner + 1} \cdot 3^{\ulcorner (\equiv v_0 v_1 \to \equiv v_1 v_0) \urcorner + 1}$.

Some points to notice:

(a) The same number may be both the Gödel number of an expression *and* of a sequence of expressions (and also the number assigned to some symbol). For example, $n_0 = 2^{55} \cdot 3^{25} = \ulcorner v_{27}v_{12} \urcorner$, but also $55 = 2 \cdot 3^3 + 1 = \ulcorner v_0 v_1 \urcorner + 1$ and $25 = 2^3 \cdot 3 + 1 = \ulcorner v_1 v_0 \urcorner$, so $n_0 = \ulcorner v_0 v_1, v_1 v_0 \urcorner$; further $n_0 = 2(2^{54} \cdot 3^{25} = g(v_{2^{54} \cdot 3^{25}})$. This ambiguity is harmless.

(b) One must be careful to distinguish between a symbol and the sequence of length 1 consisting only of that symbol. For example, v_0 is *not* the same as the one-place sequence which, unfortunately, we also write as v_0—writing (v_0) would confuse it with the three-place sequence $(, v_0,)$. Our notation for Gödel-numbering is chosen to distinguish these two uses, however. Thus $g(v_0) = 0$, since g is just defined on symbols, and $\ulcorner v_0 \urcorner = \langle g(v_0) \rangle = 2$, since $\ulcorner\ \urcorner$ is defined just on sequences. In particular since terms are always *sequences*, the term v_0 is different from the symbol v_0.

Similarly, the expression α must be distinguished from the sequence consisting just of α. In particular, since formal deductions are *sequences* of formulas, the formal deduction consisting just of ϕ where ϕ is a logical axiom is different from the formula ϕ, and they will have different Gödel numbers.

One can now see that the sets $\{\ulcorner t \urcorner \colon t \in Tm_{\mathcal{L}}\}$, $\{\ulcorner \phi \urcorner \colon \phi \in \mathrm{Fm}_{\mathcal{L}}\}$, $\{\ulcorner \sigma \urcorner \colon \sigma \in \mathrm{Sn}_{\mathcal{L}}\}$, $\{\ulcorner \phi_0, \ldots, \phi_n \urcorner \colon \phi_0, \ldots, \phi_n \in \mathrm{Fm}_{\mathcal{L}}\}$, $\{\ulcorner \phi_0, \ldots, \phi_n \urcorner \colon\ \phi_0, \ldots, \phi_n$ is a deduction from $\varnothing\}$ are all decidable subsets of ω, and hence are also listable.

More generally, we see that a set X of expressions is decidable if and only if $\{\ulcorner \alpha \urcorner \colon \alpha \in X\}$ is decidable. In particular a set Σ of sentences is decidable if and only if $\{\ulcorner \sigma \urcorner \colon \sigma \in X\}$ is decidable. Thus, if Σ is a decidable set of sentences then $\{(\phi_0, \ldots, \phi_n) \colon \phi_0, \ldots, \phi_n$ is a deduction from $\Sigma\}$ and $\{\ulcorner \phi_0, \ldots, \phi_n \urcorner \colon \phi_0, \ldots, \phi_n$ is a deduction from $\Sigma\}$ are both decidable, and thus listable.

The following is of utmost importance:

PROPOSITION 2.1. *Assume the theory T has a decidable set of axioms. Then T is listable.*

PROOF. Let Σ be a decidable set of axioms for T, so $T = \{\sigma \colon \Sigma \vdash \sigma\}$. As pointed out above, the set of deductions from Σ is listable. Since the function taking a finite sequence of formulas to the last formula in the sequence is computable, it follows that $Y = \{\phi \in \mathrm{Fm}_{\mathcal{L}} \colon \Sigma \vdash \phi\}$ is listable. Since one can decide whether or not a formula is a sentence it follows that $T = \{\phi \in Y \colon \phi \in \mathrm{Sn}_{\mathcal{L}}\}$ is decidable *as a subset of* Y, hence T is listable by Proposition 1.2 □

WARNING: In the above proof Y is not necessarily decidable (as a subset of $\mathrm{Fm}_{\mathcal{L}}$), hence we do not have T decidable *as a subset of* $\mathrm{Fm}_{\mathcal{L}}$ (or $\mathrm{Sn}_{\mathcal{L}}$).

We will eventually see that the converse to Proposition 2.1 also holds. In the case of complete theories we can do better.

PROPOSITION 2.2. *Let T be a complete theory. Then T is decidable if and only if T has a decidable set of axioms.*

PROOF. If T is decidable then, as was already pointed out, T is itself a decidable set of axioms for T. Conversely, suppose T has a decidable set of axioms. Then by Proposition 2.1 T is listable, so $T = \{f(n) \colon n \in \omega\}$ for some computable f. We need to define an effectively computable g on $\mathrm{Sn}_{\mathcal{L}}$ such that

$$g(\sigma) = \begin{cases} 0 & \text{if } \sigma \in T \\ 1 & \text{if } \sigma \notin T \end{cases}$$

Since T is complete we know that for any $\sigma \in \mathrm{Sn}_{\mathcal{L}}$ either $\sigma \in T$ or $\neg\sigma \in T$, so there is some $n_0 \in \omega$ such that $f(n_0) = \sigma$ or $f(n_0) = \neg\sigma$. Our effective procedure to compute g is then as follows: evaluate $f(0), f(1), \ldots,$ until you find this n_0; then $g(\sigma) = 0$ if $f(n_0) = \sigma$, and $g(\sigma) = 1$ if $f(n_0) = \neg\sigma$. □

We point out here that when, in the next chapter, we introduce our formal, precise definition of computable, the above proofs will have to be reviewed to see that they can be carried out for the formal concept. This will, of course, be the case since otherwise the formal concept would not correspond to the intuitive concept. But it would be begging the question to assume in advance that the formal concept is as desired.

Everything can be done for languages $\mathcal{L}$ with infinitely many non-logical symbols provided $\mathcal{L}$ has an *admissible Gödel-numbering* g; that is, a Gödel-numbering g which is as on page 7 for logical symbols and such that for every n we can decide whether $n = g(S)$ for a non-logical symbol and precisely what sort of non-logical symbol S is (as, k-ary function symbol, etc.).

3. Gödel's Incompleteness Theorem

If $f\colon \omega \to \omega$ is a computable function then, in particular, the effective procedure for computing f gives an (English) definition of f. Similarly if X is a decidable subset of ω then the effective procedure for computing K_X yields an (English) definition of X as a subset of ω. Gödel's most important technical lemma for his Incompleteness Theorem states that a decidable subset of ω is in fact *first-order* definable in the usual structure for arithmetic on ω.

The language for arithmetic on ω has as non-logical symbols $+, \cdot, <, \overline{0}, s$. The "standard model" for arithmetic is $\mathfrak{N} = (\omega, +, \cdot, <, 0, s)$. Here s is a unary function symbol whose interpretation in $\mathfrak{N}$ is the immediate successor function, $s(n) = n+1$. We use $\overline{1}, \overline{2}, \overline{3}, \ldots,$ to stand for the closed terms $s\overline{0}, ss\overline{0}, \ldots,$ which are interpreted in $\mathfrak{N}$ by the elements $1, 2, 3, \ldots$. Note of course that every element of the universe of $\mathfrak{N}$ is the interpretation of some one of the terms $\overline{n}$.

Gödel's essential technical lemma can be stated as follows. (In the next chapter we will actually prove a stronger result, but this will suffice for now.)

3.1. Definability Lemma. *Let $R \subseteq \omega^n$ be a decidable n-ary relation on ω. Then there is some formula $\phi(x_0, \ldots, x_{n-1})$ of the language of arithmetic such that for any $k_0, \ldots, k_{n-1} \in \omega$ $R(k_0, \ldots, k_{n-1})$ holds if and only if $\mathfrak{N} \models \phi(\overline{k}_0, \ldots, \overline{k}_{n-1})$.*

We will not be able to *prove* this rather surprising result until we have a formal definition of computable function in the next chapter. We give here two examples of decidable relations on ω for which the existence of such a defining formula is highly non-obvious:

(1) Let $R = \{(k, 2^k)\colon k \in \omega\}$. $R \subseteq \omega \times \omega$ is clearly decidable, but not obviously definable in $\mathfrak{N}$ since exponentiation 2^x is not a function in the language.
(2) let $R = \{(k, n)\colon k \neq 0, \ln(k) \leq n\}$. Since we can calculate the natural log of any positive integer k to any required degree of accuracy (and it is non-integral for $k > 1$), R is a decidable subset of $\omega \times \omega$. Since $\ln(k)$ is not even an integer for $k > 1$, it is not easy to see even how to start defining R.

In the rest of this section we will *assume* the Definability Lemma and use it to prove Gödel's original incompleteness result.

3.2. Incompleteness Theorem (Gödel 1931). *There is no decidable set of axioms for $Th(\mathfrak{N})$*

We will in fact give two proofs of this result in this section. The first (and easiest) is a proof by contradiction. The second is a direct proof and yields a little more information.

Proof Number 1. Suppose $Th(\mathfrak{N})$ does have some decidable set of axioms. Then, by Proposition 2.2, $Th(\mathfrak{N})$ is decidable, hence one can decide whether or not $\mathfrak{N} \models \sigma$. Define the following binary relation R on ω:

$$(*)\begin{cases} R(k,l) \text{ holds if and only if } k = \ulcorner\phi\urcorner \text{ for} \\ \text{some formula } \phi(v_0) \text{ and } \mathfrak{N} \models \phi(\bar{l}). \end{cases}$$

Then R is decidable, and hence by the Definability Lemma R is defined in $\mathfrak{N}$ by some formula $\theta(x,y)$. Let $\phi(v_0)$ be $\neg\theta(v_0,v_0)$, let $k = \ulcorner\phi\urcorner$, and let σ be $\phi(\bar{k})$. The following are then equivalent:

$$\begin{array}{ll} \mathfrak{N} \models \sigma & \\ R(k,k) \text{ holds} & \text{(by (*))} \\ \mathfrak{N} \models \theta(\bar{k},\bar{k}) & \text{(by definition of } \theta) \\ \mathfrak{N} \models \neg\sigma & \text{(by definition of } \sigma) \end{array}$$

This contradiction shows that $Th(\mathfrak{N})$ has no decidable set of axioms. □

Our second proof follows the same general outline but actually shows us how to define a sentence independent of any given decidable set of sentences satisfied by $\mathfrak{N}$.

Proof Number 2. Let Σ be a decidable set of sentences such that $\mathfrak{N} \models \Sigma$. We show that $Th(\mathfrak{N}) \neq \mathrm{Cn}(\Sigma)$ by defining a sentence σ such that $\mathfrak{N} \models \sigma$ but $\Sigma \nvdash \sigma$.

We first define a ternary relation S on ω as follows:

$$\begin{cases} S(k,l,m) \text{ holds if and only if } k = \ulcorner\phi\urcorner \text{ for some} \\ \text{formula } \phi(v_0) \text{ and } m = \ulcorner\psi_0,\dots,\psi_n\urcorner, \text{ where} \\ \psi_0,\dots,\psi_n \text{ is a deduction from } \Sigma \text{ of } \phi(\bar{l}). \end{cases}$$

Then S is decidable, since Σ is decidable, hence by the Definability Lemma S is definable in $\mathfrak{N}$ by some $\chi(x,y,z)$. Now let $R(k,l)$ hold if and only if $k = \ulcorner\phi\urcorner$ for some formula $\phi(v_0)$ and $\Sigma \vdash \phi(\bar{l})$. Then $R(k,l)$ holds if and only if $S(k,l,m)$ holds for some $m \in \omega$. Therefore R is defined in $\mathfrak{N}$ by $\theta(x,y) = \exists z\chi(x,y,z)$. As in the first proof, let $\phi(v_0) = \neg\theta(v_0,v_0)$, let $k = \ulcorner\phi\urcorner$, and let $\sigma = \phi(\bar{k})$. Then the following are equivalent:

$$\begin{array}{ll} \mathfrak{N} \models \sigma & \\ \mathfrak{N} \models \neg\theta(\bar{k},\bar{k}) & \text{(by definition of } \sigma) \\ R(k,k) \text{ fails} & \text{(by definition of } \theta) \\ \Sigma \nvdash \phi(\bar{k}) & \text{(by definition of } R) \\ \Sigma \nvdash \sigma & \text{(by definition of } \sigma) \end{array}$$

We claim that in fact $\mathfrak{N} \models \sigma$. If not, then by the above equivalences $\Sigma \vdash \sigma$. But $\mathfrak{N} \models \Sigma$, so we must then have $\mathfrak{N} \models \sigma$. This contradiction shows that $\mathfrak{N} \models \sigma$ and $\Sigma \nvdash \sigma$, as desired. $\square$

Because of the equivalences in this proof, σ is usually interpreted as saying *of itself* that it is not provable from Σ. This is the "self-referential" aspect of the result.

Gödel's incompleteness result was especially surprising since it seems to have been widely assumed that the "Peano axioms" were a set of acioms for $Th(\mathfrak{N})$ We give these axioms here, but we first give a finite subset of them which we will use extensively.

The set Q consists of the following nine sentences:

$$
\begin{aligned}
&\forall x(\neg sx \equiv \overline{0}) \\
&\forall x \forall y(sx \equiv sy \to x \equiv y) \\
&\forall x(x + \overline{0} \equiv x) \\
&\forall x \forall y(x + sy \equiv s(x+y)) \\
&\forall x(x \cdot \overline{0} \equiv \overline{0}) \\
&\forall x \forall y(x \cdot sy \equiv x \cdot y + x) \\
&\forall x(\neg x < 0) \\
&\forall x \forall y(x < sy \leftrightarrow (x < y \vee x \equiv y)) \\
&\forall x \forall y(x < y \vee x \equiv y \vee y < x)
\end{aligned}
$$

The set P of Peano axioms consists of Q together with all "induction axioms," that is, all sentences $\forall y_0 \ldots \forall y_{n-1}[\phi(\overline{0}, \vec{y}) \wedge \forall x(\phi(x, \vec{y}) \to \phi(sx, \vec{y})) \to \forall x \phi(x, \vec{y})]$ as $\phi(x, \vec{y})$ varies over formulas of $\mathcal{L}$.

Clearly $\mathfrak{N} \models P$, and both P and Q are decidable. Finding sentences true on $\mathfrak{N}$ not provable from P is essentially as difficult as the general result.

4. Some Positive Decidability Results

Proposition 2.2 can be used to show that many complete theories are decidable. For example $Th((\mathbb{Q}, \leq))$ and $Th((\omega, \leq))$ are both finitely axiomatizable, hence decidable. The theories $Th((\mathbb{R}, +, \cdot, \leq, 0, 1))$ and $Th((\mathbb{C}, +, \cdot, 0, 1))$ can each be shown to have decidable (but infinite) sets of axioms, hence are decidable.

The primary question we started with—whether $\{\sigma \in \mathrm{Sn}_{\mathcal{L}} \colon\ \models \sigma\}$ is decidable—was answered in general by Church in 1936. Previous to that some positive results were obtained by restricting the classes of sentences considered. In the remainder of this section we briefly present some of these results.

Throughout this section we assume $\mathcal{L}$ has just finitely many non-logical symbols, but they hold for languages with an admissable Gödel-numbering.

These positive results all follow using this lemma:

LEMMA 4.1. *Assume that S is a decidable subset of $\mathrm{Sn}_{\mathcal{L}}$ such that for every $\sigma \in S$, $\models \sigma$ if and only if $\mathfrak{A} \models \sigma$ for every* finite $\mathfrak{A}$. *Then $\{\sigma \in S \colon\ \models \sigma\}$ is decidable.*

PROOF. First note that, since $\mathcal{L}$ is finite, there are for each $n \in \omega$ just finitely many non-isomorphic $\mathcal{L}$-structures $\mathfrak{A}$ with $|A| = n$. Further, for any $\sigma \in \mathrm{Sn}_{\mathcal{L}}$ and any given finite $\mathfrak{A}$ we can effectively decide whether or not $\mathfrak{A} \models \sigma$ (because $\mathfrak{A} \models \forall x \phi(x)$ is equivalent to $\mathfrak{A}_A \models \phi(\overline{a}_1) \wedge \ldots \wedge \phi(\overline{a}_k)$, where $A = \{a_1, \ldots, a_k\}$, etc.). Thus, for each n we can effectively decide whether or not $\mathfrak{A} \models \sigma$ for every $\mathfrak{A}$

with $|A| = n$. Further, we know we can effectively list all $\sigma \in \mathrm{Sn}_{\mathcal{L}}$ such that $\models \sigma$. Thus our effective procedure to decide, given $\sigma_0 \in S$, whether or not $\models \sigma_0$ is as follows: we simultaneously begin listing all σ such that $\models \sigma$ *and* begin deciding, for each $n \in \omega - \{0\}$, whether or not $\mathfrak{A} \models \sigma_0$ for every $\mathfrak{A}$ with $|A| = n$. We will, after some finite number of steps, *either* find σ_0 in our list of validities *or* find some n such that $\mathfrak{A} \not\models \sigma_0$ for some $\mathfrak{A}$ such that $|A| = n$ (by our assumption on S that if $\not\models \sigma$ then $\mathfrak{A} \not\models \sigma$ for some finite $\mathfrak{A}$). Since S is a decidable subset of $\mathrm{Sn}_{\mathcal{L}}$ this in fact shows that $\{\sigma \in S \colon \models \sigma\}$ is decidable as a subset of $\mathrm{Sn}_{\mathcal{L}}$. □

The above procedure is not very efficient, and it turns out that, in each case below in which it is used, lookiing closely at the proof one can obtain a more efficient procedure.

Definition 4.1. A formula is a $\forall$-*formula* (universal formula) provided it has the form $\forall y_0 \dots \forall y_{k-1}\alpha$, where α is open.

Definition 4.2. A formula is an $\forall\exists$-*formula* provided it has the form $\forall y_0 \dots \forall y_{k-1} \exists z_0 \dots \exists z_{l-1}\alpha$, where α is open.

Our first result concerns the valid $\forall\exists$-sentences in languages without function symbols.

Proposition 4.2. *Assume $\mathcal{L}$ has no function symbols. Then $\{\sigma \colon \models \sigma$, σ is an $\forall\exists$-sentence of $\mathcal{L}\}$ is decidable.*

Proof. It suffices to show that for any $\forall\exists$-sentence σ, if $\not\models \sigma$ then $\mathfrak{A} \models \neg\sigma$ for some finite $\mathfrak{A}$. So suppose $\mathfrak{B} \models \neg\sigma$. We find a finite $\mathfrak{A} \subseteq \mathfrak{B}$ such that $\mathfrak{A} \models \neg\sigma$. If $\mathfrak{B} \models \neg\sigma$, where σ is $\forall\vec{y}\,\exists\vec{z}\alpha(\vec{y},\vec{z})$, then $\mathfrak{B} \models \exists y_0 \dots \exists y_{k-1} \forall z_0 \dots \forall z_{l-1} \neg\alpha(\vec{y},\vec{z})$, so $\mathfrak{B}_{B_0} \models \forall z_0 \dots \forall z_{l-1} \neg\alpha(\bar{b}_0, \dots, \bar{b}_{k-1}, \vec{z})$ for some $b_0, \dots, b_{k-1} \in B$, with $B_0 = \{b_0, \dots, b_{k-1}\}$. Let $\mathfrak{A} \subseteq \mathfrak{B}$ be such that $B_0 \subseteq A$ and A is finite—this is possible since $\mathcal{L}$ has no function symbols and just finitely many constant symbols. One then easily sees that $\mathfrak{A}_{B_0} \models \forall z_0 \dots \forall z_{l-1} \neg\alpha(\bar{b}_0, \dots, \bar{b}_{k-1}, \vec{z})$ since $\neg\alpha$ has no quantifiers. Thus $\mathfrak{A} \models \neg\sigma$ as desired. □

Note that this proof in fact enables us to effectively compute, given an $\forall\exists$-sentence σ of $\mathcal{L}$, an integer $n_0 \in \omega$ such that if $\not\models \sigma$ then in fact $\mathfrak{A} \models \neg\sigma$ for some $\mathfrak{A}$ with $|A| \le n_0$. We can thus decide whether or not $\models \sigma$ by just checking whether $\mathfrak{A} \models \sigma$ for all $\mathfrak{A}$ with $|A| \le n_0$. This is much more efficient than the procedure given by Lemma 4.1.

If $\mathcal{L}$ has function symbols the best we can do is the following:

Proposition 4.3. *$\{\sigma \colon \models \sigma$, σ is a $\forall$-sentence$\}$ is decidable.*

Once again, you need to show that if $\mathfrak{B} \models \neg\sigma$ for some $\mathfrak{B}$ then $\mathfrak{A} \models \neg\sigma$ for some finite $\mathfrak{A}$. Here, one cannot necessarily find $\mathfrak{A} \subseteq \mathfrak{B}$ but the $\mathfrak{A}$ can be obtained from $\mathfrak{B}$. Details are left to the reader.

If we do not restrict the form of the sentence then we must severely restrict the language.

Proposition 4.4. *Assume the only non-logical symbols of $\mathcal{L}$ are unary predicates and individual constants. Then $\{\sigma \in \mathrm{Sn}_{\mathcal{L}} \colon \models \sigma\}$ is decidable.*

Here again, given some $\mathfrak{B} \models \neg\sigma$ you must find a finite $\mathfrak{A} \subseteq \mathfrak{B}$ such that $\mathfrak{A} \models \neg\sigma$.

5. Exercises

(1) Assume that X and Y are both listable.
   (a) Prove that $(X \cup Y)$ is listable.
   (b) Prove that $(X \cap Y)$ is listable.

(2) Let X be an infinite subset of ω. Prove that X is decidable iff X is the range of some $f : \omega \to X$ which is computable and strictly increasing (i.e. if $k < l$ then $f(k) < f(l)$).

(3) Let $X \subseteq \omega$. Assume that both X and $(\omega \setminus X)$ are listable. Prove that X is decidable.

(4) Let $\mathcal{L}^{nl} = \{+, \cdot, <, x, \overline{0}\}$, and let $\mathfrak{N} = (\omega, +, \cdot, <, s, 0)$. Give a formula $\varphi(x)$ of $\mathcal{L}$ which defines the set of sequence numbers in $\mathfrak{N}$.

CHAPTER 8

Recursive Functions, Representability, and Incompleteness

0. Introduction

In this chapter we give a precise, formal, mathematical definition of computable function to replace the informal concept of the preceding chapter. Because of Gödel-numbering, we can restrict our attention to functions on the natural numbers. Our mathematical definition of computable is *recursive*. The recursive functions are defined in Section 1, and numerous (intuitively) computable functions (including many concerning Gödel numbers of formulas, etc.) are shown to be recursive. In Section 3 we prove the *representability* of recursive functions in Q, a stronger version of the Definability Lemma from Section 3. An essential tool is the β-function defined in Section 2. Finally, in Section 4, we can prove Gödel's Incompleteness Theorem, essentially as in Section 3.

Of course, for this development to be convincing, one must accept the identification of the intuitive concept of "computable" with "recursive." This identification is called Church's Thesis, and some arguments for it are given in Section 5.

1. Recursive Functions and Relations

The set of recursive functions on ω is defined by recursion. That is, we are given some set of starting functions and some rules for forming new functions from given functions. A function then is said to be recursive if and only if it is obtained from the starting functions after some (finite) number of applications of the rules. A relation is recursive if and only if its characteristic function is recursive, where:

Definition 1.1. Let $R \subseteq \omega^n$ The *characteristic function* of R is the function $K_R\colon \omega^n \to \{0,1\}$ such that

$$K_R(k_1,\dots,k_n) = \begin{cases} 0 & \text{if } R(k_1,\dots,k_n) \text{ holds,} \\ 1 & \text{otherwise.} \end{cases}$$

Definition 1.2. For all $1 \le i \le n$ the *projection function* $P_i^n\colon \omega^n \to \omega$ is defined by $P_i^n(k_1,\dots,k_n) = k_i$.

Definition 1.3. The set $\mathcal{S}$ of *starting functions* is the set $\{s, +, \cdot,\ K_<\} \cup \{P_i^n\colon 1 \le i \le n \in \omega\}$.

We have three rules for forming new functions:

***R*1 (Composition).** From functions $G(x_1,\dots,x_k)$, $H_1(y_1,\dots,y_m)$, …, $H_k(y_1,\dots,y_m)$ form $F(y_1,\dots,y_m) = G(H_1(\vec{y}),\ \dots,\ H_k(\vec{y}))$.

***R*2 (Primitive Recursion).** From functions $G(y_1, \ldots, y_m)$ and $H(w, x, y_1, \ldots, y_m)$ form $F(x, y_1, \ldots, y_m)$ defined by

$$\begin{cases} F(0, y_1, \ldots, y_m) = G(y_1, \ldots, y_m), \\ F(x+1, y_1, \ldots, y_m) = H(F(x, \vec{y}), x, \vec{y}). \end{cases}$$

***R*3 (μ-Recursion).** From a function $G(y_1, \ldots, y_m, x)$ satisfying the condition that for all $y_1, \ldots, y_m \in \omega$ there exists $x \in \omega$ such that $G(y_1, \ldots, y_m, x) = 0$ form the function $F(y_1, \ldots, y_m)$ defined by $F(y_1, \ldots, y_m) =$ the smallest $x \in \omega$ such that $G(\vec{y}, x) = 0$. Our notation is: $F(\vec{y}) = \mu x[G(\vec{y}, x) = 0]$.

Definition 1.4. The class $\mathcal{R}$ of *recursive functions* is the closure of $\mathcal{S}$ under $R1, R2, R3$; that is, a function is in $\mathcal{R}$ if and only if it is obtained from functions in $\mathcal{S}$ by (finitely many) applications of $R1, R2, R3$.

Two smaller classes of functions are also of interest.

Definition 1.5.

(a) The class $\mathcal{P}$ of *primitive recursive* functions is the closure of $\mathcal{S}$ under $R1, R2$.

(b) The class $\mathcal{R}'$ of *strictly recursive* functions is the closure of $\mathcal{S}$ under $R1, R3$.

Clearly $\mathcal{P}, \mathcal{R}' \subseteq \mathcal{R}$, but the relation between $\mathcal{R}'$ and $\mathcal{P}$ is not clear, nor is it obvious whether or not the inclusions are proper.

Definition 1.6. A relation R on ω is *recursive* (or *primitive recursive*, or *strictly recursive*) if K_R is.

Note that the functions in $\mathcal{S}$ are all effectively computable. Further, if the given functions are computable, so is the function F produced by each of our rules $R1, R2, R3$—if F is obtained by $R2$ we compute $F(k, \vec{y})$ by using the defining equations to successively compute $F(0, \vec{y}), F(1, \vec{y}), \ldots, F(k-1, \vec{y}), F(k, \vec{y})$; if F is obtained by $R3$ we compute $F(\vec{y})$ by successively computing $G(\vec{y}, 0), G(\vec{y}, 1), G(\vec{y}, 2)$ until we find a k with $G(\vec{y}, k) = 0$, the first such being the value of F. Thus, by induction, we conclude that all recursive functions are computable, hence that all recursive relations are decidable.

From $R1, R2, R3$ follow corresponding rules for relations:

***R*1′.** From $R(x_1, \ldots, x_k)$ and functions $H_i(\vec{y}), \ldots, h_k(\vec{y})$ form $S(\vec{y}) \Leftrightarrow R(H_i(\vec{y}), \ldots, h_k(\vec{y}))$.

***R*2′.** From $R(\vec{y})$ and $Q(w, x, \vec{y})$ form $S(x, \vec{y})$ defined by

$$\begin{cases} S(0, \vec{y}) \Leftrightarrow R(\vec{y}), \\ S(x+1, \vec{y}) \Leftrightarrow Q(K_S(x, \vec{y}), x, \vec{y}). \end{cases}$$

***R*3′.** From $R(\vec{y}, x)$ which satisfies the condition that $\forall y_1, \ldots, y_m \in \omega\, \exists x \in \omega\, R(\vec{y}, x)$ holds, form $F(\vec{y}) = \mu x R(\vec{y}, x)$.

Then the S or F obtained will be recursive provided the given relations and functions are.

Since all the functions and relations we are considering are *on* ω, we will often not explicitly say so; similarly, the variables we use in writing them down all vary over elements of ω, which will be tacitly assumed. Also, of course, the functions we consider are all *total*, that is, defined on *all* of ω^n.

We now begin to list a large number of facts about recursive functions and relations. These will state that many common functions and relations are recursive and will give further closure properties of $\mathcal{R}$, and also $\mathcal{P}$ and $\mathcal{R}'$. We will sometimes say a relation R belongs to $\mathcal{R}$ (or $\mathcal{P}$ or $\mathcal{R}'$) when strictly we mean K_R does.

Fact 1. *The constant functions are in $(\mathcal{P} \cap \mathcal{R}')$. That is, for every $m, n \in \omega$ with $m \neq 0$ the function $F_n^m \colon \omega^n \to \omega$ defined by $F_n^m(x_1, \ldots, x_m) = n$ for all $x_1, \ldots, x_m$ is in $(\mathcal{P} \cap \mathcal{R}')$.*

Proof. Fixing m, we show $F_n^m \in (\mathcal{P} \cap \mathcal{R}')$ by induction on n. The inductive step is clear, since $F_{n+1}^m(\vec{x}) = s(F_n^m(\vec{x}))$ is obtained by $R1$ from s, F_n^m. To show $F_0^m \in (\mathcal{P} \cap \mathcal{R}')$, note that $K_<(x_1, s(x_1)) = 0$ for all x_1. We can therefore define F_0^m by

$$F_0^m(x_1, \ldots, x_m) = K_<(P_1^m(x_1, \ldots, x_m), s(P_1^m(x_1, \ldots, x_m))),$$

thus obtaining F_0^m by *two* uses of $R1$. $\square$

Note that simply writing $F_0^m(x_1, \ldots, x_m) = K_<(x_1, s(x_1))$ is *not* a correct use of $R1$—we need the projection functions so that each function put into $K_<(\, , \,)$ is a function of m arguments.

At this point we should emphasize that functions and relations do not come with any particular variables attached—

$$K_<(P_1^m(x_1, \ldots, x_m), s(P_1^m(x_1, \ldots, x_m)))$$

is the same function as

$$K_<(P_1^m(y_1, \ldots, y_m), s(P_1^m(y_1, \ldots, y_m))),$$

or

$$K_<(P_1^m(y, x_1, \ldots, x_{m-1}), s(P_1^m(y, x_1, \ldots, x_{m-1}))).$$

Also, be careful not to think of relations as formulas. They are simply subsets of some ω^n, although we will sometimes define them by some mathematical formula.

We will (perhaps confusingly) use logical notations in defining relations.

Definition 1.7. Given a relation $R(\vec{x})$, $\neg R(\vec{x})$ is the relation (on ω) which holds if and only if R fails. Similarly, given relations $R(\vec{x}), S(\vec{x})$ we understand the relations $(R \wedge S), (R \vee S), (R \to S)$ in the obvious way. Given $R(x, \vec{y}), \exists x R(x, \vec{y})$ is the relation $S(\vec{y})$ which holds if and only if $R(x, \vec{y})$ holds for some x. The relation $\forall x R(x, \vec{y})$ is similar.

Fact 2. *Assume R, S are both in $\mathcal{R}(or \mathcal{P} or \mathcal{R}')$. Then so are $\neg R$, $(R \wedge S)$, $(R \vee S)$, $(R \to S)$.*

Proof. Given $R(\vec{x}), K_{\neg R}(\vec{x}) = 0$ if $K_R(\vec{x}) = 1$ and $K_{\neg R}(\vec{x}) = 1$ if $K_R(\vec{x}) = 0$. Thus $K_{\neg R}(\vec{x}) = K_<(F_0^m(\vec{x}), K_R(\vec{x}))$, where F_0^m is constantly 0. Since $K_{(R \vee S)}(\vec{x}) = 0$ if and only if at least one of $K_R(\vec{x}), K_S(\vec{x}) = 0$ we can define $K_{(R \vee S)}(\vec{x}) = K_R(\vec{x}) \cdot K_S(\vec{x})$. The results $(R \wedge S), (R \to S)$ follow from these. $\square$

Fact 3. *The relations $<, >, \leq, \geq, =$ are in $(\mathcal{P} \cap \mathcal{R}')$.*

PROOF. $K_<$ is in $\mathcal{S}$, so $<$ is in $(\mathcal{P} \cap \mathcal{R}')$. $K_>(x, y) = 0$ if $y < x, 1$ if not. Therefore $K_>(x, y) = K_<(P_2^2(x, y), P_1^2, (x, y))$. The others now follow by using Fact 2. □

In general, there is no reason for $\exists y R$ to be recursive just because R is recursive. But *bounded* quantification does preserve recursiveness.

DEFINITION 1.8.

(a) $(\exists x)_{<y} R(x, y, \vec{z}) \Leftrightarrow \exists x[x < y \wedge R(x, y, \vec{z})]$.
(b) $(\forall x)_{<y} R(x, y, \vec{z}) \Leftrightarrow \forall x[x < y \rightarrow R(x, y, \vec{z})]$.
(c) $(\mu x)_{<y} R(x, y, \vec{z}) = (\mu x)[(x < y \wedge R(x, y, \vec{z})) \vee x = y]$.

Fact 4. *If R is in $\mathcal{R}$ (or $\mathcal{P}$ or $\mathcal{R}'$) then so are $(\mu x)_{<y} R$, $(\exists x)_{<y} R$, $(\forall x)_{<y} R$.*

PROOF. If R is in $\mathcal{R}$ (or $\mathcal{R}'$) then so is $(x < y \wedge R(x, y, \vec{z})) \vee x = y$, by Facts 2 and 3, hence $R3$ yields $(\mu x)_{<y} R$ in the same class. If R is in $\mathcal{P}$ a different argument, using $R2$, must be used. We leave this to the reader. We now have $(\exists x)_{<y} R \Leftrightarrow (\mu x)_{<y} R < y$, hence this relation is in $\mathcal{R}$ (or $\mathcal{R}'$ or $\mathcal{P}$) if R is. Similarly for $(\forall x)_{<y} R$. □

The reader should note that we are suppressing use of the projection functions. We need to have "$x < y$" and "$x = y$" as relations in $x, y, \vec{z}$, just like R, in order to combine them with R. We will continue to be inexact in this matter in the future.

The way we have written the preceding material, it seems as if the argument "quantified out" in $(\exists x)_{<y} R$, etc., must be the *first* argument in R. Obviously, of course, this is irrelevant, and we will apply these results and notations to any argument. The reader should be able to prove that if a function or relation is in $\mathcal{R}$ (or $\mathcal{P}$ or $\mathcal{R}'$) then it remains in that collection for any permutation of the arguments.

DEFINITION 1.9.

$$x \dot{-} y = \begin{cases} x - y & \text{if } y \leq x, \\ 0 & \text{otherwise.} \end{cases}$$

Fact 5. *$\dot{-}$ is in $(\mathcal{P} \cap \mathcal{R}')$.*

PROOF. $x \dot{-} y = (\mu z)_{<x}(y + z \geq z)$. □

Of course, the function $x - y$ can't be in $\mathcal{R}$ since it is not total with values in ω.

Fact 5 (Definition by Cases). *Let $G_1(\vec{y}), \ldots, G_k(\vec{y})$ be functions and $R_1(\vec{y}), R_k(\vec{y})$ be relations which are all in $\mathcal{R}$ (or $\mathcal{P}$ or $\mathcal{R}'$). Assume that for all $\vec{y}$ (from ω) exactly one of $R_1(\vec{y}), \ldots, R_k(\vec{y})$ holds. Let $F(\vec{y}) = G_i(\vec{y})$ provided $R_i(\vec{y})$ holds. Then F is also in $\mathcal{R}$ (or $\mathcal{P}$ or $\mathcal{R}'$).*

PROOF. $F(\vec{y}) = G_1(\vec{y}) \cdot K_{\neg R_1}(\vec{y}) + \cdots + G_k(\vec{y}) \cdot K_{\neg R_k}(\vec{y})$ establishes the conclusion. □

We next want to see that various operations on sequence numbers are recursive. This is the first step to being able to work with Gödel numbers recursively. It turns out that we need rule $R2$ to make these definitions, so it is not clear if the functions are in $\mathcal{R}'$.

Fact 7. *x^y is in $\mathcal{P}$.*

PROOF. The equations
$$\begin{cases} x^0 = 1, \\ x^{y+1} = x^y \cdot x, \end{cases}$$
define x^y by $R2$. □

DEFINITION 1.10. $p\colon \omega \to \omega$ is defined by $p(0) = 2, p(i) =$ the i^{th} odd prime for $i \neq 0$. We usually write p_i in place of $p(i)$.

Fact 8. *p is in $\mathcal{P}$.*

PROOF. First notice that the relation $P(x)$, "x is prime," is in $\mathcal{P}$ since it can be defined using just bounded quantifiers. Thus the equations
$$\begin{cases} p(0) = 2, \\ p(i+1) = (\mu x)[P(x) \wedge p(i) < x], \end{cases}$$
define p with $R2$ and show that p is in $\mathcal{R}$. Since we can bound x by $p(i)!$, and since $y!$ is in $\mathcal{P}$, we in fact find that p is in $\mathcal{P}$. □

DEFINITION 1.11. $\mathrm{Seq}(n) \Leftrightarrow n$ is a *sequence number*, that is, $n = 1$ or $n = \langle n_0, \ldots, n_k \rangle$ for some $n_0, \ldots, n_k, k \in \omega$.

Fact 9. *Seq is in $\mathcal{P}$.*

PROOF. $\mathrm{Seq}(n)$ holds if and only if $n \neq 0$ and $(\forall i)_{<n}(p_{i+1}|n \to p_i|n)$, and $x|y$ is easily seen to be in $\mathcal{P}$, by bounded quantification. □

DEFINITION 1.12. If $n = \langle n_0, \ldots, n_k \rangle$ is a sequence number then $\mathrm{lh}(n) = k+1$, the *length* of the sequence (number); $\mathrm{lh}(\langle\,\rangle) = 0$.

Fact 10. lh *is in $\mathcal{P}$.*

NOTE: As defined lh is not *total*, hence can't be in $\mathcal{P}$. What we mean here is that there is some function lh' in $\mathcal{P}$ (hence total) which agrees with lh on sequence numbers. Since we don't care about the values of this function for numbers which are not sequence numbers, we simply use lh to also refer to this total extension lh'

PROOF. $\mathrm{lh}(n) = (\mu x)_{<n}(p_x \nmid n)$. □

DEFINITION 1.13. $C(n,i) = n_i$ if $n = \langle n_0, \ldots, n_k \rangle, i \leq k$, is the *coordinate* function. We usually write $(n)_i$ for the value of $C(n,i)$, the i^{th} coordinate.

Fact 11. *C is in $\mathcal{P}$.*

PROOF. We can define C on all pairs n, i by
$$C(n,i) = (\mu x)_{<n}[(p_i)^{x+2} \nmid n].$$

□

Note that we have the following:

$$\mathrm{Seq}(n) \Leftrightarrow n = \langle (n)_0, (n)_1, \ldots, (n)_{\mathrm{lh}(n) \dot{-} 1} \rangle.$$

1.1. Definition (Concatenation). If $n = \langle n_0, \ldots, n_k \rangle$ and $m = \langle m_0, \ldots, m_l \rangle$ then the *concatenation* of n and m is the sequence number

$$n * m = \langle n_0, \ldots, n_k, m_0, \ldots, m_l \rangle.$$

Fact 12. *$*$ is in $\mathcal{P}$.*

PROOF. $n * m = (\mu x)[\mathrm{Seq}(x) \wedge \mathrm{lh}(x) = \mathrm{lh}(n) + \mathrm{lh}(m) \wedge (\forall i)_{<\mathrm{lh}(n)} [(x)_i = (n)_i] \wedge (\forall i)_{<\mathrm{lh}(m)}[(m)_i = (x)_{i+\mathrm{lh}(n)}]]$. This shows $*$ is recursive—note that there always is an x with these properties, even if n, m are not sequence numbers. To get $*$ in $\mathcal{P}$ we must bound x, which we leave to the reader. □

DEFINITION 1.14. If $n = \langle n_0, \ldots, n_{l-1} \rangle$ is a sequence number of length l, then for any $i \leq l$ the *initial segment* of n of length i is $\mathrm{In}(n, i) = \langle n_0, \ldots, n_{i-1} \rangle$.

Fact 13. In *is in $\mathcal{P}$.*

PROOF. The equations

$$\begin{cases} \mathrm{In}(n, 0) = 1, \\ \mathrm{In}(n, i+1) = \mathrm{In}(n, i) \cdot p_i^{(n)_i + 1}, \end{cases}$$

define In via $R2$. □

DEFINITION 1.15. Given $F(y, \vec{x})$ the *course of valuescourse of values function* function $\overline{F}(y, \vec{x})$ is defined by

$$\overline{F}(y, \vec{x}) = \langle F(0, \vec{x}), F(1, \vec{x}), \ldots, F(y-1, \vec{x}) \rangle.$$

Fact 14. *F is in $\mathcal{R}$ (or $\mathcal{P}$) if and only if $\overline{F}$ is.*

PROOF. From right to left is easy and left to the reader. We show how to get $\overline{F}$ from F in a recursive way. Clearly $\overline{F}(y, \vec{x}) = (\mu z)[\mathrm{Seq}(z) \wedge \mathrm{lh}(z) = y \wedge (\forall i)_{<y} (z)_i = F(i, \vec{x})]$. Once again, to get $\overline{F}$ in $\mathcal{P}$ if F is we need appropriately bound z. □

We next introduce a stronger form of primitive recursion ($R2$). This corresponds to so-called "strong" induction. That is, we define the value of F at k not just in terms of the value of F at $k - 1$, but in terms of the entire sequence of previous values of F. In the relational form, it enables one to use whether or not k is in a set S for *each* $k < n$ in defining whether or not n should belong to S. This is exactly the sort of definition one has for the set of terms, of formulas, etc.

Fact 14 (Course-of-Values Recursion).

(a) If $G(w, y, \vec{x})$ is in $\mathcal{R}$ (or $\mathcal{P}$) then so is the function $F(y, \vec{x})$ defined by

$$F(y, \vec{x}) = G(\overline{F}(y, \vec{x}), y, \vec{x}).$$

(b) If $S(w, x, \vec{x})$ is in $\mathcal{R}$ (or $\mathcal{P}$) then so is the relation $R(y, \vec{x})$ defined by

$$R(y, \vec{x}) \Leftrightarrow S(\overline{K}_R(y, \vec{x}), y, \vec{x}).$$

Proof. We just prove (a), from which (b) immediately follows. Given G we first define $H(y, \vec{x})$ by

$$H(y, \vec{x}) = (\mu z)[\mathrm{Seq}(z) \wedge \mathrm{lh}(z) = y \wedge (\forall i)_{<y}(z)_i = G(\mathrm{In}(z, i), \vec{x})].$$

One can see that $H(y, \vec{x}) = \overline{F}(y, \vec{x})$, so $\overline{F}$ (and hence F) is in $\mathcal{R}$ if G is. We leave to the reader the task of showing that H is in $\mathcal{P}$ if G is. □

Since we know that the standard operations with sequence numbers are recursive, we are now in a position to prove that most of the standard syntactical concepts are recursive—when expressed in terms of Gödel numbers.

We continue our assumption that $\mathcal{L}$ has just finitely many non-logical symbols. For purposes of examples, we take $\mathcal{L}$ to be the language of arithmetic, and so $g(+) = 13, g(\cdot) = 15, g(<) = 17, g(s) = 19, g(\overline{0}) = 21$.

Definition 1.16.

(a) $\mathrm{Tm}(n) \Leftrightarrow n = \ulcorner t \urcorner$ for some $t \in Tm_{\mathcal{L}}$.

(b) $\mathrm{Fm}(n) \Leftrightarrow n = \ulcorner \phi \urcorner$ for some $\phi \in \mathrm{Fm}_{\mathcal{L}}$.

(c) $\mathrm{Free}(k, n) \Leftrightarrow k = \ulcorner v_i \urcorner$ for some i and $n = \ulcorner \phi \urcorner$ for some $\phi \in \mathrm{Fm}_{\mathcal{L}}$ and $v_i \in \mathrm{Fr}(\phi)$.

(d) $\mathrm{Num}(k) = \ulcorner \overline{k} \urcorner$.

(e) $\mathrm{Neg}(k) = \ulcorner \neg\phi \urcorner$, if $k = \ulcorner \phi \urcorner$ for some $\phi \in \mathrm{Fm}_{\mathcal{L}}$.

(f) $\mathrm{Sn}(n) \Leftrightarrow n = \ulcorner \sigma \urcorner$ for some $\sigma \in \mathrm{Sn}_{\mathcal{L}}$.

(g) $\mathrm{Sub}(n, k, m) = \ulcorner \phi_t^{v_i} \urcorner$, provided $n = \ulcorner \phi \urcorner, k = \ulcorner v_i \urcorner$, and $m = \ulcorner t \urcorner$.

(h) $\mathrm{Fm}_0(n) \Leftrightarrow n = \ulcorner \phi \urcorner$ for some formula $\phi(v_0)$.

Fact 16. *Everything in the preceding definition is in $\mathcal{P}$.*

Proof. We simply show Tm is in $\mathcal{P}$ and leave the rest to the diligent student. We see that Tm can be defined as follows:

$$\begin{aligned}\mathrm{Tm}(n) \quad \Leftrightarrow \quad & (\exists x)_{<n}(n = 2^{2x+1}) \vee n = 2^{g(\overline{0})+1} \vee \\ & (\exists x)_{<n}(\mathrm{Tm}(x) \wedge n = 2^{g(s)+1} * x) \vee \\ & (\exists x)_{<n}(\exists y)_{<n}[\mathrm{Tm}(x) \wedge \mathrm{Tm}(y) \wedge n = 2^{g(+)+1} * x * y] \vee \\ & (\exists x)_{<n}(\exists y)_{<n}[\mathrm{Tm}(x) \wedge \mathrm{Tm}(y) \wedge n = 2^{g(\cdot)+1} * x * y].\end{aligned}$$

This definition is justified by Fact 14b, course-of-values recursion for relations, since the right-hand side just involves $\mathrm{Tm}(x), \mathrm{Tm}(y)$ for $x, y < n$. □

Definition 1.17. $\mathrm{LAx}(n) \Leftrightarrow n = \ulcorner \phi \urcorner$ for some logical axiom ϕ.

Fact 17. LAx *is in $\mathcal{P}$.*

We will say that a set Σ of sentences is *recursive* if $\{\ulcorner\sigma\urcorner\colon \sigma\in\Sigma\}$ is recursive.

DEFINITION 1.18.

(a) $\mathrm{Prf}_\Sigma(n) \Leftrightarrow n = \ulcorner\phi_0,\dots,\phi_k\urcorner$ for some deduction $\phi_0,\dots,\phi_k$ from Σ.
(b) $\mathrm{Prv}_\Sigma(n,m) \Leftrightarrow n = \ulcorner\phi_0,\dots,\phi_k\urcorner$ for some deduction $\phi_0,\dots,\phi_k$ from Σ and $m=\ulcorner\phi_k\urcorner$.

Fact 18. *If Σ is recursive, so are* Prf_Σ *and* Prv_Σ.

DEFINITION 1.19. $\mathrm{Thm}_\Sigma(m) \Leftrightarrow m=\ulcorner\sigma\urcorner$ for some sentence σ such that $\Sigma\vdash\sigma$.

If $T=\mathrm{Cn}(\Sigma)$ then T is *decidable* if and only if Thm_Σ is recursive. This is, of course, not obviously guaranteed by Σ being recursive. It is useful to note the following:

LEMMA 1.2. *T is decidable if and only if $\exists z\mathrm{Prv}_\Sigma(z,m)$ is in $\mathcal{R}$ for some (or any) Σ which axiomatizes T.*

PROOF. $\mathrm{Thm}_\Sigma(m) \Leftrightarrow \mathrm{Sn}(m)\wedge\exists z\mathrm{Prv}_\Sigma(z,m)$. Thus T is decidable provided $\exists z\mathrm{Prv}_\Sigma(z,m)$ is in $\mathcal{R}$ for some Σ with $T=\mathrm{Cn}(\Sigma)$. For the other direction, note that for formulas $\phi(v_0,\dots,v_n)$ we have $\Sigma\vdash\phi$ if and only if $\Sigma\vdash\forall V_0\dots\forall v_n\phi$. As an exercise, define a primitive recursive function cl such that $cl(\ulcorner\phi\urcorner)=\ulcorner\forall v_0\dots\forall v_n\phi\urcorner$ for some closure $\forall v_0\dots\forall v_n\phi$ of ϕ. Then

$$\exists z\mathrm{Prv}_\Sigma(z,m)\Leftrightarrow\mathrm{Thm}_\Sigma(cl(m))$$

shows that $\exists z\mathrm{Prv}_\Sigma(z,m)$ is recursive provided T is decidable. □

2. Gödel's β–Function

All that we need in order to carry out the (second) proof of the Incompleteness Theorem in Chapter **??** is the Definability Lemma for recursive functions and predicates. Clearly all the starting functions are definable in $\mathfrak{N}$. Also it is easy to see that $R1$ and $R3$ preserve definability. Further, a relation is definable in $\mathfrak{N}$ if and only if its characteristic function is. Thus, by induction, we see that every *strictly* recursive function and relation is definable in $\mathfrak{N}$.

The problem, then, is with $R2$, the rule of primitive recursion. The reason $R2$ is a problem is that the function $F(x,\vec{y})$ obtained by use of the rule is not given an explicit definition. Gödel showed how to give an explicit definition of an F obtained in this way *and* how to turn it into a first-order definition in $\mathfrak{N}$ if the given functions $G(\vec{y}), H(w,x,\vec{y})$ are first-order definable in $\mathfrak{N}$.

In fact, Gödel's argument shows how to eliminate uses of $R2$ altogether, and thus establishes the following result:

THEOREM 2.1. $\mathcal{R}=\mathcal{R}'$.

This follows from the proposition asserting that $\mathcal{R}'$ is closed under applications of $R2$, that is:

PROPOSITION 2.2. *Assume $G(y_1,\dots,y_m)$ and $H(w,x,\vec{y})$ are both in $\mathcal{R}'$. Then so is the function $F(x,\vec{y})$ defined by*

$$\begin{cases} F(0,\vec{y})=G(\vec{y}),\\ F(x+1,\vec{y})=H(F(x,\vec{y}),x,\vec{y}).\end{cases}$$

The idea behind the proof is to see that we can explicitly define F in terms of sequences as follows:

> $F(x, \vec{y}) = z$ if and only if there is a sequence $s_0, \ldots, s_x$ of length $x+1$ such that $s_0 = G(\vec{y}), s_x = z$, and $(\forall i)_{<x}[s_{i+1} = H(s_i, i, \vec{y})]$.

Of course, our treatment of sequence numbers in the previous section uses $R2$, so we can't use them here. Gödel's technical result was to define a function in $\mathcal{R}'$ which enables one to code sequences.

LEMMA 2.3. *There is a function $\beta(a, i)$ in $\mathcal{R}'$ such that for all $n \in \omega$ and for all $a_0, \ldots, a_n$ there exists a such that $(\forall i)_{\leq n}[\beta(a, i) = a_i]$.*

PROOF OF PROPOSITION FROM LEMMA. The (English) explicit definition of F above can be written using the β-function as: $F(x, \vec{y}) = \beta((\mu a)[\beta(a, 0) = G(\vec{y}) \wedge (\forall i)_{<x}\beta(a, i+1) = H(\beta(a, i), i, \vec{y})], x)$. □

Set-theoretically, a finite sequence $(a_0, \ldots, a_n)$ is a set of ordered pairs $\{(0, a_0), (1, a_1), \ldots, (n, a_n)\}$. What we want to do is *code* such a set by a single number a in such a manner that the uncoding function β is in $\mathcal{R}'$. We will break this into two parts—first, code ordered pairs of numbers by numbers; second, code finite sets of numbers by numbers.

Coding ordered pairs is very easy.

DEFINITION 2.1. $\mathrm{OP}(a, b) = (a+b)^2 + a + 1$.

Fact 0. *If $\mathrm{OP}(a, b) = \mathrm{OP}(a', b')$ then $a = a'$ and $b = b'$.*

PROOF. If $a + b < a' + b'$ then we have $\mathrm{OP}(a, b) \leq (a+b+1)^2 < \mathrm{OP}(a', b')$. Hence $\mathrm{OP}(a, b) = \mathrm{OP}(a', b')$ implies $a + b = a' + b'$, which then implies $a = a'$ and thus $b = b'$. □

Fact 1. *$a, b < \mathrm{OP}(a, b)$.*

So our problem is reduced to coding finite sets of numbers. What we will actually do is, for each number m, code sets of non-zero *numbers less than m* by a number. Our code for a finite set will then be an ordered pair, the first coordinate giving a number m bounding all numbers in the set and the second coordinate giving the code of the set relative to the bound m.

We will use the following elementary fact:

Fact 2. *Assume that k_i, l_j are relatively prime for all $0 \leq i \leq n, 0 \leq j \leq m$. Then $\exists c[(\forall i)_{\leq n} k_i | c \wedge (\forall j)_{\leq m}\, l_j \nmid c]$.*

PROOF. Take $c = k_0 \cdot k_1 \cdot \;\cdots\; \cdot k_n$. □

Fact 3. *There is a function $\rho(x, y)$ in $\mathcal{R}'$ such that for every m and for all i, j with $0 < i < j < m$, $\rho(m, i)$ and $\rho(m, j)$ are relatively prime.*

PROOF. First note that $(1 + iz)$ and $(1 + jz)$ are relatively prime provided $0 < i < j$ and $(j - i)|z$. Thus we want $z(m)$ to be divisible by every such $(j - i)$, which can be guaranteed by defining

$$z(m) = (\mu w)[(\forall k)_{<m}[k \neq 0 \to k|w] \wedge w \neq 0],$$

which is in $\mathcal{R}'$. Finally then $\rho(m,i) = 1 + i \cdot z(m)$ works. $\square$

Thus, given a set $S \subseteq \{x : x < m\}$ we define the code of S *relative to* m as:

$$c = \prod_{x \in S} \rho(m,x).$$

By Facts 2 and 3 we see that if $x < m$ then $x \in S$ if and only if $\rho(m,x)|c$. Next, given any finite $S \subseteq \omega$, a code of S is a number $a = \mathrm{OP}(m,c)$ where m is a number larger than every number in S and c is the code of S relative to m. Therefore, $x \in S$ if and only if $x < m$ and $\rho(m,x)|c$. If the set S we start with is a set of ordered pairs of a finite sequence, that is, $S = \{\mathrm{OP}(i,a_i) : i \leq n\}$, then the resulting a is the code of the finite sequence. Finally, we see that decoding the number assigned in this way to a finite sequence is done by β defined in the following manner:

Definition 2.2. $\beta(a,i) = (\mu x)_{<a}(\exists m)_{<a}(\exists c)_{<a}[a = \mathrm{OP}(m,c) \wedge \rho(m,\mathrm{OP}(i,x))|c]$.

This then completes the proof of Lemma 2.3 establishing the existence of the β-function.

The careful reader will have noted that there was some redundancy in our definition of recursive functions in the previous section, since $+$ and $\cdot$ can be defined from the starting functions using $R2$ (and $R1$ of course). We need them in our starting set, however, if we don't have $R2$, in particular to obtain the result of this section.

3. Representability of Recursive Functions

As we pointed out previously, the result of the preceding section makes it almost obvious that all recursive functions are definable in $\mathfrak{N}$. What we will prove in this section is a stronger result, which will yield improvements in the Incompleteness Theorem. Roughly speaking, we will prove that the recursive functions are definable in *every* model of Q, and that this in fact *characterizes* the recursive functions.

In this section, all languages will include at least $\overline{0}$ and s, so that the *numerals* $\overline{n}$ for $n \in \omega$ are all terms of the language. Our stronger notion of definability is then as follows:

Definition 3.1.

(a) Let $R \subseteq \omega^n$. R is (strongly) *representable in T by* $\phi(x_0,\dots,x_{n-1})$ if and only if for all $k_0,\dots,k_{n-1} \in \omega$ we have

$$R(k_0,\dots,k_{n-1}) \text{ holds } \Rightarrow T \vdash \phi(\overline{k}_0,\dots,\overline{k}_{n-1}),$$

and

$$R(k_0,\dots,k_{n-1}) \text{ fails } \Rightarrow T \vdash \neg\phi(\overline{k}_0,\dots,\overline{k}_{n-1}).$$

(b) Let $F : \omega^n \to \omega$. F is *representable in T by* $\phi(x_0,\dots,x_{n-1},y)$ if and only if for all $k_0,\dots,k_{n-1} \in \omega$ we have $T \vdash \forall y[\phi(\overline{k}_0,\dots,\overline{k}_{n-1},y) \leftrightarrow y \equiv \overline{m}]$ for $m = F(k_0,\dots,k_{n-1})$.

We will say that R (or F) is representable in T to mean that R (or F) is representable in T by some formula of the language of T. Note that, unless T is complete, the definition in (a) is stronger than the requirement that $R(k_0, \ldots, k_{n-1})$ holds if and only if $T \vdash \phi(\overline{k}_0, \ldots, \overline{k}_{n-1})$. The following lemma is easily established and thus left to the reader:

LEMMA 3.1.

(i) *If R (or F) is representable in T_1 by ϕ and $T_1 \subseteq T_2$ then R (or F) is also representable by ϕ in T_2.*

(ii) *Assume that $T \vdash \neg\overline{0} \equiv \overline{1}$. Then R is representable in T if and only if K_R is.*

(iii) *Assume that R is representable in T by ϕ. Let $\mathfrak{A} \models T$. Then for all $k_0, \ldots, k_{n-1} \in \omega, R(k_0, \ldots, k_{n-1})$ holds if and only if $\mathfrak{A} \models \phi(\overline{k}_0, \ldots, \overline{k}_{n-1})$.*

A statement analogous to (iii) for functions also holds. In addition, the converse to (iii) holds. The reader should note that the (stronger) requirement in (a) is used in establishing each part of the lemma. If T is inconsistent then every relation and function is representable in T. Otherwise, of course, only countably many relations and functions can be representable in T.

Our goal is to show that a function is recursive if and only if it is representable in Q, and that Q could be replaced by any consistent recursively axiomatized T such that $Q \subseteq T$ (where T might possibly be in a larger language). To accomplish this we must establish that a large number of sentences (true on $\mathfrak{N}$) are in fact provable from Q. The following definition allows us to be precise in the description of the sort of consequences of Q we will need.

DEFINITION 3.2. The set of Σ-*formulas* of $\mathcal{L}$ is defined as follows:

(i) Every atomic and negated atomic formula is a Σ-formula.

(ii) If ϕ, ψ are Σ-formulas then so are $(\phi \wedge \psi), (\phi \vee \psi)$.

(iii) If ϕ is a Σ-formula then so are $\exists x\phi$ and $\forall x(x < y \to \phi)$.

A Σ-*sentence* is then a Σ-formula with no free variables.

THEOREM 3.2. *Let σ be a Σ-sentence (of the language of $\mathfrak{N}$). Then $\mathfrak{N} \models \sigma$ if and only if $Q \vdash \sigma$.*

We will prove the corresponding statement about Σ-formulas by induction. The base case (atomic formulas) would be tedious done directly because of the complexity of terms allowed. We call an atomic formula *primitive* if it has one of the following forms: $x \equiv \overline{0}, x \equiv y, x < y, sx \equiv y, x + y \equiv z, x \cdot y \equiv z$, where x, y, z are variables.

LEMMA 3.3. *For any Σ-formula $\phi(x_0, \ldots, x_{n-1})$ there is a Σ-formula $\phi^*(x_0, \ldots, x_{n-1})$ containing only primitive atomic formulas such that $\phi \models\dashv \phi^*$.*

The idea of the proof is to note, for example, that $s(x+y) \equiv z$ is equivalent to $\exists u(x + y \equiv u \wedge su \equiv z)$. We leave the proof to the reader who, if unwilling to do this, could just change the definition of Σ-formula so that all the atomic formulas are primitive.

PROOF OF THEOREM 3.2. We prove, by induction on the collection of Σ-formulas built using only primitive atomic formulas, that for such $\phi(x_0, \ldots, x_{n-1})$ and any $k_0, \ldots, k_{n-1} \in \omega$, if $\mathfrak{N} \models \phi(\overline{k}_0, \ldots, \overline{k}_{n-1})$ then $Q \vdash \phi(\overline{k}_0, \ldots, \overline{k}_{n-1})$. [Note, of course, that $Q \vdash \sigma$ implies $\mathfrak{N} \models \sigma$ for any σ since $\mathfrak{N} \models Q$.] There are ten base cases for $x \equiv y, x < y, sx \equiv y, x + y \equiv z, x \cdot y \equiv 0$ *and their negations.*

(i) If $\mathfrak{N} \models \overline{k} \equiv \overline{l}$ then $k = l$ hence $\overline{k}$ is $\overline{l}$ hence $\vdash \overline{k} \equiv \overline{l}$ (since $x \equiv x \in \Lambda$).

(ii) If $\mathfrak{N} \models \neg\overline{k} \equiv \overline{l}$ then $k \neq l$. Since $\vdash (x \equiv y \to y \equiv x)$ we may suppose without loss of generality that $k > l$. We will prove by induction on l that for every l and for every k, if $k > l$ then $Q \vdash \neg\overline{k} \equiv \overline{l}$. Suppose $l = 0$. Then $k = n + 1$ for some n, so $\overline{k}$ is $s\overline{n}$. But $Q \vdash \neg s\overline{n} \equiv \overline{0}$ by axiom 1 of Q (page 11), i.e., $Q \vdash \neg\overline{k} \equiv \overline{l}$. Knowing the result for l, we prove it for $l+1$. Once again, if $k > l+1$ then $k = n+1$ for some $n > l$. By inductive hypothesis, $Q \vdash \neg\overline{n} \equiv \overline{l}$. But $Q \vdash (s\overline{n} \equiv s\overline{l} \to \overline{n} \equiv \overline{l})$ by axiom 2, so $Q \vdash \neg s\overline{n} \equiv s\overline{l}$, i.e., $Q \vdash \neg\overline{k} \equiv \overline{l}$ as required.

(iii) If $\mathfrak{N} \models s\overline{n} \equiv \overline{l}$ then $n + 1 = l$ so $s\overline{n}$ *is* $\overline{l}$, so $\vdash s\overline{n} \equiv \overline{l}$.

(iv) If $\mathfrak{N} \models \neg s\overline{n} \equiv \overline{l}$ then $n+1 \neq l$ so by (ii) $Q \vdash \neg\overline{n+1} \equiv \overline{l}$, i.e., $Q \vdash \neg s\overline{n} \equiv \overline{l}$ since $\overline{n+1}$ is $s\overline{n}$.

(v)–(viii) The cases of $x + y \equiv z, \neg x + y \equiv z, x \cdot y \equiv z, \neg x \cdot y \equiv z$ are left to the reader. You will use induction for the atomic cases and axioms 3–6 of Q.

(ix)–(x) We show simultaneously by induction on l that for all k, l if $k < l$ then $Q \vdash \overline{k} < \overline{l}$ and if $k \not< l$ then $Q \vdash \neg\overline{k} < \overline{l}$. This uses axioms 7 and 8.

The inductive steps for $\wedge, \vee$, and $\exists$ are easy and left to the reader (note, of course, that we use the fact that the universe of $\mathfrak{N}$ is $\{\overline{k}^{\mathfrak{N}} : k \in \omega\}$ in $\exists$-step). The step for bounded universal quantification is harder. Assume the inductive hypothesis for $\phi(x, y, z_1, \dots, z_n)$ and let $\psi(y, \overline{z})$ be $\forall x(x < y \to \phi)$. Given $k, k_1, \dots, k_n \in \omega$ suppose that $\mathfrak{N} \models \psi(\overline{k}, \overline{k}_1, \dots, \overline{k}_n)$. Then $\mathfrak{N} \models \phi(\overline{l}, \overline{k}, \overline{k}_1, \dots, \overline{k}_n)$ for every $l < k$, hence by inductive hypothesis, $Q \vdash \phi(\overline{l}, \overline{k}, \overline{k}_1, \dots, \overline{k}_n)$ for every $l < k$. To conclude that $Q \vdash \forall x(x < \overline{k} \to \phi(x, \overline{k}, \overline{k}_1, \dots \overline{k}_n))$ we need to establish that, for every k, we have $Q \vdash \forall x(x < \overline{k} \to x \equiv \overline{0} \vee x \equiv \overline{1} \vee \cdots \vee x \equiv \overline{k-1})$. This is easily established by induction on k, again using axioms 7 and 8 of Q. $\square$

Finally, the representability result we are after is as follows:

THEOREM 3.4. *Let T be a consistent, recursively axiomatized theory containing Q. Let $F \colon \omega^n \to \omega$ be given. Then the following are equivalent:*

(1) F is recursive.
(2) F is representable in Q by a Σ-formula.
(3) F is representable in T.

PROOF. (1) $\Rightarrow$ (2) By the results of the preceding section it suffices to show that the functions in $\mathcal{S}$ are representable in Q by Σ-formulas and that $R1$ and $R3$ preserve this property. We claim that $K_<$ is represented by $(x_0 < x_1 \wedge y \equiv \overline{0}) \vee (\neg x_0 < x_1 \wedge y \equiv \overline{1})$, that $s, +, \cdot$ are represented by $sx_0 \equiv y, x_0 + x_1 \equiv y$, and $x_0 \cdot x_1 \equiv y$, respectively, and that P^n_i is represented by $x_i \equiv y$ (considered as a formula in $x_0, \dots, x_n, y$). The verifications are straightforward using Theorem 3.2—note, however, that the directions $Q \vdash \forall y[\phi(\overline{k}_0, \dots, \overline{k}_{n-1}, y) \to y \equiv \overline{m}]$ use the fact (*not* stated by a Σ-formula) that in each of the above, we in fact have that $\forall x_0 \dots \forall x_n \forall y \forall y'[\phi(\vec{x}, y) \wedge \phi(\vec{x}, y') \to y \equiv y']$ is logically valid.

To show $R1$ preserves representability, suppose G is representable in Q by $\phi(x_1, \dots, x_k, z)$ and H_i is representable in Q by $\psi_i(y_1, \dots, y_m, x_i)$ for each $i = 1, \dots, k$. We then claim that $F(\vec{y}) = G(H_i(\vec{y}), \dots, H_k(\vec{y}))$ is representable in Q by the formula $\chi(y_1, \dots, y_m, z)$: $\exists x_1 \dots \exists x_k[\phi(\vec{x}, z) \wedge \psi_1(\vec{y}, x_1) \wedge \dots \wedge \psi_k(\vec{y}, x_k)]$.

Thus given $k_1, \ldots, k_m \in \omega$ let $l_i = H_i(k_1, \ldots, k_m)$ and let $n = G(l_1, \ldots, l_k)$. Then $Q \vdash \phi(\bar{l}_1, \ldots \bar{l}_k, \overline{n})$ and $Q \vdash \psi_i(\overline{k}_1, \ldots \overline{k}_m, \bar{l}_i)$, hence $Q \vdash \chi(\overline{k}_1, \ldots, \overline{k}_m, \overline{n})$. Further, $Q \vdash \chi(\overline{k}_1, \ldots, \overline{k}_m, z) \to \psi_i(\overline{k}_1, \ldots, \overline{k}_m, \bar{l}_i)$ and $Q \vdash \phi(\bar{l}_1, \ldots, \bar{l}_k, z) \to z \equiv \overline{n}$, hence $Q \vdash \chi(\overline{k}_1, \ldots, \overline{k}_m, z) \to z \equiv \overline{n}$. Thus χ does represent F in Q, and χ is a Σ-formula provided $\phi, \psi_1, \ldots, \psi_k$ are Σ-formulas.

Finally we need to prove that $R3$ preserves representability. Suppose G is represented in Q by $\psi(y_1, \ldots, y_m, x, z)$, and let $F(\vec{y}) = (\mu x)[G(\vec{y}, x) = 0]$. We claim that F is represented in Q by $\phi(y_1, \ldots, y_m,\ x)\colon \psi(y_1, \ldots, y_m, x, \overline{0}) \wedge \forall u[u < x \to \exists v(\neg v \equiv \overline{0} \wedge \psi(y_1, \ldots, y_m, u, v))]$. First of all, if $F(k_1, \ldots, k_m) = n$ then $Q \vdash \phi(\overline{k}_1, \ldots, \overline{k}_m, \overline{n})$, since ψ represents G *and* since we know that $Q \vdash \forall u(u < \overline{n} \to u \equiv \overline{0} \vee \ldots \vee u \equiv \overline{n-1})$. Now we need to establish that $Q \vdash [\phi(\overline{k}_1, \ldots, \overline{k}_m, x) \to x \equiv \overline{n}]$ for $n = F(k_1, \ldots, k_m)$. Here we finally use axiom 9 of Q, which yields $Q \vdash (x \not\equiv \overline{n} \to x < \overline{n} \vee \overline{n} < x)$. We have $Q \vdash (x < \overline{n} \to \neg\psi(\overline{k}_1, \ldots, \overline{k}_m, x, \overline{0}))$, and $Q \vdash \overline{n} < x \to \exists u(u < x \wedge \neg\exists v[\neg v \equiv \overline{0} \wedge \psi(\overline{k}_1, \ldots, \overline{k}_m, u, v)])$, so we can conclude that ϕ represents F. And, of course, ϕ is a Σ-formula provided ψ is.

(2) $\Rightarrow$ (3) is immediate by Lemma 3.1.

(3) $\Rightarrow$ (1). Let T be consistent and be axiomatized by the recursive set Σ. Suppose F is representable in T by $\phi(x_0, \ldots, x_{n-1}, y)$. Then $F(k_0, \ldots, k_{n-1}) = l$ if and only if $\Sigma \vdash \phi(\overline{k}_0, \ldots, \overline{k}_{n-1}, \bar{l})$, since T is consistent. We therefore can define $F(k_0, \ldots, k_{n-1})$ as the least l such that $\Sigma \vdash \phi(\overline{k}_0, \ldots, \overline{k}_{n-1}, \bar{l})$; we need to see that this is recursive. Well, we have $F(k_1, \ldots, k_n) = ((\mu w)[\mathrm{Prv}_\Sigma((w)_0, \ulcorner\phi(\overline{k}_0, \ldots, \overline{k}_{n-1}, \overline{(\omega)_1})\urcorner)])_1$, which is recursive since the function $G(k_0, \ldots, k_{n-1}, m) = \ulcorner\phi(\overline{k}_0, \ldots, \overline{k}_{n-1}, \overline{m})\urcorner$ is recursive since it is obtained by $(n+1)$ iterations of the function *Sub* defined in Definition 1.16. $\square$

Note that for the direction (3) $\Rightarrow$ (1) we do not need to assume $Q \subseteq T$, and T might be in a larger language than Q; we do need $T \models \overline{k} \not\equiv \overline{n}$ if $k \neq n$.

As a consequence we also obtain the same equivalence for recursive relations, using the lemma and the observation that R is representable by as Σ-formula provided K_R is representable by a Σ-formula.

4. Gödel's Incompleteness Theorem

We can now quickly derive Gödel's Theorem following our proof number two from Section 3.

4.1. Incompleteness Theorem (Gödel 1931). *Assume that T is recursively axiomatized and $\mathfrak{N} \models T$. Then there is a sentence σ, the negation of a Σ-sentence, such that $\mathfrak{N} \models \sigma$ but $T \nvdash \sigma$.*

Proof. We may suppose that $Q \subseteq T$, since otherwise we could take as σ one of the axioms of Q. Let Σ be a recursive set of axioms for T. The following relation is then recursive:

$$\mathrm{Prv}_\Sigma(z, \mathrm{Sub}(v_0, \ulcorner v_0\urcorner, \mathrm{Num}(v_0))),$$

hence represented in Q (and also T) by some Σ-formula $\theta(v_0, z)$. Now let $\phi(v_0)$ be $\neg\exists z\theta(v_0, z)$, let $k = \ulcorner\phi\urcorner$ and let σ be $\phi(\overline{k})$. Note that σ is the negation of the

Σ-sentence $\exists z\theta(\overline{k}, z)$. And the following are equivalent:

$$\begin{gathered}\mathfrak{N} \models \sigma,\\ \neg(\exists n \mathfrak{N} \models \theta(\overline{k}, \overline{n})),\\ \neg[\exists n \mathrm{Prv}_\Sigma(\overline{n}, \mathrm{Sub}(k, \ulcorner v_0 \urcorner, \mathrm{Num}(k)))],\\ \Sigma \nvdash \sigma,\end{gathered}$$

since $\ulcorner\sigma\urcorner = \mathrm{Sub}(\ulcorner\phi\urcorner, \ulcorner v_0\urcorner, \mathrm{Num}(\ulcorner\phi\urcorner))$. Since $\mathfrak{N} \models \Sigma$ we must have $\Sigma \nvdash \sigma$, and $\mathfrak{N} \models \sigma$, whence also $T \nvdash \sigma$. □

Corollary 4.2. *$Th(\mathfrak{N})$ is not recursively axiomatizable.*

This theorem should be contrasted with Theorem 3.2, which asserts that every Σ-sentence true on $\mathfrak{N}$ is a consequence of Q. In fact, even the set of Σ-sentences true on $\mathfrak{N}$ is more complicated than might be thought.

Corollary 4.3. *$\{\theta \colon \mathfrak{N} \models \theta, \theta$ is a Σ-sentence$\}$ is not recursive.*

Proof. the set of all Σ-sentences is easily shown to be recursive, so if the above mentioned set were recursive so would $\Sigma_0 = \{\theta \colon \mathfrak{N} \models \neg\theta, \theta \text{ is a } \Sigma\text{-sentence }\}$, hence also $\Sigma^* = \{\neg\theta \colon \theta \in \Sigma_0\}$. But then $T = \mathrm{Cn}(\Sigma^*)$ would contradict the statement of Gödel's Theorem. □

The only place in the proof of Theorem 4.1 that we used the assumption that $\mathfrak{N} \models T$ was in concluding that $\Sigma \nvdash \sigma$. We can eliminate this assumption by using the fact that $\neg\sigma$ is (equivalent to) a Σ-sentence, hence (for $T \supseteq Q$) we must have $T \vdash \neg\sigma$ provided $\mathfrak{N} \models \neg\sigma$. We thus obtain:

Theorem 4.4. *Assume that T is recursively axiomatized and consistent. Then there is a sentence σ, the negation of a Σ-sentence, such that $\mathfrak{N} \models \sigma$ but $T \nvdash \sigma$.*

[Of course, we might have $T \vdash \neg\sigma$.]

5. Church's Thesis

For our precise results of the previous section to have the full, intuitive force of the statements in Chapter **??**, we must accept that every intuitively effectively computable function on ω is in fact recursive. This assertion is known as Church's Thesis, after the logician who first explicitly enunciated it.

Church's Thesis can hardly be regarded as "obvious," since our definition of recursive is somewhat ad hoc. We spent enough effort showing that a relatively small number of functions are recursive—why should we accept that *all* computable functions are recursive?

On the other hand, Church's Thesis can hardly be susceptible to formal proof, since the intuitive concept of computable is necessarily informal and potentially ambiguous. Nevertheless, we can adduce several reasons for accepting Church's Thesis.

First, of course, is the lack of a counterexample. No one has been able to come up with a function on ω which is effectively computable but not recursive.

Second, and more important, is the fact that a number of other ways have been introduced to formalize the notion of effective computability—Turing machines, Markov algorithms, equationally derivable, etc. Although independently developed using different points of departure, these have all turned out to yield the same "computable" functions, namely the recursive functions.

A third argument is as follows. Suppose you are given an effectively computable function $F\colon \omega \to \omega$. That is, you are given a finite list of instructions which can be used to compute all the values of F. Surely this finite list of instructions must be formalizable in set theory of some sort (where, after all, *all* mathematics is carried out), and the axioms of some usual set theory must be strong enough to imply that the instructions yield the correct value. That is, there must be some formula $\phi(x, y)$ of set theory (allowing also $s, \overline{0}$) and some (recursive) set S^* of set theoretical axioms such that for every $k \in \omega$, if $m = F(k)$ then $S^* \vdash \phi(\overline{k}, \overline{m})$ and $S^* \vdash \forall y[\phi(\overline{k}, y) \to y \equiv \overline{m}]$. That is, F is representable in S^* by ϕ, hence F is recursive by Theorem 3.4 (remembering that the implication (3) $\Rightarrow$ (1) does not require the theory to contain the axioms of Q).

The presumption that the axioms of any reasonable set theory must be recursive seems clear—in practice you could only be given an axiomatization by finitely many schemas, which will end up being recursive.

In any case, henceforth we will accept Church's Thesis. It can be used to eliminate formal proofs that a function is recursive in favor of intuitive descriptions of a method to compute the function. Of course, there can be no *essential* applications of Church's Thesis, and we will normally not use it unless the alternative is horrendously complicated.

6. Primitive Recursive Functions

Functions of two arguments can be viewed as indexed families of functions of one argument analogously to relations. That is, given $F\colon \omega \times \omega \to \omega$ we define $F_k\colon \omega \to \omega$ by $F_k(x) = F(k, x)$ for all x, and we think of F as $\{F_k\}_{k\in\omega}$.

Our main goal in this section is to prove that there is a recursive function U of two arguments which "lists all primitive recursive functions of one argument"—that is, a function F of one argument is primitive recursive if and only if there is some k such that $F = U_k$.

Once we have such U we can define $F\colon \omega \to \omega$ by $F(n) = U(n, n) + 1$ for all n. Then F is recursive but $F \neq U_k$ for all k, hence F is not primitive recursive. Thus we have established $\mathcal{P} \subsetneq \mathcal{R}$.

To accomplish this we consider how we *define* primitive recursive functions. A sequence of steps showing exactly how some primitive recursive function F is obtained from the starting functions using $R1$ and $R2$ is called a *primitive recursive definition* of F. Of course the same primitive recursive function will have many different primitive recursive definitions, but each primitive recursive definition determines a unique function and gives complete instructions for computing its values.

We show how to *code* each primitive recursive definition by a sequence number in such a way that we can effectively obtain the values of a primitive recursive function from its code. This will enable us to obtain the desired recursive function U.

A sequence number coding a primitive recursive definition will be called a *primitive recursive index*, and we will use IPr for the set of all primitive recursive indices. Our definition will be such that the following hold for $f \in IPr$:

$(f)_1$ is the number of arguments of the function whose definition is coded by f;

$(f)_0 = 0$ if the coded definition just specifies some starting function;
$(f)_0 = 1$ if the last rule applied in the definition is $R1$;
$(f)_0 = 2$ if the last rule applied in the definition is $R2$;
if $(f)_0 = 0$ then the numbers $(f)_i$ for $i > 1$ specify which starting function;
if $(f)_0 = 1$ or 2 then the numbers $(f)_i$ for $i > 1$ are indices for the functions to which the last rule is applied.

The indices could be virtually anything "reasonable" (i.e., effective) satisfying the above guidelines. For the sake of definiteness we make the following choices:

the index of s is $\langle 0, 1, 0\rangle$,
the index of $+$ is $\langle 0, 2, 1\rangle$,
the index of $\cdot$ is $\langle 0, 2, 2\rangle$,
the index of $K_<$ is $\langle 0, 2, 3\rangle$,
the index of P_i^n for any $1 \le i \le n$ is $\langle 0, n, 4, i\rangle$.

If $g, h_1, \dots, h_k \in IPr$, corresponding to functions $G, H_1, \dots, H_k$ respectively, and if F is defined using $R1$ as

$$F(\vec{y}) = G(H_1(\vec{y}), \dots, H_k(\vec{y}))$$

then the primitive recursive index f of this definition of F is $\langle 1, n, g, h_1, \dots, h_k\rangle$ where n is the number of arguments of F (i.e., of each H_i). Note that $R1$ can actually be applied to G and $H_1, \dots, H_k$ to yield F (and so f actually is a primitive recursive index) if and only if $(g)_1 = k$ and for all $i = 1, \dots, k$ $(h_i)_1 = (f)_1$.

If $g, h \in IPr$ correspond to G and H, and if F is defined from them by $R2$, that is

$$\begin{cases} F(0, \vec{y}) = G(\vec{y}), \\ F(x+1, \vec{y}) = H(F(x, \vec{y}), x, \vec{y}), \end{cases}$$

then this definition of F has index $f = \langle 2, n+1, g, h\rangle$, where $n = (g)_1$. Furthermore, $R2$ can actually be applied, and so f is really an index if and only if $(h)_1 = (g)_1 + 2 = h + 2$.

The reader should check that the preceding paragraphs can be rewritten (without reference to the actual functions F, G, H, etc.) to yield a definition of IPr by course-of-values recursion, and thus IPr is primitive recursive.

Next define the "evaluation" relation $E \subseteq \omega \times \omega \times \omega$ by $E(f, m, l)$ holds if and only if $f \in IPr, \mathrm{lh}(m) = (f)_1$, and $F((m)_0, \dots, (m)_{n-1}) = l$, where $n = (f)_1$ and F is the function defined by the primitive recursive definition with index f.

E is certainly intuitively computable, since f codes the entire history of F and thus gives complete instructions for computing its values. The reader should check that we can easily give a formal definition showing that E is recursive.

Finally the function U which we are after can be defined by $U(k, x) = (\mu z)([k \in IPr \wedge (k)_1 = 1 \wedge E(k, \langle x\rangle, z)] \vee k \notin IPr \vee (k)_1 \neq 1)$. Thus U_k is the identically 0 function whenever k is not a primitive recursive index defining a function of one argument.

With U we can define other interesting recursive but not primitive recursive functions, for example, $G \colon \omega \to \omega$ with the property that whenever $H \colon \omega \to \omega$ is primitive recursive then there is some $k_0 \in \omega$ such that $G(x) > H(x)$ for all $x > k_0$.

Recursive but not primitive recursive functions can also be defined in a more mathematically "natural" fashion, but the proofs of non-primitive recursivity are not easy. One such function is $K\colon \omega\times\omega\to\omega$ determined by

$$\begin{aligned}&K(n,0)=n+1,\\&K(0,m+1)=K(1,m),\\&K(n+1,m+1)=K(K(n,m+1),m).\end{aligned}$$

What happens if we try to mimic this process for *recursive* functions? Well a recursive definition allows use of $R3$, and thus we will need indices $f=\langle 3,\ldots\rangle$ to reflect this. But the corresponding E can't be recursive, else we would have a contradiction. The reader should figure out what precisely goes wrong.

7. Exercises

(1) Assume that X and Y are each the range of some recursive function on ω into ω. Prove that $(X\cap Y)$ is either empty or the range of some recursive function on ω.

(2) Let $X\subseteq\omega$. Assume that both X and $(\omega\setminus X)$ are each the range of some recursive function on ω. Prove that X is recursive.

(3) Let $R\subseteq\omega\times\omega$ be primitive recursive. Define $F:\omega\times\omega\to\omega$ by $F(k,l)=(\mu n)_{<l}[R(k,n)\text{ holds}]$. Prove that F is primitive recursive.

(4) Let $R\subseteq\omega\times\omega$ be recursive. Define

$$A=\{k\in\omega : R(k,l)\text{ holds for some } l\in\omega\}.$$

   (a) Assume that A is non-empty. Prove that A is the image of some recursive function $f:\omega\to\omega$.

   (b) Assume that A is infinite. Prove that A is the image of some $1-1$ recursive function $f:\omega\to\omega$.

CHAPTER 9

Undecidability and Further Developments

0. Introduction

This chapter continues the development of the techniques used to prove Gödel's Incompleteness Theorem. In Section 1 we present some generalizations and related results obtained in the mid-1930s by Rosser, Church, and Tarski. In Section 2, following the approach of Tarski, Mostowski, and R.M. Robinson, we finally obtain methods of showing the undecidability of theories which do not contain Q. In particular, we finally derive Church's celebrated solution to the decision problem for pure logic. In Section 3 we pause to present some of the fundamental properties of r.e. relations, in particular proving that they are precisely the relations definable in $\mathfrak{N}$ by Σ-formulas. In Section 4 we discuss the solution to Hilbert's tenth problem. The "technical" lemma needed for its solution states that in fact every r.e. relation is definable in $\mathfrak{N}$ by an *existential* formula. In Section 5 we give a "fixed point" theorem which is lurking behind the Incompleteness Theorem. We use this result to discuss Gödel's *second* Incompleteness Theorem which states that no consistent extension of P can prove its own consistency.

1. The Theorems of Church and Tarski

Recall that a theory T is *decidable* if and only if $\mathrm{Thm}_T = \{\ulcorner\sigma\urcorner : \sigma \in \mathrm{Sn}_{\mathcal{L}}, T \vdash \sigma\}$ is recursive. Gödel's Incompleteness Theorem implies that $Th(\mathfrak{N})$ is not decidable. A considerable strengthening of this, due essentially to Church, states that no consistent extension of Q is decidable.

The following notation will be helpful:

Definition 1.1. For any $R \subseteq \omega \times \omega$ and any $k \in \omega$, $R_k = \{n : R(k, n) \text{ holds}\}$.

We will frequently think of a binary relation R on ω as an indexed family of sets R_k, as defined above. Similarly, of course, an $(n+1)$-ary relation could be considered as an indexed family of n-ary relations.

The following, trivial lemma is the main component of a surprising number of sophisticated arguments:

1.1. Lemma (Diagonalization). *Given a binary relation R on ω, define Q by $Q(x) \Leftrightarrow \neg R(x, x)$. Then $Q \neq R_k$ for every k.*

Proof. If $Q = R_{k_0}$ then $Q(k_0)$ holds if and only if $\neg R(k_0, k_0)$ holds if and only if $R_{k_0}(k_0)$ fails, a contradiction. □

1.2. Church's Theorem. *Assume T is consistent and $Q \subseteq T$. Then T is not decidable.*

Proof. Suppose T were decidable, so Thm_T is recursive. Define R by

$$R(k,n) \Leftrightarrow \mathrm{Thm}_T(\mathrm{Sub}(k, \ulcorner v_0 \urcorner, \mathrm{Num}(n))),$$

that is, $R(k,n)$ holds if and only if $k = \ulcorner \phi \urcorner$ for some formula $\phi(v_0)$ and $T \vdash \phi(\overline{n})$. Then R is recursive, by our supposition.

Since $T \supseteq Q$ we know that every recursive set is representable in T, that is, if $X \subseteq \omega$ is recursive then there is some $\phi(v_0)$ such that $X = \{n \in \omega \colon T \vdash \phi(\overline{n})\}$. This means that every recursive $X \subseteq \omega$ is R_k for some k.

Finally define Q by $Q(x) \Leftrightarrow \neg R(x,x)$. Then Q is recursive, since R is. But $Q \neq R_k$ for all k by diagonalization, which contradicts the result of the previous paragraph. □

To derive (Rosser's improvement of) Gödel's Incompleteness Theorem from Church's result, we need to know that a complete recursively axiomatized theory is decidable. We gave an informal argument for this in Chapter **??**, but can now give a very neat, formal proof.

The following definition introduces one of the fundamental concepts of recursion theory:

Definition 1.2. $R \subseteq \omega^n$ is *recursively enumberable* (r.e.) if and only if there is some recursive $Q \subseteq \omega^{n+1}$ such that $R(\vec{x}) \Leftrightarrow \exists y\, Q(\vec{x}, y)$.

Clearly, a recursive relation is r.e., and we will soon see that the converse fails. The next important fact gives us one way of showing that an r.e. relation is in fact recursive.

Proposition 1.3. *R is recursive if and only if R and $\neg R$ are both r.e.*

Proof. We need just show the implication from right to left. Suppose R and $\neg R$ are both r.e. Then there are recursive relations Q, S such that

$$R(\vec{x}) \Leftrightarrow \exists y\, Q(\vec{x}, y)$$

and

$$\neg R(\vec{x}) \Leftrightarrow \exists y\, S(\vec{x}, y).$$

Since for any $\vec{x}$ either $R(\vec{x})$ or $\neg R(\vec{x})$ must hold, we see that

$$(\mu y)[Q(\vec{x}, y) \vee S(\vec{x}, y)]$$

is a recursive function. Since $R(\vec{x})$ and $\neg R(\vec{x})$ can't both hold we see that

$$R(\vec{x}) \Leftrightarrow Q(\vec{x}, (\mu y)[Q(\vec{x}, y) \vee S(\vec{x}, y)]),$$

hence R is recursive. □

Using the above result, the proof of the result on complete recursively axiomatizable theories is easy.

Theorem 1.4. *Assume T is complete and recursively axiomatizble. Then T is decidable.*

Proof. Let Σ be a recursive set of axioms for T. Then Thm_T is r.e. since

$$\mathrm{Thm}_T(x) \Leftrightarrow \exists z[\mathrm{Prv}_\Sigma(z,x) \wedge \mathrm{Sn}(x)].$$

But $\neg\mathrm{Thm}_T$ is also r.e., because $T = \mathrm{Cn}(\Sigma)$ is complete, and so

$$\neg\mathrm{Thm}_T(x) \Leftrightarrow \exists z[\mathrm{Prv}_\Sigma(z, \mathrm{Neg}(x)) \vee \neg\mathrm{Sn}(x)].$$

Thus T is decidable. □

Thus, Church's undecidability theorem yields as a consequence the following improvement, due to Rosser, of the Incompleteness Theorem:

THEOREM 1.5. *No recursively axiomatizable theory T with $Q \subseteq T$ is complete.*

Analyzing the proof of Church's Theorem we see that the only use of the assumption that T is decidable is to conclude that Thm_T is representable in T (since the functions representable in T are closed under composition—cf. page 26). Using this observation we are able to derive Tarski's well-known result about the undefinability of truth. The details are left to the reader.

1.6. Theorem (Tarski). *Let $\mathfrak{A} \models Q$. Then there is no formula $\theta(x)$ such that for every sentence σ we have $\mathfrak{A} \models \sigma$ if and only if $\mathfrak{A} \models \theta(\overline{\ulcorner \sigma \urcorner})$.*

A variant of this states:

THEOREM 1.7. *Let T be consistent, $Q \subseteq T$. Then there is no formula $\theta(x)$ such that for every sentence σ we have $T \vdash [\sigma \leftrightarrow \theta(\overline{\ulcorner \sigma \urcorner})]$.*

Finally, the following result graphically illustrates the way in which completeness fails:

THEOREM 1.8. *Let T be a consistent, recursively axiomatized theory, $T \supseteq Q$. Then there is some $\theta(y)$, a negation of a Σ-formula, such that $T \vdash \theta(\overline{k})$ for every $k \in \omega$ but $T \nvdash \forall y\, \theta(y)$.*

PROOF. We know there is some r.e. set X which is not recursive. Then there is some recursive R such that $X = \{n : \exists y\, R(n, y) \text{ holds}\}$. R is represented in T by some $\phi(x, y)$. Therefore, $n \in X$ implies $T \vdash \exists y\, \phi(\overline{n}, y)$. Since X is *not* recursive, there is some $n_0 \notin X$ such that $T \nvdash \neg\exists y\, \phi(\overline{n}_0, y)$ even though $T \vdash \neg\phi(\overline{n}_0, \overline{k})$ for every k. Thus, we can take $\theta(y) = \neg\phi(\overline{n}_0, y)$. $\square$

If $\mathfrak{N} \models T$, then the above yields incompleteness.

2. Undecidable Theories

We now know that every consistent extension of Q is undecidable. In this section we present some general methods (due to Tarski, Mostowski, and R.M. Robinson) to show that theories, not containing the axioms of Q, are undecidable. In particular, we will derive Church's negative solution to the decision problem, that is, that pure logic, $\mathrm{Cn}_{\mathcal{L}}(\varnothing)$, is undecidable provided $\mathcal{L}$ contains at least a binary predicate symbol.

The methods we develop could be called "relative undecidability" results, since they will say that a theory T_2 is undecidable provided it is related in a certain way to a theory T_1 known to be undecidable. Repeated application of these methods, beginning with the particular theory Q, will enable us to conclude, for example, that the theory of fields, the theory of groups, and $Th((\mathbb{Q}, +, \cdot))$ are all undecidable, in addition to Church's result about pure logic mentioned above.

The theory Q has a stronger property that will enter into our considerations.

DEFINITION 2.1. A consistent theory T of $\mathcal{L}$ is *essentially undecidable* (EU) if and only if every consistent extension of T (in $\mathcal{L}$) is undecidable.

Examples show that an undecidable theory need not be essentially undecidable, but clearly any consistent extension of an essentially undecidable theory (in the same language) is again essentially undecidable.

Our first relative undecidability results concern the following sorts of extensions:

DEFINITION 2.2. Let T_i be a theory of $\mathcal{L}_i$, $i = 1, 2$.

(a) T_2 is a *finite extension* of T_1 if and only if $\mathcal{L}_1 = \mathcal{L}_2$ and $T_2 = \mathrm{Cn}(T_1 \cup \Sigma)$ for some finite set Σ of sentences.

(b) T_2 is a *conservative extension* of T_1 if and only if $\mathcal{L}_1 \subseteq \mathcal{L}_2$ and $T_1 = (T_2 \cap \mathrm{Sn}_{\mathcal{L}_1})$.

In particular, note that if Σ is a set of sentences of $\mathcal{L}_1$ and $\mathcal{L}_1 \subseteq \mathcal{L}_2$, then $\mathrm{Cn}_{\mathcal{L}_2}(\Sigma)$ is a conservative extension of $\mathrm{Cn}_{\mathcal{L}_1}(\Sigma)$.

THEOREM 2.1. *Given theories T_i of $\mathcal{L}_i$, $i = 1, 2$:*

(a) If T_2 is a finite extension of T_1 and T_2 is undecidable, then T_1 is undecidable.

(b) If T_2 is a conservative extension of T_1 and T_1 is undecidable, then T_2 is undecidable.

(c) If T_2 is a conservative extension of T_1 and T_1 is EU, then T_2 is EU.

PROOF. (a) Let $T_2 = \mathrm{Cn}(T_1 \cup \Sigma)$, where $\mathcal{L}_1 = \mathcal{L}_2$ and Σ is finite. Let σ be the conjunction of all the sentences in Σ. Then $T_2 \models \theta$ if and only if $T_1 \models (\sigma \to \theta)$, so we have $\mathrm{Thm}_{T_2}(x) \leftrightarrow \mathrm{Thm}_{T_1}(\ulcorner(\sigma \to\urcorner * x * \ulcorner)\urcorner)$, thus Thm_{T_2} is recursive provided Thm_{T_1} is.

(b) We immediately conclude that Thm_{T_1} is recursive if Thm_{T_2} is, since $\mathrm{Sn}_{\mathcal{L}_1}$ is recursive.

(c) Given a consistent extension T_2' of T_2, in $\mathcal{L}_2$, let $T_1' = (T_2' \cap \mathrm{Sn}_{\mathcal{L}_1})$. Then T_1' is undecidable since T_1 is EU, and T_2' is a conservative extension of T_1', hence undecidable by (b). □

COROLLARY 2.2. *If the theory T of $\mathcal{L}$ is EU, then for every $\mathcal{L}' \supseteq \mathcal{L}$ and every consistent theory $T' \supseteq T$ of $\mathcal{L}'$, T' is undecidable (in fact, EU).*

The finite axiomatizability of Q also enters into our development in an essential way.

THEOREM 2.3. *Assume T is a finitely axiomatizable EU theory of $\mathcal{L}$. Let T' be any theory of any $\mathcal{L}' \supseteq \mathcal{L}$ such that $(T \cup T')$ is consistent. Then T' is undecidable.*

PROOF. $\mathrm{Cn}_{\mathcal{L}'}(T \cup T')$ is undecidable by Corrolary 2.2, and it is a finite extension of T', since T is finitely axiomatizable, hence T' is undecidable by Theorem 2.1. □

COROLLARY 2.4. *If there is some finitely axiomatizable EU theory of $\mathcal{L}$, then for every $\mathcal{L}' \supseteq \mathcal{L}$, $\mathrm{Cn}_{\mathcal{L}'}(\varnothing)$ is undecidable. Hence $\mathrm{Cn}_{\mathcal{L}'}(\varnothing)$ is undecidable whenever $\mathcal{L}' \supseteq \{+, \cdot, <, s, \overline{0}\}$.*

We will find it more convenient to work with structures rather than theories. The sort of structures we are interested in are given in the following definition:

DEFINITION 2.3. An $\mathcal{L}$-structure $\mathfrak{A}$ is *strongly undecidable* if and only if there is some finitely axiomatizable EU theory T_0 of $\mathcal{L}$ such that $\mathfrak{A} \models T_0$.

Thus, $\mathfrak{N}$ is strongly undecidable. Theorem 2.3 immediately yields the following fact:

COROLLARY 2.5. *Assume $\mathfrak{A}$ is strongly undecidable. Let T be any theory of $\mathcal{L}$ (the language of $\mathfrak{A}$) such that $\mathfrak{A} \models T$. Then T is undecidable.*

Our goal is to establish certain "transfer" results for the property of strong undecidability. We will then use these results, beginning with the strongly undecidable structure $\mathfrak{N}$, to conclude that a number of theories of interest are undecidable.

We begin with a simple lemma.

LEMMA 2.6. *Let $\mathfrak{A} = \mathfrak{B} \upharpoonright \mathcal{L}_1$ where $\mathfrak{B}$ is an $\mathcal{L}_2$-structure.*

(a) If $\mathfrak{A}$ is strongly undecidable, so is $\mathfrak{B}$.
(b) If $\mathcal{L}_2 - \mathcal{L}_1 = \{c_0, \dots, c_n\}$ consists just of individual constants and if $\mathfrak{B}$ is strongly undecidable, then so is $\mathfrak{A}$.

PROOF. (a) This is clear from Theorem 2.1.

(b) Let T_2 be a finitely axiomatizable EU theory of $\mathcal{L}_2$ such that $\mathfrak{B} \models T_2$. Let Σ be a finite set of axioms for T_2 and let σ be the conjunction of the sentences in Σ. Then $\sigma = \phi(c_0, \dots, c_n)$ for some formula $\phi(x_0, \dots, x_n)$ of $\mathcal{L}_1$. Let $T_1 = \mathrm{Cn}_{\mathcal{L}_1}(\{\exists x_0 \dots \exists x_n\, \phi\})$. Then $\mathfrak{A} \models T_1$. We show that T_1 is EU. Let T be a consistent $\mathcal{L}_1$ theory, $T_1 \subseteq T$. Let $T^* = \mathrm{Cn}_{\mathcal{L}_2}(T)$. Then, for an arbitrary sentence θ of $\mathcal{L}_2$ we know that $\theta = \psi(c_0, \dots, c_n)$ for some $\psi(x_0, \dots, x_n)$ of $\mathcal{L}_1$ and can conclude that $T^* \models \theta$ if and only if $T \models \forall x_0 \dots \forall x_n \psi$. Thus T is undecidable provided T^* is. But $T^* \cup T_2$ is consistent (since $T \models \exists \vec{x}\, \phi$), hence T^* is undecidable by Theorem 2.3. Thus T_1 is EU. $\Box$

Our main results concern the following two relations between structures:

DEFINITION 2.4. Let $\mathfrak{A}$ be an $\mathcal{L}_1$-structure and let $\mathfrak{B}$ be an $\mathcal{L}_2$-structure, where $\mathcal{L}_1 \subseteq \mathcal{L}_2$.

(a) $\mathfrak{B}$ is an *expansion by definitions* of $\mathfrak{A}$ if and only if $\mathfrak{A} = \mathfrak{B} \upharpoonright \mathcal{L}_1$ and for every R (F, c) $\in \mathcal{L}_2 - \mathcal{L}_1$, $R^{\mathfrak{B}}$ ($F^{\mathfrak{B}}$, $\{c^{\mathfrak{B}}\}$) is definable in $\mathfrak{A}$ by an $\mathcal{L}_1$-formula (therefore also in $\mathfrak{B}$).
(b) $\mathfrak{A}$ is *definable in* $\mathfrak{B}$ if and only if $\mathfrak{A} = \mathfrak{B} \upharpoonright \mathcal{L}_1$ and A is definable in $\mathfrak{B}$ (by an $\mathcal{L}_2$-formula).

Our main transfer results are contained in the next theorem.

THEOREM 2.7. *Assume that $\mathfrak{B}$ is strongly undecidable and that either $\mathfrak{B}$ is an expansion by definitions of $\mathfrak{A}$ or $\mathfrak{B}$ is definable in $\mathfrak{A}$. Then $\mathfrak{A}$ is strongly undecidable.*

We prove this result via a sequence of lemmas. For the first part of the theorem we assume that $\mathcal{L}_1 \subseteq \mathcal{L}_2$, $\mathfrak{B}$ is a strongly undecidable $\mathcal{L}_2$-structure, $\mathfrak{A} = \mathfrak{B} \upharpoonright \mathcal{L}_1$ and $\mathfrak{B}$ is an expansion by definitions of $\mathfrak{A}$. For each R (or F or c) $\in \mathcal{L}_2 - \mathcal{L}_1$ we fix an $\mathcal{L}_1$-formula ϕ_R (or ϕ_F or ϕ_c) defining $R^{\mathfrak{B}}$ ($F^{\mathfrak{B}}$, $\{c^{\mathfrak{B}}\}$) in $\mathfrak{A}$. we let

$$\begin{aligned}\Delta &= \{\forall \vec{x}[R\vec{x} \leftrightarrow \phi_R(\vec{x})] \colon R \in \mathcal{L}_2 - \mathcal{L}_1\} \\ &\cup \{\forall \vec{x} \forall y[F\vec{x} \equiv y \leftrightarrow \phi_F(\vec{x}, y)] \colon F \in \mathcal{L}_2 - \mathcal{L}_1\} \\ &\cup \{\forall x[x \equiv c \leftrightarrow \phi_c(x)] \colon c \in \mathcal{L}_2 - \mathcal{L}_1\}.\end{aligned}$$

We further define Γ to be

$$\{\forall \vec{x}\, \exists! y \phi_F(\vec{x}, y) \colon F \in \mathcal{L}_2 - \mathcal{L}_1\} \cup \{\exists! x \phi_c(x) \colon c \in \mathcal{L}_2 - \mathcal{L}_1\}.$$

Notice that Γ is a set of sentences of $\mathcal{L}_1$, $\mathfrak{A} \models \Gamma$, and $\mathfrak{B} \models \Delta$. For each $\mathcal{L}_2$-formula $\phi(\vec{x})$ we let $\phi^*(\vec{x})$ be the $\mathcal{L}_1$-formula obtained from ϕ by replacing all symbols of $\mathcal{L}_2 - \mathcal{L}_1$ by their $\mathcal{L}_1$-definitions.

If $\mathcal{L}_2 - \mathcal{L}_1$ consists only of relations, ϕ^* has a straightforward recursive definition. In the general case, ϕ must first be rewritten so that functions and constants of $\mathcal{L}_2 - \mathcal{L}_1$ occur only in contexts of $F\vec{x} \equiv y$ and $x \equiv c$. We leave further details to the reader. The basic properties of this translation are given in the following lemma:

LEMMA 2.8.

(1) $\Delta \models [\phi \leftrightarrow \phi^*]$ *for all* $\phi \in \mathrm{Fm}_{\mathcal{L}_2}$.
(2) *If* $\mathfrak{A}'$ *is any* $\mathcal{L}_1$*-structure then* $\mathfrak{A}' \models \Gamma$ *if and only if* $\mathfrak{A}' = \mathfrak{B}' \restriction \mathcal{L}_1$ *for some* $\mathfrak{B}' \models \Delta$.
(3) *If* $\Sigma \subseteq \mathrm{Sn}_{\mathcal{L}_2}$, *let* $\Sigma^* = \{\sigma^* : \sigma \in \Sigma\}$, *then* $\Delta \cup \Sigma$ *has a model if and only if* $\Gamma \cup \Sigma^*$ *has a model.*

PROOF. (1) and (2) are left to the reader.

(3) By compactness it suffices to consider the case where Σ is finite. Since $(\phi \wedge \psi)^* = (\phi^* \wedge \psi^*)$ it suffices to consider the case $\Sigma = \{\sigma\}$. By (2), $\Gamma \cup \{\sigma^*\}$ has a model if and only if $\Delta \cup \{\sigma^*\}$ has a model, and by (1), $\Delta \cup \{\sigma^*\}$ has a model if and only if $\Delta \cup \{\sigma\}$ has a model, so (3) holds. $\square$

Since $\mathfrak{B}$ is strongly undecidable we know $\mathfrak{B} \models T_2$ where $T_2 = \mathrm{Cn}_{\mathcal{L}_2}(\Sigma)$ is EU and Σ is finite. We define $T_1 = \mathrm{Cn}_{\mathcal{L}_1}(\Sigma^* \cup \Gamma)$. Then $\mathfrak{A} \models T_1$, by Lemma 2.8, and T_1 is finitely axiomatizable. (Note we are using here our blanket assumption in decidability contexts that all languages are finite.) Thus to show that $\mathfrak{A}$ is strongly undecidable it suffices to show that T_1 is EU.

Let T be a consistent $\mathcal{L}_1$-theory, $T_1 \subseteq T$. Let $T^* = \mathrm{Cn}_{\mathcal{L}_2}(T \cup \Delta)$. We establish:

LEMMA 2.9.

(3) *For every sentence* σ *of* $\mathcal{L}_2$, $T^* \models \sigma$ *if and only if* $T \models \sigma^*$.
(4) $T_2 \subseteq T^*$.
(5) T *is undecidable.*

PROOF. (3) From right to left is immediate by Lemma 2.8 (1). The other implication (in its contrapositive form) is immediate from Lemma 2.8 (3), since $(\neg\sigma)^*$ is $\neg\sigma^*$.

(4) If $\sigma \in T_2$ then $\Sigma \models \sigma$ hence $T_1 \models \sigma^*$ by Lemma 2.8 (3), and so $T^* \models \sigma^*$, and therefore $T^* \models \sigma$ by Lemma 2.8 (1).

(5) By (3), T^* is consistent, and so by (4) T^* is undecidable, hence by (3) again T is also undecidable (since the function sending ϕ to ϕ^* is recursive). $\square$

Thus the first part of Theorem 2.7 holds.

For the second part of Theorem 2.7 we assume that $\mathcal{L}_1 \subseteq \mathcal{L}_2$, $\mathfrak{B}$ is an $\mathcal{L}_1$-structure which is strongly undecidable and definable in the $\mathcal{L}_2$-structure $\mathfrak{A}$. We first note that we may assume that $B = P^{\mathfrak{A}}$, where P is some 1-ary predicate symbol in $\mathcal{L}_2 - \mathcal{L}_1$ (since otherwise we could replace $\mathcal{L}_2$ by $\mathcal{L}_2 \cup \{P\}$ and expand $\mathfrak{A}$ to $\mathfrak{A}'$ by defining $P^{\mathfrak{A}} = B = \phi^{\mathfrak{A}}$ for some ϕ; by the first part of Theorem 2.7 the strong undecidability of $\mathfrak{A}'$ implies the strong undecidability of $\mathfrak{A}$).

We next define:

DEFINITION 2.5. For arbitrary $\mathcal{L}_1$-formulas ϕ, the *relativization* $\phi^{(P)}$ *of* ϕ *to* P is given by the following recursion:

$$\begin{array}{rcl} \phi^{(P)} & = & \phi \text{ if } \phi \text{ is atomic,} \\ (\neg\phi)^{(P)} & = & \neg\phi^{(P)}, \\ (\phi \to \psi)^{(P)} & = & (\phi^{(P)} \to \psi^{(P)}), \\ (\forall x\phi)^{(P)} & = & \forall x[Px \to \phi]. \end{array}$$

We let Γ be the following finite set of $\mathcal{L}_2$-sentences:

$$\{\exists x Px\} \cup \{Pc\colon c \in \mathcal{L}_1\} \cup$$
$$\{\forall x_1 \ldots \forall x_n (Px_1 \wedge \ldots \wedge Px_n \to PFx_1, \ldots, x_n)\colon F \in \mathcal{L}_1\}.$$

The following results can then be established as for the first part of Theorem 2.7:

LEMMA 2.10.

(1) For any $\mathcal{L}_2$-structure $\mathfrak{A}'$, $\mathfrak{A}' \models \Gamma$ if and only if there is some $\mathfrak{B}' \subseteq \mathfrak{A}' \restriction \mathcal{L}_1$ with $B' = P^{\mathfrak{A}'}$.

(2) Given $\mathfrak{A}'$, $\mathfrak{B}'$ as in (1)*, for any $\mathcal{L}_1$-sentence σ we have $\mathfrak{B}' \models \sigma$ if and only if $\mathfrak{A} \models \sigma^{(P)}$.*

(3) If $\Sigma \subseteq \mathrm{Sn}_{\mathcal{L}_2}$ let $\Sigma^* = \{\sigma\colon \sigma^{(P)} \in \Sigma\}$. Then Σ^* has a model if and only if $(\Sigma \cup \Gamma)$ has a model.

Now let T_1 be a finitely axiomatizable EU theory such that $\mathfrak{B} \models T_1$, say $T_1 = \mathrm{Cn}(\Sigma)$ where Σ is finite. We define

$$T_2 = \mathrm{Cn}_{\mathcal{L}_2}(\Sigma^{(P)} \cup \Gamma).$$

Then T_2 is finitely axiomatizable and $\mathfrak{A} \models T_2$, by Lemma 2.10 (1) and (2). To show T_2 is EU, let T be a consistent L_2-theory, $T_2 \subseteq T$. Define

$$T^* = \mathrm{Cn}_{\mathcal{L}_1}(\{\sigma \in \mathrm{Sn}_{\mathcal{L}_1}\colon \sigma^{(P)} \in T\}).$$

We then have, as before, the following:

LEMMA 2.11.

(3) For any $\theta \in \mathrm{Sn}_{\mathcal{L}_1}$, $T^* \models \theta$ if and only if $T \models \theta^{(P)}$.

(4) $T_1 \subseteq T^*$.

(5) T is undecidable.

Further details are left to the reader.

We list some of the consequences of Theorem 2.7:

Fact 4. $(\omega, +, \cdot)$ *is strongly undecidable, hence* $\mathrm{Cn}_{\mathcal{L}}(\varnothing)$ *is undecidable whenever* $\mathcal{L}$ *contains at least two* 2*-ary function symbols.*

Fact 5. $(\mathbf{Z}, +, \cdot)$ *is strongly undecidable, hence the elementary theory of rings (i.e., the consequences of the ring axioms in this language, or in the language adding 0 or also 1) is undecidable.*

Fact 6. $(\mathbf{Q}, +, \cdot)$ *is strongly undecidable, hence the elementary theory of fields is undecidable.*

We prove Fact 4 by noting that $\mathfrak{N} = (\omega, +, \cdot, <, s, 0)$ is an expansion by definitions of $(\omega, +, \cdot)$. To prove Fact 5 we show that $(\omega, +, \cdot)$ is definable in $(\mathbb{Z}, +, \cdot)$—this follows from the well-known number theoretic fact that if $k \in \mathbb{Z}$ then $k \geq 0$ if and only if there are $n_1, n_2, n_3, n_4 \in \mathbb{Z}$ such that $k = n_1^2 + n_2^2 + n_3^2 + n_4^2$. A less familiar fact shows that $(\mathbb{Z}, +, \cdot)$ is definable in $(\mathbb{Q}, +, \cdot)$.

Our next goal is to show there is a strongly undecidable structure for the language with just a 2-ary predicate. To do this we require the following:

LEMMA 2.12. *Assume that $\mathcal{L}_2 = \mathcal{L}_1 \cup \{c_1, \ldots, c_n\}$ where $c_1, \ldots, c_n$ are individual constants. Let $\mathfrak{B}$ be an $\mathcal{L}_2$-structure and let $\mathfrak{A} = \mathfrak{B} \restriction \mathcal{L}_1$. If $\mathfrak{B}$ is strongly undecidable then $\mathfrak{A}$ is also strongly undecidable.*

PROOF. Assume that $\mathfrak{B} \models T_2$ where T_2 is EU and $T_2 = \mathrm{Cn}(\Sigma)$ for finite Σ. If σ is any sentence of $\mathcal{L}_2$ then σ is $\phi(c_1, \ldots, c_n)$ for some formula $\phi(x_1, \ldots, x_n)$ of $\mathcal{L}_1$; we let σ^* be the $\mathcal{L}_1$-sentence $\forall x_1 \ldots \forall x_n \phi$. Note that for any theory T of $\mathcal{L}_1$ and any $\sigma \in \mathrm{Sn}_{\mathcal{L}_2}$, $T \models \sigma$ if and only if $T \models \sigma^*$—hence T is undecidable if (and only if) $T' = \mathrm{Cn}_{\mathcal{L}_2}(T)$ is undecidable.

Now let $\theta(c_1, \ldots, c_n)$ be the conjunction of the sentences in Σ, let $\theta^{\#}$ be the $\mathcal{L}_1$-sentence $\exists x_1 \ldots \exists x_n \theta(\vec{x})$, and let $T_1 = \mathrm{Cn}_{\mathcal{L}_1}(\theta^{\#})$. Then $\mathfrak{A} \models T_1$. We claim that T_1 is EU. Let $T \supseteq T_1$ be any consistent $\mathcal{L}_1$-theory and let $T' \subseteq \mathrm{Cn}_{\mathcal{L}_2}(T)$. Then $(T' \cup T_2)$ is consistent, hence T' is undecidable, by Theorem 2.3, and so T is undecidable by the remark in the first paragraph of this proof. $\square$

THEOREM 2.13. *There is a strongly undecidable $\mathcal{L}$-structure where $\mathcal{L}$ is the language whose only non-logical symbol is a 2-ary predicate R.*

PROOF. Given $\mathcal{L}$ as in the statement of the theorem we let $\mathcal{L}^+ = \mathcal{L} \cup \{\overline{0}, \overline{1}, \overline{u}\}$, where $\overline{0}, \overline{1}, \overline{u}$ are individual constant symbols, and we let $\mathcal{L}^{++} = \mathcal{L}^+ \cup \{+, \cdot\}$, where $+, \cdot$ are 2-ary function symbols. We define an $\mathcal{L}^{++}$-structure $\mathfrak{A}^{++}$ such that $(\omega, +, \cdot)$ is definable in $\mathfrak{A}^{++}$ and such that $\mathfrak{A}^{++}$ is an expansion by definitions of $\mathfrak{A}^+ = \mathfrak{A}^{++} \restriction \mathcal{L}^+$. It then follows, using Lemma 2.12 and Theorem 2.7, that $\mathfrak{A} = \mathfrak{A}^+ \restriction \mathcal{L}$ is strongly undecidable.

We let $A = \omega \cup (\omega \times \omega) \cup \{u\}$, where $u \notin \omega \cup (\omega \times \omega)$, and we define $R^{\mathfrak{A}} \subseteq A \times A$ as follows:

$$\begin{array}{rcl} R^{\mathfrak{A}} & = & \{(u, k) \colon k \in \omega\} \\ & \cup & \{(k, (k, n)) \colon k, n \in \omega\} \\ & \cup & \{((k, n), n) \colon k, n \in \omega\} \\ & \cup & \{((0, k), (n, k + n)) \colon k, n \in \omega\} \\ & \cup & \{((1, k), (n, k \cdot n)) \colon k, n \in \omega\}. \end{array}$$

In $\mathfrak{A}^+$ the constants $\overline{0}, \overline{1}, \overline{u}$ are interpreted by $0, 1, u$, respectively. In $\mathfrak{A}^{++}$ the interpretation of $+$ is defined by the following $\mathcal{L}^+$-formula:

$$\begin{array}{l} [\neg(R\overline{u}x \wedge R\overline{u}y) \wedge z \equiv \overline{u}] \vee \\ \quad [R\overline{u}x \wedge R\overline{u}y \wedge R\overline{u}z \wedge \exists v_1 v_2 (R\overline{0}v_1 \wedge Rv_1 x \wedge Ryv_2 \wedge Rv_2 z \wedge Rv_1 v_2)]. \end{array}$$

The interpretation of $\cdot$ in $\mathfrak{A}^{++}$ is similarly defined, with $\overline{0}$ replaced by $\overline{1}$. The assertions about $\mathfrak{A}^{++}$ are then easily verified. $\square$

We thus can conclude the following negative solution to the decision problem, originally obtained by Church in 1936.

COROLLARY 2.14. $\mathrm{Cn}_{\mathcal{L}}(\varnothing)$ *is undecidable provided $\mathcal{L}$ contains at least one n-ary predicate or function symbol with $n \geq 2$.*

If $\mathcal{L}$ contains only 1-ary predicate symbols and individual constants then $\mathrm{Cn}_{\mathcal{L}}(\varnothing)$ is decidable, as was pointed out in Section 4. If $\mathcal{L}$ contains at least two 1-ary function symbols then $\mathrm{Cn}_{\mathcal{L}}(\varnothing)$ is undecidable. On the other hand, if $\mathcal{L}$ contains just a single 1-ary function symbol then $\mathrm{Cn}_{\mathcal{L}}(\varnothing)$ is decidable.

Methods similar to the above can be used to show that there are strongly undecidable partial orders and strongly undecidable groups. Hence the elementary theory of partial orders and the elementary theory of groups are both undecidable.

3. Recursively Enumerable Relations

In the further development of recursion theory the r.e. sets and relations will take on increased significance. In this section we begin developing the theory of r.e. sets. In particular we characterize the r.e. relations as those weakly representable in Q, which corresponds to the similar characterization of recursive relations. We further show that the r.e. relations are precisely those definable in $\mathfrak{N}$ by Σ-formulas and that there are "universal" r.e. relations. These last two results have no analogues for recursive relations and are an indication that the r.e. relations have a "smoother" theory than the recursive relations.

Recall that we have shown that a relation R is recursive if and only if R and $\neg R$ are both r.e. Further, we know (from the Incompleteness Theorem) that there are r.e. sets which are not recursive, so the r.e. relations are not closed under complement. Our first lemma gives some closure properties of the r.e. relations.

Lemma 3.1.

(a) Assume that $R, S \subseteq \omega^n$ are both r.e. Then so are $(R \wedge S)$, $(R \vee S)$.
(b) Assume $R(\vec{x}, y, z) \subseteq \omega^{n+2}$ is r.e. Then so are $\exists y R$ and $(\forall y)_{<z} R$.
(c) Assume $R \subseteq \omega^n$ is r.e. and $F_i \colon \omega^k \to \omega$ are recursive for all $i = 1, \ldots, n$. Then $S \subseteq \omega^k$ defined by

$$S(\vec{y}) \Leftrightarrow R(F_1(\vec{y}), \ldots, F_n(\vec{y}))$$

is r.e.

Proof. (a) Assume that $R(\vec{x}) \Leftrightarrow \exists y R^*(\vec{x}, y)$ and $S(\vec{x}) \Leftrightarrow \exists S^*(\vec{x}, y)$, where R^* and S^* are both recursive. Then we have

$$[R(\vec{x}) \wedge S(\vec{x})] \Leftrightarrow \exists w[R^*(\vec{x}, (w)_0) \wedge S^*(\vec{x}, (w)_1)],$$

so $(R \wedge S)$ is r.e.

(b) Assume that $R(\vec{x}, y, z) \Leftrightarrow \exists u R^*(\vec{x}, y, z, u)$, where R^* is recursive. Then we have

$$(\forall y)_{<z} R \Leftrightarrow (\forall y)_{<z} \exists u R^*(\vec{x}, y, z, u) \Leftrightarrow (\exists w)(\forall y)_{<z} R^*(\vec{x}, y, z, (w)_y),$$

so this is also r.e.

The other parts are left to the reader. $\square$

It is interesting to note that the r.e. sets are precisely the "listable" sets of Chapter **??**. The proof of this is left to the reader.

Theorem 3.2. *Assume $X \subseteq \omega$ and $X \neq \varnothing$. Then X is r.e. if and only if $X = \mathrm{ran}(F)$ for some recursive function $F \colon \omega \to \omega$.*

DEFINITION 3.1. Let T be a theory of some language $\mathcal{L} \supseteq \{s, \overline{0}\}$, let $R \subseteq \omega^n$, and let $\phi(x_1, \ldots, x_n)$ be an $\mathcal{L}$-formula. Then R is *weakly representable in T by ϕ* provided the following holds:

$$\begin{aligned}&\text{for every } k_1, \ldots, k_n \in \omega, R(k_1, \ldots, k_n) \text{ holds}\\ &\text{if and only if } T \vdash \phi(\overline{k}_1, \ldots, \overline{k}_n).\end{aligned}$$

Note that if T is consistent and R is representable in T by ϕ then R is weakly representable in T by ϕ. Conversely, if T is complete and R is weakly representable in T by ϕ then R is representable in T by ϕ. Note however that it is *not* true that if R is weakly representable in T_1 by ϕ and $T_1 \subseteq T_2$, T_2 consistent, then R is weakly representable in T_2 by ϕ. Our first theorem states that weak representability corresponds exactly to recursive enumerability.

THEOREM 3.3. *Let T be a theory of $\mathcal{L} = \{+, \cdot, <, s, \overline{0}\}$ such that $\mathfrak{N} \models T, Q \subseteq T$, and T is recursively axiomatizable. Let $R \subseteq \omega^n$. The following are equivalent:*

(1) R is r.e.
(2) R is weakly represented in T by a Σ-formula.
(3) R is weakly represented in T.

PROOF. (2) $\Rightarrow$ (0): Assume R is weakly represented in T by ϕ. Then for any $k_1, \ldots, k_n \in \omega$ we know that $R(k_1, \ldots, k_n)$ holds if and only if $\mathrm{Thm}_T(\ulcorner \phi(\overline{k}_1, \ldots, \overline{k}_n)\urcorner)$. Thus R is r.e. since Thm_T is r.e. and the function $F : \omega^n \to \omega$ defined by $F(k_1, \ldots, k_n) = \ulcorner \phi(\overline{k}_1, \ldots, \overline{k}_n)\urcorner$ is recursive.

(0) $\Rightarrow$ (1): Since R is r.e. we know that $R(k_1, \ldots, k_n)$ holds if and only if there exists $l \in \omega$ such that $R^*(\vec{k}, l)$ holds, where R^* is recursive. Thus we know R^* is representable in T by some Σ-formula $\psi(\vec{x}, y)$. We claim that the Σ-formula $\phi(\vec{x}) = \exists y \psi$ weakly represents R in T. It is clear that if $R(k_1, \ldots, k_n)$ holds then $T \vdash \phi(\overline{k}_1, \ldots, \overline{k}_n)$. Conversely, if $T \vdash \phi(\overline{k}_1, \ldots, \overline{k}_n)$ then $\mathfrak{N} \models \phi(\overline{k}_1, \ldots, \overline{k}_n)$, hence $\mathfrak{N} \models \psi(\overline{k}_1, \ldots, \overline{k}_n, \overline{l})$ for some $l \in \omega$. Since ψ is a Σ-formula we know that $Q \vdash \psi(\overline{k}_1, \ldots, \overline{k}_n, \overline{l})$ and so $R^*(\vec{k}, l)$ holds, and thus $R(\vec{k})$ holds. $\square$

Since $Q \vdash \phi(\overline{k}_1, \ldots, \overline{k}_n)$ if and only if $\mathfrak{N} \models \phi(\overline{k}_1, \ldots, \overline{k}_n)$ for Σ-formulas $\phi(\vec{x})$ we can immediately conclude:

COROLLARY 3.4. *$R \subseteq \omega^n$ is r.e. if and only if R is definable in $\mathfrak{N}$ by a Σ-formula.*

As in Section 1 we use the following notation: if $R \subseteq \omega^{n+1}$ and $k \in \omega$ then $R_k = \{(l_1, \ldots, l_n) : R(k, \vec{l}) \text{ holds}\}$. We can thus think of R as the indexed family $\{R_k\}_{k \in \omega}$ of n-ary relations. The Diagonalization Lemma 1.1 then establishes that there in no 2-ary recursive relation which indexes all recursive sets. On the other hand there are r.e. relations which index all r.e. sets.

THEOREM 3.5. *For each positive integer n there is an r.e. relation $S^n \subseteq \omega^{n+1}$ such that $\{S^n_k : k \in \omega\}$ lists all the n-ary r.e. relations.*

PROOF. Let $F : \omega^{n+1} \to \omega$ be a recursive function such that whenever $k = \ulcorner \phi \urcorner$ where $\phi(v_0, \ldots, v_{n-1})$ is a formula of $\mathcal{L} = \{+, \cdot, <, s, \overline{0}\}$ then $F(k, l_0, \ldots, l_{n-1}) = \ulcorner \phi(\overline{l}_0, \ldots, \overline{l}_{n-1})\urcorner$. Let S^n be defined by the following:

$$S^n(k, l_0, \ldots, l_{n-1}) \text{ holds if and only if } \mathrm{Thm}_Q(F(k, l_0, \ldots, l_{n-1})).$$

Then S^n is r.e. and, by Theorem 3.3, every n-ary r.e. relation is S^n_k for some $k \in \omega$. $\square$

If $n = 1$ we will normally omit the superscript. This result, giving the existence of "universal" r.e. relations, is also known as the Enumeration Theorem.

We may similarly, given a function $F\colon \omega^{n+1} \to \omega$, define functions $F_k\colon \omega^n \to \omega$ by $F_k(l_1, \dots, l_n) = F(k, \vec{l})$ for all $l_1, \dots, l_n \in \omega$ and then think of F as indexing the family $\{F_k\}_{k\in\omega}$ of functions. It is easy to see that there is no 2-ary recursive function F which indexes all 1-ary recursive functions. This is a clear defect—it would be desirable to have such a universal recursive function. The question then arises: is there some natural extension of the notion of computable function for which there are such universal functions? In light of Theorem 3.5, it would seem natural to consider "r.e." functions, i.e., those whose graphs are r.e. Unfortunately this does not get us anywhere.

Lemma 3.6. *Let $F\colon \omega^n \to \omega$ be a function whose graph is r.e. Then F is recursive.*

Proof. By hypothesis, there is some recursive $R \subseteq \omega^{n+2}$ such that $F(k_1, \dots, k_n) = l \Leftrightarrow \exists z R(\vec{k}, l, z)$. Then $F(\vec{k}) = ((\mu w)R(\vec{k}, (w)_0, (w)_1))_0$, so F is recursive. □

4. Hilbert's Tenth Problem

We know that every Σ-formula (in $\mathcal{L} = \{+, \cdot, <, s, \overline{0}\,\}$) defines an r.e. relation on ω in $\mathfrak{N}$. More importantly we have the converse that every r.e. relation on ω is definable in $\mathfrak{N}$ by a Σ-formula. The presence of the bounded universal quantifiers in Σ-formulas makes it difficult to interpret them in a mathematically "natural" way. Surprisingly, it turns out that we don't need bounded universal quantifiers to define r.e. relations. As a consequence we see that all r.e. relations are definable in terms of polynomial equalities, and this fact then enables us to solve Hilbert's Tenth Problem, on diophantine equations.

Definition 4.1. A formula $\phi(x_0, \dots, x_{n-1})$ of $\mathcal{L} = \{+, \cdot, <, s, \overline{0}\}$ is an *equational $\exists$-formula* if and only if it has the form $\exists y_0 \dots \exists y_{m-1}[t_1(\vec{x}, \vec{y}) \equiv t_2(\vec{x}, \vec{y})]$, where t_1 and t_2 are terms of $\mathcal{L}$.

Theorem 4.1. *For any Σ-formula $\phi(x_0, \dots, x_{n-1})$ of $\mathcal{L}$ there is an equational $\exists$-formula $\phi^*(x_0, \dots, x_{n-1})$ of $\mathcal{L}$ such that $\mathfrak{N} \models \forall\vec{x}[\phi \leftrightarrow \phi^*]$.*

4.2. Corollary (Matijasevič). *A relation on ω is r.e. if and only if it is definable in $\mathfrak{N}$ by an equational $\exists$-formula.*

The proof of Theorem 4.1 is by induction on the definition of Σ-formulas. The only hard case is that of the bounded universal quantifier, which uses some tricky (but elementary) number theory. We will discuss the easy steps of the proof later but will not attempt the bounded universal quantifier step.

Assuming this result for the present, we show how this leads to the solution of Hilbert's famous problem (from his 1900 address at the International Congress of Mathematicians) on the solvability of diophantine equations in integers.

Definition 4.2.

(a) A *diophantine equation* is an equation of the form $P(x_1, \dots, x_n) = 0$ where $P(x_1, \dots, x_n)$ is a polynomial in the unknowns $x_1, \dots, x_n$ with coefficients from $\mathbb{Z}$.

(b) A *solution* to the equation $P(x_1, \ldots, x_n) = 0$ is a tuple $(k_1, \ldots, k_n)$ of numbers such that $P(k_1, \ldots, k_n) = 0$. This is a solution *in* $\mathbb{Z}$ (or, simply, *in integers*) provided $k_1, \ldots, k_n \in \mathbb{Z}$.

The connection between diophantine equations and equational $\exists$-formulas is as follows:

LEMMA 4.3. *Let $\phi(y)$ be an equational $\exists$-formula of $\mathcal{L}$. Then there is a diophantine equation $P(\vec{x}, y) = 0$ such that for each $k \in \omega$ $\mathfrak{N} \models \phi(\overline{k})$ if and only if $P(\vec{x}, k) = 0$ has a solution in $\mathbb{Z}$.*

PROOF. Say $\phi(y)$ is $\exists u_0 \ldots \exists u_{m-1}[t_1(y, \vec{u}) \equiv t_2(y, \vec{u})]$ for terms t_1 and t_2 of $\mathcal{L}$. A polynomial with coefficients from $\mathbb{Z}$ is just a term in $\mathcal{L} \cup \{-\}$, so let $Q(\vec{u}, y)$ be the polynomial $(t_1 - t_2)$. Then for any $k \in \omega$ $\mathfrak{N} \models \phi(\overline{k})$ if and only if $Q(\vec{u}, k) = 0$ has a solution in ω. To obtain the desired polynomial $P(\vec{x}, y)$ we once again use Lagrange's theorem that an integer is non-negative if and only if it is the sum of four squares of integers. So we let $\vec{x}$ be $x_0, \ldots, x_{n-1}$, where $n = 4m$, and we let $P(\vec{x}, y)$ be obtained from $Q(\vec{u}, y)$ by replacing each u_i in Q by

$$x_{4i}^2 + x_{4i+1}^2 + x_{4i+2}^2 + x_{4i+3}^2.$$

Then clearly P is as desired. □

Note here that $P(\vec{x}, k)$ is also a polynomial, in the unknowns $\vec{x}$, and thus that $P(\vec{x}, k) = 0$ is a diophantine equation, for each fixed k.

We can now state Hilbert's famous problem from his International Congress address in 1900, concerning diophantine equations.

Hilbert's Tenth Problem. *Is there an effective procedure which, given any diophantine equation $P(\vec{x}) = 0$, will decide whether or not it has a solution in integers?*

Matijasevič showed (1970) that this problem has a negative solution. Assuming his result from Corollary 4.2, we do the same.

THEOREM 4.4. *There is no effective procedure as asked for in Hilbert's Tenth Problem.*

PROOF. Let $X \subseteq \omega$ be r.e. but not recursive. So by Corollary 4.2 together with Lemma 4.3 we know there is a polynomial $P(\vec{x}, y)$ with coefficients from $\mathbb{Z}$ such that for every $k \in \omega$, $k \in X$ if and only if the diophantine equation $P(\vec{x}, k) = 0$ has a solution in $\mathbb{Z}$. Suppose there were an effective procedure to decide, given any diophantine equation $P'(\vec{x}) = 0$, whether or not it has a solution in $\mathbb{Z}$. Then in particular this procedure would decide, given any $k \in \omega$, whether or not $P(\vec{x}, k) = 0$ has a solution in $\mathbb{Z}$, that is, whether or not $k \in X$. But X is not recursive, so there is no such effective procedure. □

As another amazing consequence of Matijasevič's general theorem we give the following:

THEOREM 4.5. *Let $X \subseteq \omega$ be r.e., $X \neq \varnothing$. Then there is a polynomial $Q(\vec{u})$ with coefficients from $\mathbb{Z}$ such that $X = \operatorname{ran}(Q) \cap \omega$.*

PROOF. We know that there is a polynomial $P(\vec{x}, y)$ such that $k \in X$ if and only if $P(\vec{x}, k) = 0$ for some $\vec{x}$ in $\mathbb{Z}$. We define $Q(\vec{x}, y) = y - (y+1) \cdot P(\vec{x}, y)^2$. Then

Q is as desired since $Q(\vec{x}, y) \geq 0$ if and only if $y \geq (y+1) \cdot P(\vec{x}, y)^2$ if and only if $y \geq 0$ and $P(\vec{x}, y) = 0$ if and only if $y \geq 0$ and $y \in X$ and $Q(\vec{x}, y) = y$. $\square$

The reader who doubts that Theorem 4.5 is truly amazing should try to come up with such a polynomial for, say, X equal to the set of primes.

So how is Theorem 4.1, which makes all of this possible, proved? We must show that all atomic and negated atomic formulas of $\mathcal{L}$ are equivalent in $\mathfrak{N}$ to equational $\exists$-formulas, and we must show the equational $\exists$-formulas are closed (up to equivalence in $\mathfrak{N}$) under $\wedge$, $\vee$, $\exists x$ and *bounded universal quantification* $(\forall x)_{<y}$. All steps except the last are easy, and we give a few of them.

The atomic $\mathcal{L}$-formula $(t_1(\vec{x}) < t_2(\vec{x}))$ is equivalent in $\mathfrak{N}$ to the equational $\exists$-formula $\exists u[t_1(\vec{x}) + su \equiv t_2(\vec{x})]$. Its negation $\neg(t_1 < t_2)$ is equivalent in $\mathfrak{N}$ to $(t_1 \equiv t_2 \vee t_2 < t_1)$, which will be fine once we see that the equational $\exists$-formulas are closed under $\vee$. Suppose $\phi(\vec{x})$ and $\psi(\vec{x})$ are equational $\exists$-formulas, say ϕ is $\exists \vec{y}(t_1 \equiv t_2)$ and ψ is $\exists \vec{w}(t_3 \equiv t_4)$. The following are then equivalent in $(\mathbb{Z}, +, \cdot, -, s, 0)$:

$$\begin{aligned}
&(t_1 \equiv t_2 \vee t_3 \equiv t_4),\\
&(t_1 - t_2) \equiv 0 \vee (t_3 - t_4) \equiv 0,\\
&(t_1 - t_2) \cdot (t_3 - t_4) \equiv 0,\\
&t_1 \cdot t_3 + t_2 \cdot t_4 \equiv t_1 \cdot t_4 + t_2 + t_3.
\end{aligned}$$

Thus, assuming $\vec{w}$, $\vec{y}$ are disjoint lists of variables, we see that on $\mathfrak{N}$, $(\phi \vee \psi)$ is equivalent to

$$\exists \vec{y}\, \exists \vec{w}[t_1 \cdot t_3 + t_2 \cdot t_4 \equiv t_1 \cdot t_4 + t_2 \cdot t_3].$$

The cases of $(\phi \wedge \psi)$ and $\neg t_1 \equiv t_2$ are similar and left to the reader.

We have thus reduced everything to the assertion that the equational $\exists$-formulas are closed (up to equivalence in $\mathfrak{N}$) under bounded universal quantification, the proof of which is long and involves elementary, but tricky, number theory.

5. Gödel's Second Incompleteness Theorem

Gödel's Second Incompleteness Theorem was even more philosophically devastating than the first. It says that if T is a consistent, recursively axiomatized extension of Peano arithmetic then there is *no* sentence—not even a logically false sentence—whose unprovability in T is provable from T. We will not dwell on the philosophical aspects, but it should be clear that this is surprising.

To begin with, we revisit the proof of Gödel's First Incompleteness Theorem and wrest from it the following theorem of independent interest:

5.1. Fixed Point Theorem. *Let T be any theory extending Q. Then for any formula $\psi(y)$ there is some sentence σ such that $T \vdash (\sigma \leftrightarrow \psi(\overline{\ulcorner \sigma \urcorner}))$.*

Proof. Let $\chi(x, y)$ be a formula representing the recursive function $\mathrm{Sub}(x, \ulcorner v_0 \urcorner, \mathrm{Num}(x))$ in T. Note that if $k = \ulcorner \phi \urcorner$ and $l \equiv \ulcorner \phi^{v_0}_{\overline{k}} \urcorner$ then

$$(*) \qquad T \vdash \forall y[\chi(\overline{k}, y) \leftrightarrow y \equiv \overline{l}\,].$$

Now let $\phi(v_0)$ be $\exists y[\psi(y) \wedge \chi(v_0, y)]$, let $k = \ulcorner \phi \urcorner$ and let $\sigma = \phi(\overline{k})$; that is, $\sigma = \exists y[\psi(y) \wedge \chi(\overline{k}, y)]$.

But then $T \vdash \chi(\overline{k}, \overline{l}\,)$, where $l = \ulcorner \sigma \urcorner$, and therefore by (*) we have

$$T \vdash \exists y[\psi(y) \wedge \chi(\overline{k}, y)] \leftrightarrow \psi(\overline{l}),$$

that is, $T \vdash (\sigma \leftrightarrow \psi(\overline{\ulcorner\sigma\urcorner}))$, as desired. $\square$

Recall that for any recursively axiomatizable theory T, $\mathrm{Thm}_T = \{\ulcorner\sigma\urcorner\colon T \vdash \sigma, \sigma$ is a sentence$\}$ is an r.e. set, and hence weakly representable in T by a Σ-formula provided $T \supseteq Q$ and $\mathfrak{N} \models T$. to make our subsequent discussion clearer we make the convention that $\underline{\mathrm{Thm}}_T(y)$ is a Σ-formula which weakly represents Thm_T in T. In fact, we need only assume T is consistent, rather than $\mathfrak{N} \models T$, to get such a formula, and we state our results under this weaker hypothesis.

The following lemma, together with the Fixed Point Theorem 5.1, has the (first) Incompleteness Theorem as an immediate consequence:

LEMMA 5.2. *Let T be a consistent, recursively axiomatizable theory extending Q. Let σ be any sentence such that $T \vdash (\sigma \leftrightarrow \neg\underline{\mathrm{Thm}}_T(\overline{\ulcorner\sigma\urcorner}))$. Then $T \nvdash \sigma$. If further $\mathfrak{N} \models T$ then $\mathfrak{N} \models \sigma$.*

PROOF. Suppose $T \vdash \sigma$. Then $\ulcorner\sigma\urcorner \in \mathrm{Thm}_T$ hence $T \vdash \underline{\mathrm{Thm}}_T(\overline{\ulcorner\sigma\urcorner})$. But by the choice of σ we also have $T \vdash \neg\underline{\mathrm{Thm}}_T(\overline{\ulcorner\sigma\urcorner})$, contradicting the consistency of T.

Thus $T \nvdash \sigma$, and so we must also have $T \nvdash \underline{\mathrm{Thm}}_T(\overline{\ulcorner\sigma\urcorner})$. If $\mathfrak{N} \models T$ it follows that $\mathfrak{N} \models \neg\underline{\mathrm{Thm}}_T(\overline{\ulcorner\sigma\urcorner})$, and so $\mathfrak{N} \models \sigma$ by the choice of σ. $\square$

COROLLARY 5.3. *Let T be a consistent, recursively axiomatizable theory extending Q. Then there is a sentence σ such that $T \nvdash \sigma$ but $T \nvdash \neg\underline{\mathrm{Thm}}_T(\overline{\ulcorner\sigma\urcorner})$.*

One's immediate reaction is that a sentence with the property in Corollary 5.3 is rather special, and that "usually" if $\ulcorner\sigma\urcorner \notin \mathrm{Thm}_T$ then this fact should be provable from T, that is, $T \vdash \neg\underline{\mathrm{Thm}}_T(\overline{\ulcorner\sigma\urcorner})$. On the contrary, however, Gödel's Second Incompleteness Theorem states that there is *no* sentence σ such that $T \vdash \neg\underline{\mathrm{Thm}}_T(\overline{\ulcorner\sigma\urcorner})$, for T a consistent and recursively axiomatizable extension of Peano arithmetic. As we will discuss later, this does assume that the formula $\mathrm{Thm}_T(y)$ is appropriately chosen to have certain "natural" properties. We first state the properties needed and prove the result using them.

***D*1.** If $T \vdash \sigma$ then $T \vdash \underline{\mathrm{Thm}}_T(\overline{\ulcorner\sigma\urcorner})$.

***D*2.** $T \vdash [\underline{\mathrm{Thm}}_T(\overline{\ulcorner\sigma\urcorner}) \rightarrow \underline{\mathrm{Thm}}_T(\overline{\ulcorner\underline{\mathrm{Thm}}_T(\overline{\ulcorner\sigma\urcorner})\urcorner})]$.

***D*3.** $T \vdash ([\underline{\mathrm{Thm}}_T(\overline{\ulcorner\sigma\urcorner}) \wedge \underline{\mathrm{Thm}}_T(\overline{\ulcorner\sigma \rightarrow \theta\urcorner})] \rightarrow \underline{\mathrm{Thm}}_T(\overline{\ulcorner\theta\urcorner}))$.

Note that $D1$ certainly holds, just from the assumption that $\underline{\mathrm{Thm}}_T$ weakly represents Thm_T in T. $D3$ says that $\underline{\mathrm{Thm}}_T(y)$ provably satisfies *Modus Ponens*—as one would reasonably expect. $D2$ is more mysterious, but we comment that the implications are all true in $\mathfrak{N}$, since if $\mathfrak{N} \models \underline{\mathrm{Thm}}_T(\overline{\ulcorner\sigma\urcorner})$ then $T \vdash \underline{\mathrm{Thm}}_T(\overline{\ulcorner\sigma\urcorner})$ since $\underline{\mathrm{Thm}}_T(y)$ is a Σ-formula and $T \supseteq Q$; thus $\mathfrak{N} \models \underline{\mathrm{Thm}}_T(\overline{\ulcorner\underline{\mathrm{Thm}}_T(\overline{\ulcorner\sigma\urcorner})\urcorner})$—and in fact this sentence is provable in T.

Using these properties the Second Incompleteness Theorem is easily derived.

THEOREM 5.4. *Let T be a consistent theory extending Q which has a formula $\underline{\mathrm{Thm}}_T(y)$ satisfying $D1$, $D2$, and $D3$. Then there is no sentence θ such that $T \vdash \neg\underline{\mathrm{Thm}}_T(\overline{\ulcorner\theta\urcorner})$.*

PROOF. Applying the Fixed Point Theorem 5.1 to $\neg\underline{\mathrm{Thm}}_T(y)$ we obtain a sentence σ such that $T \vdash (\sigma \leftrightarrow \neg\underline{\mathrm{Thm}}_T(\overline{\ulcorner\sigma\urcorner}))$ and hence $T \nvdash \neg\underline{\mathrm{Thm}}_T(\overline{\ulcorner\sigma\urcorner})$. Our goal is to show that

$$T \vdash (\underline{\mathrm{Thm}}_T(\overline{\ulcorner \sigma \urcorner}) \to \underline{\mathrm{Thm}}_T(\overline{\ulcorner \theta \urcorner}))$$

for every sentence θ, whence the conclusion follows.

Well, by the choice of σ we have

$$T \vdash (\sigma \to \neg\underline{\mathrm{Thm}}_T(\overline{\ulcorner \sigma \urcorner})),$$

and so by $D1$ we can conclude

$$T \vdash \underline{\mathrm{Thm}}_T(\overline{\ulcorner \sigma \to \neg\underline{\mathrm{Thm}}_T(\overline{\ulcorner \sigma \urcorner})\urcorner}).$$

Using $D3$ we see

$$T \vdash \underline{\mathrm{Thm}}_T(\overline{\ulcorner \sigma \urcorner}) \to \underline{\mathrm{Thm}}_T(\overline{\ulcorner \neg\underline{\mathrm{Thm}}_T(\overline{\ulcorner \sigma \urcorner})\urcorner}).$$

But also by $D2$ we have

$$T \vdash \underline{\mathrm{Thm}}_T(\overline{\ulcorner \sigma \urcorner}) \to \underline{\mathrm{Thm}}_T(\overline{\ulcorner \underline{\mathrm{Thm}}_T(\overline{\ulcorner \sigma \urcorner})\urcorner}).$$

Let χ be $\underline{\mathrm{Thm}}_T(\overline{\ulcorner \sigma \urcorner})$. We then have

$$T \vdash \chi \to \underline{\mathrm{Thm}}(\overline{\ulcorner \chi \urcorner})$$

and

$$T \vdash \chi \to \underline{\mathrm{Thm}}_T(\overline{\ulcorner \neg\chi \urcorner}).$$

Now $T \vdash (\chi \to (\neg\chi \to \theta))$ for any θ, and thus by $D1$ and $D3$ again we find that

$$T \vdash \underline{\mathrm{Thm}}_T(\overline{\ulcorner \chi \urcorner}) \to [\underline{\mathrm{Thm}}_T(\overline{\ulcorner \neg\chi \urcorner}) \to \underline{\mathrm{Thm}}_T(\overline{\ulcorner \theta \urcorner})],$$

and thus

$$T \vdash \chi \to \underline{\mathrm{Thm}}_T(\overline{\ulcorner \theta \urcorner}),$$

exactly what we needed. □

Let θ be some "obviously" false sentence, like $\overline{0} \not\equiv \overline{0}$. Then we write $\underline{\mathrm{Con}}_T$ for the sentence $\neg\underline{\mathrm{Thm}}_T(\overline{\ulcorner \theta \urcorner})$, and we think of it as expressing the statement "T is consistent." Of course, under the hypothesis of Theorem 5.4 we then conclude $T \nvdash \underline{\mathrm{Con}}_T$, that is, "$T$ cannot prove its own consistency."

We finally need to argue that if T is any consistent, recursively axiomatizable extension of Peano arithmetic then there is in fact a formula $\underline{\mathrm{Thm}}_T(y)$ satisfying $D1$, $D2$, and $D3$.

We already noted that any formula weakly representing Thm_T in T will satisfy $D1$.

We next assert that any Σ-formula weakly representing Thm_T in T will satisfy $D2$. We do not give a proof of this since the details are quite involved, but we try to indicate the idea. In Chapter **??** we proved that Σ-sentences true on $\mathfrak{N}$ were provable from any T extending Q. To be precise we established for Σ-formulas $\phi(x_0, \dots, x_{n-1})$ that for every $k_0, \dots, k_{n-1} \in \omega$ if $\mathfrak{N} \models \phi(\overline{k}_0, \dots, \overline{k}_{n-1})$ then $T \vdash \phi(\overline{k}_0, \dots, \overline{k}_{n-1})$, or, in other words,

$$\mathfrak{N} \models \phi(\overline{k}_0, \dots, \overline{k}_{n-1}) \to \underline{\mathrm{Thm}}_T(\overline{\ulcorner \phi(\overline{k}_0, \dots, \overline{k}_{n-1})\urcorner}).$$

This argument used induction on the natural numbers. What we need to know is that this argument can be formalized inside T, provided $T \supseteq P$, to yield

$$T \vdash \phi(\overline{k}_0, \dots, \overline{k}_{n-1}) \to \underline{\mathrm{Thm}}_T(\overline{\ulcorner \phi(\overline{k}_0, \dots, \overline{k}_{n-1})\urcorner}).$$

$D2$ will then be a special case. The difficulty is that to follow the inductive argument we need to establish a stronger result involving free variables, that is, something of the form

$$T \vdash \forall x_0 \ldots \forall x_{n-1}[\phi(x_0, \ldots, x_{n-1}) \to \psi(\vec{x})],$$

where ψ "says" that "the sentence obtained from ϕ by replacing each x_i by $\mathrm{Num}(x_i)$ is provable from T."

Condition $D3$ is deceptively innocuous. This is a property one expects to hold because *Modus Ponens* is our one rule of proof. Indeed, if $\underline{\mathrm{Thm}}_T(y)$ is the Σ-formula we "naturally" get by translating the intuitive definitions into first-order formulas then $D3$ *is* satisfied (but note that there is a long chain of definitions involved so the verification is tedious!). Unfortunately it is *not* true that every Σ-formula weakly representing Thm_T has this property, as the following example shows.

Let T be any consistent, recursively axiomatizable theory extending Peano arithmetic, and suppose that $\underline{\mathrm{Thm}}_T(y)$ is a Σ-formula satisfying $D1$, $D2$, and $D3$. Let θ be some "clearly false" sentence like $\overline{0} \not\equiv \overline{0}$, let $m = \ulcorner\theta\urcorner$, and define $\underline{\mathrm{Thm}}^*_T(y)$ to be $(\underline{\mathrm{Thm}}_T(y) \wedge y \not\equiv \overline{m})$. Then $\underline{\mathrm{Thm}}^*_T(y)$ is a Σ-formula representing Thm_T in T (supposing $\underline{\mathrm{Thm}}_T$ did)—the point is that since T is consistent we have $T \nvdash \theta$, and so for every n we will have $T \vdash \underline{\mathrm{Thm}}_T(\overline{n})$ if and only if $T \vdash \underline{\mathrm{Thm}}^*_T(\overline{n})$. It thus follows that $D1$ and $D2$ both hold. But obviously

$$T \vdash \neg\underline{\mathrm{Thm}}^*_T(\overline{\ulcorner\sigma\urcorner}).$$

This contradiction to Theorem 5.4 shows that $D3$ must fail. The reader should try to establish the failure of $D3$ directly.

6. Exercises

(1)

Part 4

Recursion Theory

CHAPTER 10

Partial Recursive Functions

0. Introduction

In giving a recursive definition of some function, you will specify the finite sequence of steps which show how the function is obtained starting with functions in $\mathcal{S}$ and applying rules $R1$, $R2$, and $R3$. But can you decide whether an arbitrary such sequence actually "works"? The difficulty, of course, is with $R3$—to apply $R3$ to a given $G(\vec{x}, y)$ to obtain $F(\vec{x}) = (\mu y)[G(\vec{x}, y) = 0]$ we need to first know that for all $\vec{k}$ from ω there is some $l \in \omega$ such that $G(\vec{k}, l) = 0$. In general, one cannot decide this, even for primitive recursive functions G. This is the real reason there are no universal recursive functions or relations, as contrasted with r.e. relations.

Not knowing whether a list of instructions (an "algorithm") really computes something is a serious drawback to an adequate analysis of algorithms. The correct approach is to broaden our idea of *what* we are computing from functions (defined on all $k_1, \ldots, k_n \in \omega$) to *partial* functions (possibly defined on just a subset of ω^n). We can then apply $R3$ to any intuitively computable G as above to obtain a computable F—except that F may be partial. In this way we will in fact be able to say that all algorithms compute, but we will be unable to decide whether an algorithm computes a total function.

1. Computability of Partial Functions

Definition 1.1. A *partial function of n arguments* is a function $F : D \to \omega$, where $D \subseteq \omega^n$. F is said to be *total* if and only if $D = \omega^n$. For $\vec{k} = (k_1, \ldots, k_n) \in \omega^n$ we say that F *converges* at $\vec{k}$, written $F(\vec{k})\downarrow$, if and only if $\vec{k} \in D$. If F does not converge at $\vec{k}$ then F *diverges* at $\vec{k}$, written $F(\vec{k})\uparrow$.

If F, G are partial functions of n, m arguments, respectively, $\vec{k} \in \omega^n$, $\vec{l} \in \omega^m$, then we write $F(\vec{k}) = G(\vec{l})$ if and only if *either* $F(\vec{k})\downarrow$ and $G(\vec{l})\downarrow$ and they have the same value, *or* both $F(\vec{k})\uparrow$ and $G(\vec{l})\uparrow$. We use $F(\vec{k}) \neq G(\vec{l})$ to mean the negation of $F(\vec{k}) = G(\vec{l})$.

Note that it is now possible to have $F(k) = F(k) + 1$ for some F, k—this will happen if and only if $F(k)\uparrow$.

We can adapt our intuitive definition of "computable" from Chapter **??** to partial functions as follows: a partial function F of n arguments is computable if and only if there is an algorithm (i.e., a finite list of instructions) such that for any $(k_1, \ldots, k_n) \in \omega^n$ the instructions can be applied to $\vec{k}$ so that if $F(\vec{k})\downarrow$ then after a finite number of discrete steps the process terminates and yields the value of $F(\vec{k})$; on the other hand if $F(\vec{k})\uparrow$ then the process yields no result.

We emphasize that the computability of F does *not* mean that we can decide whether or not $F(\vec{k})\downarrow$.

EXAMPLE: Let $R \subseteq \omega \times \omega$ be recursive. Define F on ω by $F(x) = (\mu y)R(x,y)$, that is, $F(k) =$ the least l such that $R(k,l)$ holds provided there is an l such that $R(k,l)$ holds, and $F(k)$ is undefined otherwise.

Then F is computable—given $k \in \omega$ check each value of $l = 0, 1, 2, \ldots$, in turn to see if $R(k,l)$ holds; once you find such an l, stop and output l as the value of F. Note that if there is no such l, then the computation of $F(k)$ never terminates. Note also that the domain of F is some r.e. subset of ω, and that every r.e. subset of ω can be obtained as the domain of some such F.

To see what more precise, formal properties computability should have we make the following observations: it is an essential component of our definition that a computation (according to a fixed algorithm) proceeds in discrete steps (indexed by natural numbers) and that after any particular (finite) number of steps it can be decided what value, if any, the computation to that point has produced. That is, if I is a list of instructions for computing a partial function of n arguments (say, F_I) then we can define the relation R_I of $(n+2)$ arguments by $R(t, \vec{k}, l)$ holds if and only if after $\leq t$ steps the algorithm I applied to input $\vec{k}$ has terminated and produced the value l.

We thus see that R_I is (intuitively) computable, hence recursive by Church's Thesis, and that $F_I(\vec{k})$ converges to l if and only if $\exists t R_I(t, \vec{k}, l)$. That is, the graph of F_I, as a subset of ω^{n+1}, is r.e.

But the converse of this observation is also clear—if $S \subseteq \omega^{n+2}$ is recursive, F is a partial function of n arguments, and $F(\vec{k}) = l \Leftrightarrow \exists w S(w, \vec{k}, l)$, then $F(\vec{k}) = ((\mu m)S((m)_0, \vec{k}, (m)_1))_1$ is clearly computable.

We have thus derived, using just Church's Thesis, the informal result that a partial function is computable if and only if its graph is r.e. To conclude that the formal notion of recursive that we are about to introduce coincides with our intuitive concept it will thus suffice to show that it too leads to precisely the set of parital functions with r.e. graphs.

Our (formal) definition of recursive for partial functions will exactly parallel the definition of recursive for total functions; that is, it will be the closure of $\mathcal{S}$ under the partial analogues of composition $(R1)$ and μ-recursion $(R3)$—as with total functions, primitive recursion is superfluous. We leave to the reader the task of writing down $(R1^P)$, composition of partial functions. We first define the result of the μ-operator with partial functions.

Definition 1.2. Let G be a partial function of $(n+1)$ arguments. Then $F(\vec{x}) = (\mu y)[G(\vec{x}, y) = 0]$ is defined by:

$$\begin{aligned}&F(\vec{k}) = l \text{ provided } G(\vec{k}, l) = 0 \text{ and}\\&(\forall j < l)[G(\vec{k}, j)\downarrow \text{ and } G(\vec{k}, j) \neq 0].\end{aligned}$$

The reader should see that F is computable provided G is—and that this depends on the requirement that $G(\vec{k}, j)\downarrow$ for all $j < l$.

$(R3^P)$ is thus the rule allowing one to derive $(\mu y)[G(\vec{x}, y) = 0]$ from G.

$\mathcal{R}^P =$ the class of recursive partial functions $=$ the closure of the functions in $\mathcal{S}$ under $(R1^P)$ and $(R3^P)$.

It is easily verified that we have:

Theorem 1.1. *For any partial function F, $F \in \mathcal{R}^P$ if and only if the graph of F is r.e.*

We thus conclude that recursive corresponds exactly to computable, even for partial functions.

$\mathcal{R}^P$ has the same sorts of closure properties as $\mathcal{R}$, which we will tacitly assume and use without explicit comment. In the next section we will see some of the properties that distinguish $\mathcal{R}^P$ from $\mathcal{R}$ and which make it a "better-behaved" class of functions to work with.

2. Universal Partial Recursive Functions

We use the existence of universal r.e. relations to prove the existence of universal partial recursive functions.

A function, perhaps partial, of $(n+1)$ arguments can be thought of as an indexed family of functions of n arguments in the following way:

DEFINITION 2.1. Let F be a (partial) function of $(n+1)$ arguments where $n > 0$. Then for each $k \in \omega$ we define the (partial) function F_k of n arguments by

$$F_k(x_1, \dots, x_n) = F(k, x_1, \dots, x_n).$$

THEOREM 2.1. *"There exist universal partial recursive functions." For each $n > 0$ there is some partial recursive function ϕ^n of $(n+1)$ arguments such that for every partial recursive F of n arguments, $F = (\phi^n)_k$ for some $k \in \omega$—i.e., $F(x_1, \dots, x_n) = \phi^n(k, x_1, \dots, x_n)$ for all $x_1, \dots, x_n$.*

PROOF. Recall from Section 3 that there is a universal r.e. relation S^{n+1} listing all r.e. relations of $(n+1)$ arguments. Thus there is some recursive $R \subseteq \omega^{n+3}$ such that

$$S^{n+1}(z, x_1, \dots, x_n, y) \Leftrightarrow \exists u R(u, z, \vec{x}, y).$$

We define ϕ^n by

$$\phi^n(z, x_1, \dots, x_n) = ((\mu w) R((w)_0, z, \vec{x}, (w)_1))_1.$$

Then ϕ^n is partial recursive, and whenever F is partial recursive of n arguments then its graph is r.e., hence given by S^{n+1}_k for some k, and thus F is $(\phi^n)_k$. □

NOTATION: The e^{th} partial recursive function (of n arguments) in the listing given by Theorem 2.1 will be referred to either as ϕ^n_e or $\{e\}^n$. Thus

$$\phi^n(e, x_1, \dots, x_n) = \phi^n_e(\vec{x}) = \{e\}^n(\vec{x}).$$

The number e is an *index* of the function in question. By long-standing tradition the letter e is invariably used for indices. When $n = 1$ we will omit the superscript.

We may use ϕ^n_-, or $\{-\}^n$, freely in defining partial recursive functions—the resulting (partial) functions are then understood as being defined using composition and the appropriate universal partial recursive function. Thus the following are all partial recursive functions of all the indicated variables, whenever f is partial recursive:

$$\begin{array}{l} f(\{x\}(y)), \\ \{f(x)\}(y), \\ \{x\}(f(y)). \end{array}$$

Note that if $f(k) \uparrow$ then $\phi^n(f(k), \vec{x}) \uparrow$ for all $\vec{x}$, and hence $\{f(k)\}^n = \varnothing$.

Similarly the following relations are r.e. relations in all of the exhibited variables, including e:

$$\begin{gathered}\{e\}(x)=y,\\ \{e\}(x)\downarrow,\\ y\in \operatorname{ran}(\{e\}),\\ \{e\}\neq\varnothing.\end{gathered}$$

If f is partial recursive the following relations are also r.e.:

$$\begin{gathered}\{e\}(x)=f(y),\\ \{f(x)\}y=z.\end{gathered}$$

We essentially already know the following characterization of r.e. sets:

Proposition 2.2. *Let $X\subseteq\omega$. The following are equivalent:*

(1) X is r.e.
(2) $X=\operatorname{dom}(f)$ for some partial recursive f.
(3) $X=\operatorname{ran}(f)$ for some partial recursive f.

Proof. We saw (0)⇒(1) in Section 1, and (1)⇒(0) and (2)⇒(0) are immediate from Theorem 1.1. In Section 3 we stated that (0)⇒(2) where we can in fact take f to be total recursive of one argument. □

We thus obtain an r.e. listing of all r.e. sets from our universal partial recursive function.

Definition 2.2. $W_e=\operatorname{dom}(\{e\})$. So $\{W_e\colon e\in\omega\}$ lists all r.e. subsets of ω, and the 2-ary relation $k\in W_e$ is r.e. We also refer to e as an *index* of the r.e. set W_e.

Clearly the relation $(k\in W_e)$ is not recursive, since there are non-recursive r.e. sets. In fact we can do better.

Definition 2.3. $K=\{e\in\omega\colon e\in W_e\}$.

Lemma 2.3. *K is r.e. but not recursive.*

Proof. K is clearly r.e. If K were recursive then the following function f would be partial recursive, and total, hence recursive:

$$f(k)=\begin{cases}\{k\}(k)+1 & \text{if } k\in K,\\ 0 & \text{if } k\notin K.\end{cases}$$

But then $f=\{e\}$ for some e, yielding $f(e)=f(e)+1$, which is a contradiction since f is total. □

In a similar way one can show, for example, that $\{e\colon W_e\neq\varnothing\}$ is r.e. but not recursive.

Let f be any partial recursive function of two arguments. Then for each $k\in\omega$, f_k is a partial recursive function of one argument, so $f_k=\{e\}$ for some e, depending on k. Perhaps surprisingly, there is a *recursive* function s which takes k to some index e of f_k.

2.4. Parameter Theorem. *Let f be a partial recursive function of $(n+1)$ arguments, where $n>0$. Then there is a total recursive $s\colon\omega\to\omega$ such that for every k, $f_k=\{s(k)\}^n$—i.e.,*

$$f_k(\vec{x})=\{s(k)\}^n(\vec{x})=\phi^n(s(k),\vec{x}).$$

PROOF. Since f is partial recursive, its graph $f(z, \vec{x}) = y$ is weakly represented in Q by some formula $\psi(z, \vec{x}, y)$. It then follows that the formula $\psi(\overline{k}, \vec{x}, y)$ weakly represents the graph of f_k in Q, for each $k \in \omega$. The (total) function $s\colon \omega \to \omega$ such that $s(k) = \ulcorner \psi(\overline{k}, \vec{x}, y) \urcorner$ is then recursive. By the definition in Theorem 3.5 of our universal r.e. relation S^{n+1} we see that $S^{n+1}_{s(k)}$ is the graph of f_k, and hence that $f_k(\vec{x}) = \phi^n(s(k), \vec{x})$ for all $\vec{x}$, as desired. $\square$

This result says that *any* partial recursive list of partial recursive functions can be recursively mapped to a subfamily of our standard universal list $\{\phi_e : e \in \omega\}$ of all partial recursive functions. Note that our argument shows that the function s can also be taken to be one-to-one.

As a simple application we show the following:

LEMMA 2.5. *Let $X \subseteq \omega$ be r.e. Then there is a total recursive $s\colon \omega \to \omega$ such that for all $k \in \omega$, $k \in X$ if and only if $W_{s(k)} \neq \varnothing$.*

PROOF. X is the domain of some partial recursive function g of one argument. We define the partial recursive function f of two arguments by $f(k, x) = g(k)$, all k, x. The Parameter Theorem 2.4 yields a total recursive $s\colon \omega \to \omega$ such that $f_k = \{s(k)\}$ for all $k \in \omega$. But $k \in X$ implies $g(k)\downarrow$, hence $f_k(x)\downarrow$ for all $x \in \omega$, thus $\{s(k)\}(x)\downarrow$ for all x, so in particular $W_{s(k)} \neq \varnothing$. On the other hand, if $k \notin X$ then $\{s(k)\}(x)\uparrow$ for all $x \in \omega$, so in particular $W_{s(k)} = \varnothing$. $\square$

As we will consider in more detail later, Lemma 2.5 implies that any r.e. set can be recursively "reduced" to $\{e : W_e \neq \varnothing\}$ via the function s. We can conclude that $\{e : W_e \neq \varnothing\}$ is *not* recursive just by considering an r.e. X which is not recursive.

3. The Recursion Theorem

The following results give important "fixed-point" properties of our universal listing of partial recursive functions.

3.1. Recursion Theorem.

(a) Let f be any total recursive function of one argument. Then there is some $e \in \omega$ such that $\{e\} = \{f(e)\}$, that is, $\{e\}(k) = \{f(e)\}(k)$ for all $k \in \omega$.

(b) Let g be any partial recursive function of two arguments. Then there is some $e \in \omega$ such that $\{e\} = g_e$, that is $\{e\}(k) = g(e, k)$ for all $k \in \omega$.

PROOF. (a) Given f, we first define the partial recursive function h by

$$h(x, y) = \{f(\{x\}(x))\}(y).$$

Applying the Parameter Theorem 2.4 to h we obtain a total recursive function s such that $h(x, y) = \{s(x)\}(y)$, that is, $\{s(x)\}(y) = \{f(\{x\}(x))\}(y)$.

Now s is $\{m\}$ for some m, and so setting $x = m$ we see

$$\{\{m\}(m)\}(y) = \{f(\{m\}(m))\}(y).$$

Finally, setting $e = \{m\}(m)$, which is allowed since $\{m\} = s$ is total, we have $\{e\}(y) = \{f(e)\}(y)$ for all y, as desired.

(b) Given g, we first apply the Parameter Theorem 2.4 to obtain a total recursive function f of one argument such that $\{f(l)\}(k) = g(l, k)$ for all $l, k \in \omega$. Applying (a) to f we obtain $e \in \omega$ such that $\{e\} = \{f(e)\} = g_e$, as desired. $\square$

The "fixed points" e in the preceding result are not unique—in fact there necessarily are infinitely many $e \in \omega$ with the property in each part of the Recursion Theorem 3.1, as we leave the reader to verify. The same arguments show the analogues of these results in which e, $f(e)$ are considered as indices of partial functions of n arguments, and in which g has $(n+2)$ arguments.

As some examples of use of the second form of the Recursion Theorem we have:

EXAMPLE 1: There are (infinitely many) $n \in \omega$ such that $\{n\}(x) = x^n$ for all $x \in \omega$.

PROOF. Consider $g(n, x) = x^n$. □

EXAMPLE 2: There are (infinitely many) $n \in \omega$ such that W_n is the set whose only element is n.

PROOF. Consider

$$g(n, x) = \begin{cases} 0 & \text{if } x = n, \\ \uparrow & \text{otherwise.} \end{cases}$$

□

It is easy to see that every partial recursive function has infinitely many different indices. In fact the set of all indices for any given partial recursive function is not even recursive. More generally we have the following fact:

3.2. Rice's Theorem. *Let $\mathcal{F}$ be any set of partial recursive functions of one argument, and let $I = I_{\mathcal{F}} = \{e \colon \{e\} \in \mathcal{F}\}$. Assume that I is recursive. Then either $\mathcal{F} = \varnothing$ or $\mathcal{F} =$ the set of all partial recursive functions. Thus either $I = \varnothing$ or $I = \omega$.*

PROOF. Suppose not, and choose e_0, e_1 such that $\{e_0\} \in \mathcal{F}$, $\{e_1\} \notin \mathcal{F}$. Define f by

$$f(k) = \begin{cases} e_1 & \text{if } k \in I, \\ e_0 & \text{if } k \notin I. \end{cases}$$

Then by assumption f is a total recursive function, so by the Recursion Theorem 3.1 there is some e such that $\{e\} = \{f(e)\}$.

Now $\{e\} \in \mathcal{F}$ if and only if $e \in I$ if and only if $f(e) \notin I$ if and only if $\{f(e) \notin \mathcal{F}\}$. This contradiction proves the theorem. □

The reader may well wonder how sensitive the above results are to the particular way in which we defined our universal partial recursive functions. That is, is there a partial recursive function ϕ^* of two arguments which is universal, so every partial recursive f of one argument is ϕ^*_e for some $e \in \omega$, but such that our other results fail when indices are taken with respect to ϕ^*? The answer is yes, such "bad" enumerations do exist—but they are bad because the Parameter Theorem 2.4 (actually its generalization, the "*s-m-n* Theorem") fails for them. That is, if we are dealing with any universal partial recursive function which is also universal for such enumerations, in the sense that the Parameter Theorem holds, then all of our results will also hold for this enumeration. This is why we needed to refer to Gödel numbers of weakly representing formulas in our proof of the Parameter Theorem but not in subsequent results.

4. Complete Recursively Enumerable Sets

In Section 2 we showed that for any r.e. set X there is a total recursive function s such that $k \in X$ if and only if $s(k) \in \{e\colon W_e \neq \varnothing\}$. We remarked that we could conclude from this that $\{e\colon W_e \neq \varnothing\}$ is not recursive. The following definitions and lemma make explicit what we were referring to:

Definition 4.1.

(a) Let $A, B \subseteq \omega$. Then A is *many-one reducible to* B, $A \leq_m B$, if and only if there is a total recursive function s such that

$$k \in A \Leftrightarrow s(k) \in B.$$

(b) A and B are *many-one equivalent*, $A \equiv_m B$, if and only if $A \leq_m B$ and $B \leq_m A$.

(c) If B is r.e., then B is *complete r.e.*, or *m-complete r.e.*, if and only if $X \leq_m B$ for every r.e. set X.

Lemma 4.1.

(a) If B is recursive and $A \leq_m B$ then A is recursive.

(b) If B is r.e. and $A \leq_m B$ then A is r.e.

(c) If B is a complete r.e. set then B is not recursive.

Proof. (a) and (b) are clear from the definitions, and (c) follows from (a) and the existence of r.e. sets that are not recursive. □

As we have seen, $\{e\colon W_e \neq \varnothing\}$ is a complete r.e. set; similarly K is complete r.e., $\{\ulcorner\sigma\urcorner\colon Q \vdash \sigma\}$ is complete r.e., etc.

In fact, the natural conjecture at this point would be that all non-recursive r.e. sets are m-complete. We will decide this question after deciding two other, apparently unrelated, questions.

Recall that a set $A \subseteq \omega$ is recursive provided A and $B = (\omega \setminus A)$ are both r.e. A reasonable generalization of this would be to assert that whenever A and B are both r.e. and $(A \cap B) = \varnothing$, then there is some recursive C such that $A \subseteq C, (B \cap C) = \varnothing$. Such A and B are *recursively separable*. Unfortunately, there are recursively inseparable r.e. sets.

Proposition 4.2. *There are r.e. sets A and B such that $(A \cap B) = \varnothing$ but there is no recursive C such that $A \subseteq C$, $(B \cap C) = \varnothing$.*

Proof. Let $A = \{e\colon \{e\}(e) = 1\}$, and let $B = \{e\colon \{e\}(e) = 0\}$. Suppose there is a recursive C with $A \subseteq C$, $B \cap C = \varnothing$. Then K_C is recursive, and so there is some e_0 such that $C = \{k\colon \{e_0\}(k) = 0\}$, where $\{e_0\}$ is total and $\{e_0\}(k) = 0, 1$ for all k. We then obtain a contradiction by considering $\{e_0\}(e_0)$. □

A weakening of the preceding question would ask whether an r.e. set A whose complement is infinite is contained in some recursive set C whose complement is infinite. This also turns out to be false.

Proposition 4.3. *There is a co-infinite r.e. set A such that $(A \cap B) \neq \varnothing$ for every infinite r.e. set.*

Proof. First define a partial recursive function f such that $f(k)\downarrow$ if and only if $\exists l[l \in W_k \wedge 2k < l]$, and whenever $f(k)\downarrow$ then $f(k)$ is some such l. Now let A be the range of f.

Certainly A is r.e., and whenever $B = W_e$ is infinite then $f(e)\downarrow$ and $f(e) \in (A \cap W_e)$, so $(A \cap B) \neq \varnothing$.

To see that $(\omega \setminus A)$ is infinite note that whenever $f(k)\downarrow$ then $f(k) > 2k$; thus if $l \in A$ and $l \leq 2k$ then $(\exists i)_{<k}[l = f(i)]$—that is,

$$|\{l \in A \colon l \leq 2k\}| \leq k,$$

so $|\{l \leq 2k \colon l \notin A\}| > k$, which implies $(\omega \setminus A)$ is infinite. □

Finally, the following result shows that no A as in the preceding result can be m-complete.

PROPOSITION 4.4. *Let A be an m-complete r.e. set. Then $(\omega \setminus A)$ contains an infinite r.e. set.*

PROOF. Let B_0, B_1 be recursively inseparable r.e. sets. Since $B_0 \leq_m A$ there is some recursive f such that $k \in B_0$ implies $f(k) \in A$ and $k \in B_1$ implies $f(k) \notin A$. Let $A^* = \{f(k) \colon k \in B_1\}$. Then A^* is r.e., and $(A \cap A^*) = \varnothing$. We show A^* is infinite.

Let C be $\{k \colon f(k) \in A^*\}$. Then $C \leq_m A^*$ so C is recursive provided A^* is recursive (in particular, finite). But $B_1 \subseteq C$ and $(B_0 \cap C) = \varnothing$, so this would contradict the recursive inseparability of B_0 and B_1. □

5. Exercises

(1)

CHAPTER 11

Relative Recursion and Turing Reducibility

0. Introduction

Since there are just countably many recursive functions, "almost all" of the functions on ω are non-computable. However, we can still compare their relative complexity, meaning their relative difficulty of computation.

As an example, let $g\colon \omega \to \omega$ be given and define $f\colon \omega \to \omega$ by $f(k) = g(2k+1)$ for all $k \in \omega$. Then we can compute f *relative to* g—that is, there is an algorithm for computing the values of f which asks for a value of g at finitely many places in each computation. That is, we can compute f given a reliable source (an "oracle") which will provide values of g upon request. We write $f \leq_T g$ in these circumstances. Note that for appropriately chosen g, we can have f recursive but g not recursive and thus $g \leq_T f$ need not hold.

As another example, note that for any set $A \subseteq \omega$ we will have $K_A \leq_T K_{(\omega\setminus A)}$ and $K_{(\omega\setminus A)} \leq_T K_A$, and so these two characteristic functions have equal degrees of complexity.

In Section 1 we give the formal definition of relative recursivity and some of its elementary properties. In Sections 2 and 3 we develop the theory of recursion relative to some fixed g following the outline of Chapter **??**. In Section 4 we look at the structure on $\mathcal{P}(\omega)$ induced by $\leq_T$ applied to the characteristic functions of the sets.

1. Relative Recursivity

Let $\mathcal{F}$ be a set of total functions on ω. Then the (partial) functions which are computable relative to $\mathcal{F}$ are precisely those which we obtain under the usual procedures for generating partial recursive functions, allowing functions in $\mathcal{F}$ to be used at will, without further justification. Thus, the formal definition is as follows:

Definition 1.1. $\mathcal{R}^P(\mathcal{F})$ is the closure of $(\mathcal{S} \cup \mathcal{F})$ under $R1^P$ and $R3^P$. If $g \in \mathcal{R}^P(\mathcal{F})$ then we say that g is *recursive in* $\mathcal{F}$ or *recursive relative to* $\mathcal{F}$. As usual, a relation is recursive relative to $\mathcal{F}$ provided its characteristic function is. $\mathcal{R}(\mathcal{F})$ is the set of total functions in $\mathcal{R}^P(\mathcal{F})$.

$\mathcal{R}^P(\mathcal{F})$ has the same closure properties as $\mathcal{R}^P$, and we will use these without specific comment. Some further easy properties of $\mathcal{R}^P(\mathcal{F})$ follow:

(1) If $\mathcal{F}_0 \subseteq \mathcal{R}(\mathcal{F}_1)$ then $\mathcal{R}^P(\mathcal{F}_0) \subseteq \mathcal{R}^P(\mathcal{F}_1)$.
(2) If $g \in \mathcal{R}^P(\mathcal{F})$ then there is some finite $\mathcal{F}_0 \subseteq \mathcal{F}$ such that $g \in \mathcal{R}^P(\mathcal{F}_0)$.
(3) For any $\mathcal{F}$ there is $\mathcal{S} \subseteq \mathcal{P}(\omega)$, $|\mathcal{S}| = |\mathcal{F}|$, such that $\mathcal{R}^P(\mathcal{F}) = \mathcal{R}^P(\mathcal{S}) = \mathcal{R}^P(\{K_A\colon A \in \mathcal{S}\})$.

Proof. For $f \in \mathcal{F}$ of n arguments let $A_f = \{\langle k_1, \ldots, k_n, f(k_1, \ldots, k_n)\rangle\colon k_1, \ldots, k_n \in \omega\}$. Then $\mathcal{S} = \{A_f\colon f \in \mathcal{F}\}$ is as desired. $\square$

(3) If $\mathcal{F}$ is finite then there is some $h\colon \omega \to \omega$ such that $\mathcal{R}^P(\mathcal{F}) = \mathcal{R}^P(h)$.

Proof. It suffices to consider $\mathcal{F} = \{f_1, f_2\}$ where each $f_i\colon \omega \to \omega$. In this case we can take $h = f_1 \oplus f_2$, where

$$h(k) = \begin{cases} f_1(n) & \text{if } k = 2n, \\ f_2(n) & \text{if } k = 2n+1. \end{cases}$$

□

If follows that for most purposes it suffices to look at $\mathcal{R}^P(h)$ where $h\colon \omega \to \omega$, and in fact we could restrict to $h\colon \omega \to 2$, that is, $h = K_A$ for $A \subseteq \omega$.

We can now define the ordering $\leq_T$ used in the introduction to this chapter.

Definition 1.2. Let f, g be total functions on ω. Then f is *Turing reducible to* g, $f \leq_T g$, if and only if $f \in \mathcal{R}(g)$. f and g are *Turing equivalent*, $f \equiv_T g$, if and only if $f \leq_T g$ and $g \leq_T f$.

If R, S are relations we will write $R \leq_T S$, etc., to mean $K_R \leq_T S$, etc.

We have two different goals in the rest of this chapter. One is to study $\mathcal{R}^P(f)$ for fixed (total) f; the other is to study the properties of $\leq_T$ on subsets of ω.

The study of the relations r.e. in some fixed f, and the definition of the "jump" of a set, are central to both topics.

2. Representation and Enumeration Theorems

Essential to our development of the theory of partial recursive functions was the observation that a partial function is recursive if and only if its graph is an r.e. relation. A similar result, which in addition shows exactly how f is used, is essential to our work on $\mathcal{R}^P(f)$.

2.1. Representation Theorem. *Fix $f\colon \omega \to \omega$. Let g be a partial function of n arguments. The following are equivalent:*

(1) $g \in \mathcal{R}^P(f)$.

(2) There is an r.e. relation $R \subseteq \omega^{n+2}$ such that $g(\vec{x}) = y$ if and only if $\exists s R(\vec{x}, y, \overline{f}(s))$.

(3) There is a recursive relation $R \subseteq \omega^{n+2}$ such that $g(\vec{x}) = y$ if and only if $\exists s R(\vec{x}, y, \overline{f}(s))$, where R satisfies the following conditions:

(i) $R(\vec{x}, y, z) \Rightarrow \mathrm{Seq}(z)$.

(ii) $R(\vec{x}, y, z) \wedge$ *"z is an initial segment of z'"* $\Rightarrow R(\vec{x}, y, z')$.

Proof. (2) ⇒ (1) is obvious, and (1) ⇒ (0) is like the corresponding fact in Chapter **??**.

The proof of (0) ⇒ (2) is by induction on the length of the "derivation" of g. That is, it suffices to show (2) holds for every $g \in (\mathcal{S} \cup \{f\})$ and that both of the rules $R1^P$ and $R3^P$ preserve this property.

If $g \in \mathcal{S}$ then we can take R as $[g(\vec{x}) = y \wedge \mathrm{Seq}(z)]$. If g is f then we can define R to be

$$[\mathrm{Seq}(z) \wedge \mathrm{lh}(z) > x \wedge (z)_x = y].$$

We leave $R1^P$ to the reader and check use of $R3^P$.

Suppose that $g(\vec{x}, y) = u$ if and only if $\exists s R_g(\vec{x}, y, u, \overline{f}(s))$ where R_g is recursive and satisfies (i) and (ii) above. Let $h(\vec{x}) = (\mu y)[g(\vec{x}, y) = 0]$. Then we see that $h(\vec{x}) = y$ if and only if

$$\exists s R_g(\vec{x}, y, 0, \overline{f}(s)) \wedge (\forall t)_{<y} \exists u \exists s[R_g(\vec{x}, t, u, \overline{f}(s)) \wedge u \neq 0].$$

Using property (ii) of R_g we see this is equivalent to

$$\exists s(R_g(\vec{x}, y, 0, \overline{f}(s)) \wedge (\forall t)_{<y}[R_g(\vec{x}, t, (s)_t, \overline{f}(s)) \wedge (s)_t \neq 0]).$$

The point is, if $(\forall t)_{<y} \exists u_t \exists s_t[R_g(\vec{x}, t, u_t, \overline{f}(s_t) \wedge u \neq 0]$, and if $\exists s_y R_g(\vec{x}, y, 0, \overline{f}(s_y))$, then we can take as s in the final formula any s such that $(s)_t = u_t$ for all $t < y$ and $s \geq s_t$ all $t \leq y$. □

Using the existence of universal r.e. relations we obtain a universal way of enumerating all partial functions (in a fixed number of arguments) recursive in f. It is especially important to notice that the recursive relation R we obtain is *independent of* f.

2.2. Enumeration Theorem. *For each positive $n \in \omega$ there is a recursive $R \subseteq \omega^{n+3}$ such that for every total $f : \omega \to \omega$ and every partial g of n arguments the following are equivalent:*

(1) $g \in \mathcal{R}^P(f)$.

(2) There is some $e \in \omega$ such that $g(\vec{x}) = y$ if and only if $\exists s R(e, \vec{x}, y, \overline{f}(s))$.

Furthermore, R satisfies:

(i) $R(e, \vec{x}, y, z) \Rightarrow \mathrm{Seq}(z)$.

(ii) $R(e, \vec{x}, y, z) \wedge$ "z is an initial segment of z'" $\Rightarrow R(e, \vec{x}, y, z')$.

(iii) $\forall e \forall \vec{x} \forall z \exists^{\leq 1} y R(e, \vec{x}, y, z)$.

Proof. The Representation Theorem 2.1 and the existence of universal r.e. relations yields a recursive $S \subseteq \omega^{n+4}$ such that if g has n arguments then $g \in \mathcal{R}^P(f)$ if and only if there is some $e \in \omega$ such that

$$g(\vec{x}) = y \Leftrightarrow \exists u \exists s S(e, \vec{x}, y, \overline{f}(s), u).$$

We first define $R^*(e, \vec{x}, y, z)$ to be

$$\mathrm{Seq}(z) \wedge (\exists z')_{<z} (\exists i)_{<z} (\exists u)_{<z} [\mathrm{In}(z, i) = z' \wedge S(e, \vec{x}, y, z', u)]$$

and finally define the desired R by

$$R(e, \vec{x}, y, z) \Leftrightarrow R^*(e, \vec{x}, y, z) \wedge (\forall y')_{<y} \neg R(u, x, y', z).$$

□

Note that R as in the Enumeration Theorem 2.2 will be such that for every $f : \omega \to \omega$ we have: $\forall e \forall \vec{x} \exists^{\leq 1} y \exists s R(e, \vec{x}, y, \overline{f}(s))$. We thus obtain universal functions Φ^f which are partial recursive in f.

Theorem 2.3. *For any $f : \omega \to \omega$ and any positive $n \in \omega$ there is a partial function Φ^f of $(n+1)$ arguments such that $\Phi^f \in \mathcal{R}^P(f)$ and for every partial function g of n arguments $g \in \mathcal{R}^P(f)$ if and only if there is some $e \in \omega$ such that $g(\vec{x}) = \Phi^f(e, \vec{x})$ for all $\vec{x} \in \omega^n$. Further, $\Phi^f(e, \vec{x}) = y$ if and only if $\exists s R(e, \vec{x}, y, \overline{f}(s))$ holds, where R is the recursive relation in the Enumeration Theorem 2.2.*

NOTATION: $\Phi^f(e,\vec{x}) = \Phi^f_e(\vec{x}) = \{e\}^f(\vec{x})$. We will write $\{e\}^A$, etc., in place of $\{e\}^{K_A}$ for relations A.

Of course our notation is ambiguous since we suppress reference to $n = \mathrm{lh}(\vec{x})$. This is harmless, and we will continue to do so. Normally we will have $n = 1$ unless explicitly stated to the contrary.

DEFINITION 2.1. Given $f\colon \omega \to \omega$ and $e, n, s \in \omega$, $\{e\}^f_s$ is the partial function of n arguments defined by

$$\{e\}^f_s(\vec{x}) = y \text{ if and only if } R(e,\vec{x},y,\overline{f}(s)) \wedge e, x_i, y < s.$$

Then certainly $\{e\}^f_s$ is partial recursive in f, but more can be said:

LEMMA 2.4.

1. *The relation $\{e\}^f_s(\vec{x}) = y$ is recursive in f (as an $(n+3)$-ary relation in $e, s, \vec{x}, y$).*
2. *$\{e\}^f(\vec{x}) = y$ if and only if $\exists s\{e\}^f_s(\vec{x}) = y$.*
3. *If $\{e\}^f_s(\vec{x}) = y$ and $s < t$ then $\{e\}^f_t(\vec{x}) = y$.*
4. *Assume that $f, g\colon \omega \to \omega$ are such that $f \restriction s = g \restriction s$. Assume that $\{e\}^f_s(\vec{x}) = y$. Then also $\{e\}^g_s(\vec{x}) = y$, so in particular $\{e\}^f(\vec{x}) = \{e\}^g(\vec{x})$.*

Finally, just as in Section 2, we obtain the Relativized Parameter Theorem.

THEOREM 2.5. *Let $g(z,\vec{x}) \in \mathcal{R}^P(f)$. Then there is a total recursive $s\colon \omega \to \omega$ such that for every $k \in \omega$, $g_k(\vec{x}) = g(k,\vec{x}) = \{s(k)\}^f(\vec{x})$.*

3. A-Recursively Enumerable Relations

The relativized notion of recursive enumerability is defined in the obvious way.

DEFINITION 3.1. A relation $R \subseteq \omega^n$ is *r.e. in A*, or *A-r.e.*, if and only if there is some $S \subseteq \omega^{n+1}$ which is recursive in A such that $\forall k_1, \dots, k_n \in \omega$,

$$R(k_1,\dots,k_n) \text{ if and only if } \exists l S(k_1,\dots,k_n,l).$$

The standard properties of r.e. relations go over into this setting. We mention the following, without proof:

LEMMA 3.1.

1. *R is recursive in A if and only if R and $\neg R$ are both r.e. in A.*
2. *A partial function g is recursive in A if and only if the graph of g is r.e. in A.*
3. *A set $X \subseteq \omega$ is A-r.e. if and only if $X = \mathrm{dom}(g)$ for some $g \in \mathcal{R}^P(A)$.*

DEFINITION 3.2. $W^A_e = \mathrm{dom}(\{e\}^A)$.

Thus $\{W^A_e\colon e \in \omega\}$ is an A-r.e. listing of all subsets of ω which are A-r.e.

Recall that $K = \{e\colon e \in W_e\}$ is an r.e. set which is not recursive and that every r.e. set is m-reducible to K. The definition of K generalizes to the relativized context to give a function yielding a canonical A-r.e. set which has the analogous maximality property.

DEFINITION 3.3. For any $A \subseteq \omega$, the *jump* of A is $A' = \{e\colon e \in W^A_e\}$.

THEOREM 3.2. *For any $A, B, C \subseteq \omega$ we have:*

(1) A' *is* A*-r.e.*
(2) $A' \not\leq_T A$.
(3) B *is* A*-r.e. if and only if* $B \leq_m A'$.
(4) *If* B *is* A*-r.e. and* $A \leq_T C$ *then* B *is* C*-r.e.*
(5) $B \leq_T A$ *if and only if* $B' \leq_m A'$.
(6) $B \equiv_T A$ *if and only if* $B' \equiv_m A'$.

PROOF. (0), (1), and (2) are established just as the corresponding facts for K.
(3) is easily proved using Definition 3.1.
(5) follows from (4).
We prove (4). First assume $B \leq_T A$. By (0), B' is B-r.e. so by (3) we see B' is A-r.e. and so $B' \leq_m A'$ by (2).

Next suppose $B' \leq_m A'$. Then in particular $B \leq_m A'$ (since $B \leq_m B'$ by (2)) so B is r.e. in A by (2). But we also have $(\omega \setminus B) \leq_m B'$ since $(\omega \setminus B) \equiv_T B$, and thus we also conclude that $(\omega \setminus B)$ is r.e. in A. Hence B is recursive in A by Lemma 3.1, i.e., $B \leq_T A$. $\square$

COROLLARY 3.3. *If* B *is* A*-r.e. then* $B \leq_T A'$*; if* $B \leq_T A$ *then* $B' \leq_T A'$.

WARNING: The converses of these implications do not hold.

COROLLARY 3.4. $K \equiv_T \varnothing'$.

NOTATION: $\varnothing^{(n+1)} = (\varnothing^{(n)})'$.

We then have:

$$\varnothing <_T \varnothing' <_T \varnothing'' <_T \cdots <_T \varnothing^{(n+1)} <_T \cdots .$$

As remarked above, if $B \leq_T A'$ it need not follow that B is A-r.e. We are, however, able to characterize such Bs as the limits of A-recursive sequences. More generally we characterize the (total) functions f such that $f \leq_T A'$.

DEFINITION 3.4. Let $g\colon \omega \times \omega \to \omega$ and $f\colon \omega \to \omega$. We consider g as the sequence $\{g_n\}_{n\in\omega}$ of 1-ary functions as usual, and we say that f is the *limit* of $\{g_n\}_{n\in\omega}$, written $f = \lim_{n\to\infty} g_n$, if and only if for every $k \in \omega$ we have $f(k) = \lim_{n\to\infty} g_n(k)$, meaning that there is some n_0 such that $g_n(k) = f(k)$ for all $n \geq n_0$. If $\lim g_n = f$ then a *modulus* for the sequence is a function $m\colon \omega \to \omega$ such that $g_n(k) = f(k)$ for all $n \geq m(k)$, for all $k \in \omega$.

THEOREM 3.5. *For any* $A \subseteq \omega$ *and* $f\colon \omega \to \omega$*,* $f \leq_T A'$ *if and only if there is an* A*-recursive sequence* $\{g_n\}_{n\in\omega}$ *(i.e.,* $g \leq_T A$*) such that* $f = \lim_{n\to\infty} g_n$.

We first prove a lemma on limits of A-recursive sequences.

LEMMA 3.6. *Assume that* $f, g\colon \omega \to \omega$*,* $f \leq_T g$ *and* g *is the limit of an* A*-recursive sequence. Then* f *is also the limit of an* A*-recursive sequence.*

PROOF. Let $g = \lim_{n\to\infty} G_n$, with $\{G_n\}_{n\in\omega}$ an A-recursive sequence. Since $f \leq_T g$ we know $f = \{e\}^g$ for some $e \in \omega$. That is, $f(k) = l$ if and only if $R(e, k, l, \overline{g}(t))$ holds for some (and hence for all sufficiently large) t, where R is as in the Enumeration Theorem 2.2. Since $g = \lim_{n\to\infty} G_n$ there is $m\colon \omega \to \omega$ such that $\overline{g}(t) = \overline{G}_n(t)$ for all $n \geq m(t)$. Thus, if $R(e, k, l, \overline{g}(t))$ holds then $R(e, k, l, \overline{G}_n(t))$ holds for all $n \geq m(t)$ and so also $R(e, k, l, \overline{G}_n(n))$ holds for all $n \geq m(t), t$. We may thus define $F(n, k) = (\mu y)_{<n} R(e, k, y, \overline{G}_n(n))$. Then $F \leq_T G \leq_T A$ and $f(k) = \lim_{n\to\infty} F_n(k)$ for all k, as desired. $\square$

Proof of Theorem 3.5. First, suppose $f \leq_T A'$. By Lemma 3.6 it suffices to show that $K_{A'}$ is the limit of an A-recursive sequence. A', being A-r.e., is the domain of $\{e\}^A$ for some $e \in \omega$. We can then define

$$G(s,x) = \begin{cases} 0 & \text{if } (\exists y)_{<s}\{e\}_s^A(x) = y, \\ 1 & \text{otherwise.} \end{cases}$$

and see that $G \leq_T A$ and $K_{A'} = \lim_{s\to\infty} G_s$ as desired.

For the other direction, suppose f is the limit of the A-recursive sequence $\{g_n\}_{n\in\omega}$. Define the function $m\colon \omega \to \omega$ by

$$m(x) = (\mu k)(\forall n)_{\geq k}[g_n(x) = g_k(x)].$$

Then m is a modulus function for the sequence, and so $f(x) = g_{m(x)}(x)$ for all $x \in \omega$. Since $g \leq_T A$ we can conclude $f \leq_T A'$ once we show that $m \leq_T A'$. Well, we have

$$\{(k,x)\colon (\forall n)_{\geq k}[g_n(x) = g_k(x)]\} \equiv_T \{(k,x)\colon (\exists n)_{\geq k} g_n(x) \neq g_k(x)\},$$

and this last set is A-r.e. since $g \leq_T A$. Thus these two sets are recursive in A', and so m is also recursive in A'. □

Corollary 3.7. *$B \leq_T A'$ if and only if there is an A-recursive sequence $\{C_n\}_{n\in\omega}$ of sets (which we can take to be finite) whose limit is B, that is $k \in B$ if and only if $\exists n_0(\forall n)_{\geq n_0}[k \in C_n]$. We will have $C_n \subseteq B$ for all n if and only if B is A-r.e.*

4. Degrees

Sets A and B are Turing equivalent if and only if deciding membership in A is precisely as difficult as deciding membership in B. In this case A and B are said to have the same *degree* (of unsolvability). Degrees are thus equivalence classes under $\equiv_T$, and Turing reducibility induces a partial order on the degrees.

Definition 4.1.

(1) If $A \subseteq \omega$ then the *(Turing) degree* of A is $[A]_T = \{B \subseteq \omega\colon A \equiv_T B\}$.
(2) $\mathcal{D} = \{[A]_T\colon A \subseteq \omega\}$.

NOTATION: Degrees are denoted by $\mathbf{a}$, $\mathbf{b}$, etc.

(3) If $\mathbf{a}, \mathbf{b} \in \mathcal{D}$ then $\mathbf{a} \leq \mathbf{b}$ if and only if $A \leq_T B$ for some (equivalently every) $A \in \mathbf{a}, B \in \mathbf{b}$; $\mathbf{a} < \mathbf{b}$ means that $\mathbf{a} \leq \mathbf{b}$ but $\mathbf{a} \neq \mathbf{b}$ (equivalently, $A <_T B forsome, orevery, A \in \mathbf{a}, B \in \mathbf{b}$).
(4) If $\mathbf{a} \in \mathcal{D}$ then $\mathbf{a}' = [A']_T$ for some (equivalently, every) $A \in \mathbf{a}$.
(5) $\mathbf{0} = [\varnothing]_T$; $\mathbf{0}^{(n)} = [\varnothing^{(n)}]_T$.

Thus the degrees are a classification of all sets of natural numbers according to computational complexity. One studies the structure of $\mathcal{D}$ with $\leq$ and $'$ to better understand the relative computational complexity of subsets of ω.

We first collect some elementary properties of the degrees in the following proposition and then state without proof some more difficult results:

Proposition 4.1.

(i) *$\leq$ is a partial order of $\mathcal{D}$.*
(ii) *$\mathbf{0} \leq \mathbf{a}$ for every $\mathbf{a} \in \mathcal{D}$.*
(iii) *Any finite set of degrees has a least upper bound in $\mathcal{D}$.*

(iv) $|\mathcal{D}| = 2^\omega$.
(v) *For any* $\mathbf{a} \in \mathcal{D}$, $|\{\mathbf{b} \in \mathcal{D} \colon \mathbf{b} \leq \mathbf{a}\}| \leq \omega$.
(vi) *Any countable set of degrees has an upper bound in* $\mathcal{D}$.
(vii) *For any* $\mathbf{a} \in \mathcal{D}$, $\mathbf{a} < \mathbf{a}'$, *and so there are no maximal degrees in the ordering.*
(viii) *If* $\mathbf{a} \leq \mathbf{b}$ *then* $\mathbf{a}' \leq \mathbf{b}'$.
(ix) $\mathcal{D}$ *is not linearly ordered by* $\leq$; *in fact there are* $\mathbf{a}, \mathbf{b} < \mathbf{0}'$ *such that* $\mathbf{a} \nleq \mathbf{b}$ *and* $\mathbf{b} \nleq \mathbf{a}$.

Proof. (i), (ii), (vii), and (viii) are immediate from properties of $\leq_T$ and the jump.

To show (iii) it suffices to produce a least upper bound for any two degrees, $\mathbf{a}, \mathbf{b}$. Choosing $A \in \mathbf{a}$ and $B \in \mathbf{b}$, choose $C \in \omega$ so that $K_C = K_A \oplus K_B$ as defined in Section 1. It is then easy to verify that $\mathbf{c} = [C]_T$ is the desired least upper bound.

Just as there are only countably many recursive functions there are only countably many functions recursive in any given A. Thus $|\{B \subseteq \omega \colon B \leq_T A\}| = \omega$, which establishes (v). In addition, since $|\mathcal{P}(\omega)| = 2^\omega$, (iv) follows.

We leave (vi) to the reader—note, however, that (v) implies that no uncountable set of degrees has an upper bound.

If $\mathcal{D}$ were linearly ordered by $\leq$ then we would have $|\mathcal{D}| \leq \omega_1$ by (v), and so $2^\omega = \omega_1$ by (iv). While not quite a contradiction this is surely suspicious. In the next section we will establish (ix) by showing that there are incomparable degrees $< \mathbf{0}'$. □

Thus $(\mathcal{D}, \leq)$ forms what is called an upper semi-lattice with least element ((i),(ii),(iii)). $(\mathcal{D}, \leq)$ fails to be a lattice since in general greatest lower bounds of pairs of elements fail to exist.

Similarly, (vi) cannot be improved to yield *least* upper bounds for countably infinite sets of degrees. In fact, if $\mathbf{a}_n < \mathbf{a}_{n+1}$ for all $n \in \omega$ then $\{\mathbf{a}_n \colon n \in \omega\}$ has no least upper bound. In particular $\{\mathbf{0}^{(n)} \colon n \in \omega\}$ has no least upper bound.

$(\mathcal{D}, \leq)$ is about as complicated as possible. Every countable partial ordering embeds isomorphically into $(\mathcal{D}, \leq)$. But this partial ordering is far from homogeneous. Thus there are minimal degrees (that is, degrees $\mathbf{a}$ such that $\mathbf{0} < \mathbf{a}$ but there is no $\mathbf{b} \in \mathcal{D}$ with $\mathbf{0} < \mathbf{b} < \mathbf{a}$). But there also are degrees $\mathbf{c} > \mathbf{0}$ such that no degree $\mathbf{a} \leq \mathbf{c}$ is minimal—in fact there are degrees $\mathbf{c} > \mathbf{0}$ such that $\{\mathbf{b} \colon \mathbf{b} < \mathbf{c}\}$ is densely and linearly ordered. More generally, every contable upper semi-lattice with least element is isomorphic to an initial segment of $\mathcal{D}$ (i.e., to M such that $\mathbf{a} \in M, \mathbf{b} \leq \mathbf{a}$ implies $\mathbf{b} \in M$). Not surprisingly, $Th((\mathcal{D}, \leq))$ is undecidable.

One might hope that a more "familiar" set like $\mathcal{D}(\leq \mathbf{0}') = \{\mathbf{a} \in \mathcal{D} \colon \mathbf{a} \leq \mathbf{0}'\}$ might be simpler. But there are minimal degrees in $\mathcal{D}(\leq \mathbf{0}')$, and every countable partial order embeds isomorphically into $\mathcal{D}(\leq \mathbf{0}')$.

5. Degree Constructions

In this section we present a general method of constructing sets whose degrees have certain specified relations. We illustrate the method by showing that there are incomparable degress less than $\mathbf{0}'$.

For our first approximation we ignore the requirement that we construct sets with degrees less than $\mathbf{0}'$ and just see what is involved in constructing sets with incomparable degrees. Thus, we wish to construct $A, B \in \omega$ such that $A \nleq_T B$

and $B \not\leq_T A$. This will be guaranteed provided A and B satisfy all of the following requirements R_e and S_e for $e \in \omega$:

$$R_e\colon K_A \neq \{e\}^B,$$
$$S_e\colon K_B \neq \{e\}^A.$$

Of course, R_e is satisfied by making sure there is some $n \in \omega$ with $K_A(n) \neq \{e\}^B(n)$, and similarly for S_e. But while we are still performing the construction we don't yet know what B is—so how can we know the value $\{e\}^B(n)$?

The solution to this difficulty is given by Lemma 2.4. $\{e\}^B(n) = l$ if and only if $\exists s \{e\}^B_s(n) = l$ *and* in this case we will also have $\{e\}^C(n) = l$ for every $C \subseteq \omega$ with $(C \cap S) = (B \cap S)$. Thus the value l to which $\{e\}^B(n)$ converges (assuming there is one) is determined by a sufficiently large *finite* piece of B—which we *will* know at some stage in the construction.

With this introduction we can now plunge into the proof of the first version of the theorem.

THEOREM 5.1. *There are degrees* $\mathbf{a}$ *and* $\mathbf{b}$ *such that* $\mathbf{a} \not\leq \mathbf{b}$ *and* $\mathbf{b} \leq \mathbf{a}$.

PROOF. We wish to define $A, B \subseteq \omega$ such that R_e and S_e above are satisfied for all $e \in \omega$. We define, by simultaneous recursion on ω, functions $f_t, g_t \in {}^{\underset{\smile}{\omega}}2$ such that $f_t \subseteq f_{t+1}$ and $g_t \subseteq g_{t+1}$ for all $t \in \omega$ and such that $\bigcup_{t\in\omega} f_t = K_A$ and $\bigcup_{t\in\omega} g_t = K_B$ yield the desired A, B.

To begin with, set $f_0 = g_0 = \varnothing$. Now, given $f_t, g_t \in {}^{\underset{\smile}{\omega}}2$ we show how to define f_{t+1} and g_{t+1}.

Case 1: $t = 2e$.
We guarantee that R_e will hold. Let $n = \mathrm{dom}(f_t)$.

Subcase (i): There is $G\colon \omega \to 2$ such that $g_t \subseteq G$ and $\{e\}^G(n)\downarrow$.
In this case choose some such G, choose some $s \in \omega$ such that $\{e\}^G_s(n)\downarrow$ and $s \geq \mathrm{dom}(g_t)$. Now define $g_{t+1} = G \restriction s$ and $f_{t+1} = f_t \cup \{(n, 1 \dot{-} \{e\}^G(n))\}$. Then no matter how the rest of the construction is carried out, in the end we will have $\{e\}^B(n) = \{e\}^G(n) \neq K_A(n)$, thus guaranteeing R_e.

Subcase (ii): There is no $G\colon \omega \to 2$ such that $g_t \subseteq G$ and $\{e\}^G(n)\downarrow$.
In this case no matter how the rest of the construction is carried out we will have $\{e\}^B(n)\uparrow$ and so R_e will necessarily hold. So we define $g_{t+1} = g_t$ and $f_{t+1} = f_t \cup \{(n, 0)\}$.

Case 2: $t = 2e + 1$.
In this case we proceed with the roles of f_t, g_t interchanged to guarantee that S_e holds.

Note that we have guaranteed that $\bigcup_{t\in\omega} f_t\colon \omega \to 2$ and $\bigcup_{t\in\omega} g_t\colon \omega \to 2$ so they will be some K_A, K_B. $\square$

Let's examine this proof to see what is really going on and determine exactly what needs to be done to find such A, B with $A, B \leq_T \varnothing'$.

First of all, what we are doing is defining a function by an informal *primitive recursion* whose value at t is the pair (f_t, g_t). To formalize this definition, the value of the function must be a natural number. We accomplish this by replacing finite functions in ${}^{\underset{\smile}{\omega}}\omega$ by the sequence number of the sequence of their values.

Definition 5.1. Let $h \in {}^{\underset{\smile}{\omega}}\omega$ and let $n = \mathrm{dom}(h)$. Then $\widehat{h} = \langle h(0), \ldots, h(n-1)\rangle$ is the sequence number coding h.

In this context we will use σ, τ to stand for sequence numbers, and we write $\sigma \subseteq \tau$ to mean σ is an initial segment of τ. Thus, if $g, h \in {}^{\underset{\smile}{\omega}}\omega$ then $g \subseteq h$ if and only if $\widehat{g} \subseteq \widehat{h}$.

So, in the proof of this theorem we are defining a function H by primitive recursion so taht $H(t) = \langle \widehat{f}_t, \widehat{g}_t\rangle$ with f_t, g_t as previously. Our goal is to ensure that $\bigcup_{t\in\omega} f_t$ and $\bigcup_{t\in\omega} g_t$ are both $\leq_T \varnothing'$. This will be guaranteed by having $H \leq_T \varnothing', Seq^* = \{\widehat{h} \colon h \in {}^{\underset{\smile}{\omega}}2\}$.

Now, to guarantee that $H \colon \omega \to \omega$ defined by primitive recursion is $\leq_T \varnothing'$ it suffices to show that the function which defines $H(t+1)$ from t and $H(t)$ is $\leq_T \varnothing'$. Thus we need to show that the division into cases and subcases, and the definitions of $\widehat{f}_{t+1}$ and $\widehat{g}_{t+1}$ from $\widehat{f}_t$ and $\widehat{g}_t$ in each case, are all $\leq_T \varnothing'$.

Finally, let $R \subseteq \omega^4$ be the recursive relation from the Enumeration Theorem 2.2. We introduce the following notation:

Definition 5.2. $\{e\}^\sigma(x) = y$ if and only if $R(e, x, y, \sigma)$ and $e, x, y < \mathrm{lh}(\sigma)$.

Lemma 5.2.

(1) $\{e\}^\sigma(x) = y$ *is a recursive relation in the four (number) variables* e, σ, x, y.
(2) $\{e\}^f_s(x) = y$ *if and only if* $\{e\}^\sigma(x) = y$, *where* $\sigma = \overline{f}(s)$.
(3) $\{e\}^\sigma(x) = y$ *implies that* $\{e\}^f(x) = y$ *for every* $f \colon \omega \to \omega$ *with* $\sigma = \overline{f}(s)$ *for* $s = \mathrm{lh}(\sigma)$.

We can now indicate how to adapt the proof of Theorem 5.1 to obtain:

Theorem 5.3. *There are degrees* $\mathbf{a}, \mathbf{b}$ *with* $\mathbf{a} \leq \mathbf{0}'$, $\mathbf{b} \leq \mathbf{0}'$, $\mathbf{a} \not\leq \mathbf{b}$, $\mathbf{b} \not\leq \mathbf{a}$.

Proof. As indicated above, it suffices to consider the way in which $\langle \widehat{f}_{t+1}, \widehat{g}_{t+1}\rangle$ is obtained from $\langle \widehat{f}_t, \widehat{g}_t\rangle$. the division into Case 1 and Case 2 is recursive. By Lemma 5.2, Subcase (i) holds if and only if

$$\exists\sigma(Seq^*(\sigma) \wedge \widehat{g}_t \subseteq \sigma \wedge \exists y[\{e\}^\sigma(n) = y]),$$

and this condition is r.e., hence $\leq_T \varnothing'$. Similarly, of course, Subcase (ii)—being the negation of Subcase (i)—is co-r.e., hence $\leq_T \varnothing'$.

In Subcase (i) we define

$$\widehat{g}_{t+1} = ((\mu w)[Seq^*((w)_0) \wedge \widehat{g}_t \subseteq (w)_0 \wedge \{e\}^{(w)_0}(n) = (w)_1])_0.$$

In other words, we look for the least pair $\langle \sigma, l\rangle$ where $Seq^*(\sigma) \wedge \widehat{g}_t \subseteq \sigma \wedge \{e\}^\sigma(n) = l$ and use σ for $\widehat{g}_{t+1}$. This choice of $\widehat{g}_{t+1}$ is, of course, recursive, and $\widehat{f}_{t+1}$ is recursively defined from it as in the proof of Theorem 5.1.

Subcase (ii) is even easier, of course, and Case 2 is analogous to Case 1. Thus the function given $\langle \widehat{f}_{t+1}, \widehat{g}_{t+1}\rangle$ in terms of t and $\langle \widehat{f}_t, \widehat{g}_t\rangle$ is $\leq_T \varnothing$ as required, completing the proof. $\square$

6. Exercises

(1)

CHAPTER 12

The Arithmetic Hierarchy

0. Introduction

In this chapter we study the subsets of ω which are first-order definable in the structure $\mathfrak{N}$. We define a hierarchy on these sets, roughly speaking by the quantifier complexity of the defining formula. The recursive sets are at the lowest level of the hierarchy, followed by the r.e. sets (which are definable using one quantifier in front of a recursive relation), etc. The most important result in this chapter is Post's Theorem 2.1, which ties this hierarchy to Turing reducibility.

1. Arithmetic Relations and the Hierarchy

Recall that $\mathfrak{N}$ is the "standard" structure on ω for the language $\mathcal{L} = \{+, \cdot, <, \overline{0}, s\}$.

Definition 1.1. A relation $R \subseteq \omega^n$ is *arithmetic* if and only if R is definable in $\mathfrak{N}$ by some $\mathcal{L}$-formula.

We define, by recursion on n, an indexed family of sets of relations on ω, which include precisely the arithmetic relations.

Definition 1.2.

(a) The collections Σ_n, Π_n are defined simultaneously via:
  - $\Sigma_0 = \Pi_0 =$ the set of all recursive relations;
  - $R \in \Sigma_{n+1}$ if and only if there is some $S \in \Pi_n$ such that $R(\vec{x}) \leftrightarrow \exists y S(\vec{x}, y)$;
  - $R \in \Pi_{n+1}$ if and only if there is some $S \in \Sigma_n$ such that $R(\vec{x}) \leftrightarrow \forall y S(\vec{x}, y)$.

(b) $\Delta_n = (\Sigma_n \cap \Pi_n)$.

The following lemma summarizes most of the elementary properties of this hierarchy:

Lemma 1.1.

i *$R \in \Sigma_n$ if and only if $\neg R \in \Pi_n$.*
ii *$(\Sigma_n \cup \Pi_n) \subseteq \Delta_{n+1}$.*
iii *Σ_n is closed under $\wedge$, $\vee$, $\exists y$, $(\forall y)_{<x}$, for all $n > 0$.*
iv *Π_n is closed under $\wedge$, $\vee$, $\forall y$, $(\exists y)_{<x}$, for all $n > 0$.*
v *Given $R \subseteq \omega^k$ define*

$$R^* = \{\langle m_1, \ldots, m_k \rangle \colon R(m_1, \ldots, m_k) \text{ holds}\}.$$

Then $R^* \subseteq \omega$ and $R^* \in \Sigma_n$ (or Π_n or Δ_n) if and only if $R \in \Sigma_n$ (or Π_n or Δ_n).

vi *R is arithmetic if and only if $R \in \Sigma_n$ for some $n \in \omega$.*

vii $A \in \Sigma_n$, $B \leq_m A \Rightarrow B \in \Sigma_n$; *same for* Π_n.

Note that $R \in \Sigma_1$ if and only if R is r.e., and thus $R \in \Delta_1$ if and only if R is recursive. Thus $\varnothing' \in (\Sigma_1 \setminus \Pi_1)$.

EXAMPLES:

(1) $\mathrm{TOT} = \{e \colon W_e = \omega\} \in \Pi_2$, since $e \in \mathrm{TOT}$ if and only if $\forall x \{e\}(x) \downarrow$ and we know $\{e\}(x) \downarrow$ is r.e., i.e., Σ_1.
(2) $\mathrm{FIN} = \{e \colon W_e \text{ is finite}\} \in \Sigma_2$, since $e \in \mathrm{FIN}$ if and only if $\exists x \forall y [x < y \to \{e\}(y) \uparrow]$.
(3) $\mathrm{COF} = \{e \colon W_e \text{ is cofinite}\} \in \Sigma_3$.

In each case, we have found the lowest level of the arithmetic hierarchy to which the set in question belongs. This follows from showing the set in question is m-complete, and Post's Theorem 2.1.

DEFINITION 1.3. A is Σ_n-*complete* if and only if $A \in \Sigma_n$ and $B \leq_m A$ whenever $B \in \Sigma_n$. Π_n-*complete* is defined analogously.

PROPOSITION 1.2. TOT *is* Π_2-*complete.*

PROOF. Let $B \in \Pi_2$ so there is some recursive $R \subseteq \omega^3$ such that

$$k \in B \text{ if and only if } \forall x \exists y R(x, y, k).$$

We can then define a partial recursive function g by

$$g(k, x) = \begin{cases} 0 & \text{if } \exists y R(x, y, k), \\ \uparrow & \text{otherwise.} \end{cases}$$

Applying the Parameter Theorem 2.4 we obtain a total recursive s such that $\{s(k)\}(x) = g(k, x)$ for all k, x. In particular, if $k \in B$ then $s(k) \in \mathrm{TOT}$, and if $k \notin B$ then $s(k) \notin \mathrm{TOT}$. $\square$

As a consequence note that $\mathrm{TOT} \notin \Sigma_2$ provided $(\Pi_2 \setminus \Sigma_2) \neq \varnothing$.

2. Post's Theorem

The following fundamental result relates the definability hierarchy just introduced to the jump hierarchy introduced in Section 3.

2.1. Post's Theorem.

(1) $B \in \Sigma_{n+1}$ *if and only if* B *is r.e. in some* Π_n *set (equivalently, in some* Σ_n *set).*
(2) $\varnothing^{(n)}$ *is* Σ_n*-complete for all* $n > 0$.
(3) $B \in \Sigma_{n+1}$ *if and only if* B *is r.e. in* $\varnothing^{(n)}$.
(4) $B \in \Delta_{n+1}$ *if and only if* $B \leq_T \varnothing^{(n)}$.

PROOF. (0) First note that since $A \equiv_T (\omega \setminus A)$ for all A we see that B is r.e. in A if and only if B is r.e. in $(\omega \setminus A)$, and so B is r.e. in some Π_n set if and only if B is r.e. in some Σ_n set.

The implication from left to right in (0) is clear from the definitions. To show the other direction, suppose B is r.e. in A, where $A \in \Pi_n$. Then there is some S which is recursive in A such that

$$k \in B \leftrightarrow \exists y S(k, y).$$

By the Representation Theorem 2.1 there is some recursive relation R such that

$$S(k, y) \leftrightarrow \exists s R(k, y, \overline{K}_A(s)).$$

Expanding on what $\overline{K}_A(s)$ is we see that $k \in B$ if and only if

$$\exists y \exists w[\mathrm{Seq}(w) \wedge R(k, y, w) \wedge (\forall i)_{<\mathrm{lh}(w)}([(w)_i = 0 \wedge (w)_i \in A] \vee [(w)_i = 1 \wedge (w)_i \notin A])].$$

$(w)_i \in A$ is Π_n, and $(w)_i \notin A$ is Σ_n, hence they are both Σ_{n+1}, so the entire right-hand side is Σ_{n+1}, and thus $B \in \Sigma_{n+1}$.

(1) We proceed by induction on n. We already know the case $n = 1$, so we can assume $\varnothing^{(n)}$ is Σ_n-complete, some $n > 0$, and prove that $\varnothing^{(n+1)}$ is Σ_{n+1}-complete. First of all, $\varnothing^{(n+1)}$ is r.e. in $\varnothing^{(n)}$ and $\varnothing^{(n)} \in \Sigma_n$ so $\varnothing^{(n+1)} \in \Sigma_{n+1}$ by part (0) of this theorem. Now let $B \in \Sigma_{n+1}$. Then, also by part (0), B is r.e. in some Σ_n set, hence in $\varnothing^{(n)}$ by inductive hypothesis, and thus $B \leq_m \varnothing^{(n+1)}$ by properties of the jump.

(2) follows immediately from (0) and (1).

(3) follows from (2). $\square$

We collect some consequences of Theorem 2.1:

i $\varnothing^{(n+1)} \in (\Sigma_{n+1} \setminus \Pi_{n+1})$, all n.
ii A is arithmetic if and only if $A \leq_T \varnothing^{(n)}$ for some n.
iii If A is Σ_{n+1}-complete then $A \notin \Pi_{n+1}$.
iv $(\Sigma_n \cup \Pi_n) \subsetneq \Delta_{n+1}$ for all $n > 0$.
v If A is arithmetic and $B \leq_T A$ then B is arithmetic.
vi There is no arithmetic A such that $B \leq_T A$ for all arithmetic B.
vii $\{\ulcorner \sigma \urcorner : \sigma \in \mathrm{Sn}_{\mathcal{L}}, \mathfrak{N} \models \sigma\}$ is not arithmetic.
viii If A is Σ_n-complete or Π_n-complete then $A \equiv_T \varnothing^{(n)}$.

3. Exercises

(1)

APPENDIX A

Appendix A: Set Theory

The natural numbers (that is, non-negative integers) are used in two very different ways. The first way is to *count* the number of elements in a (finite) set. The second way is to *order* the elements in a set–in this way one can prove things about all the elements in the set *by induction*. These two roles of natural numbers both are generalized to infinite numbers, but these are split into two groups according to the function they perform: cardinal numbers (to count) and ordinal numbers (to order). The basic facts and concepts are surveyed in this Appendix.

1. Cardinals and Counting

It was noted as long ago as Galileo that (some) infinite sets can be put into one-to-one correspondence with proper subsets of themselves, and thus, for example, that the set of integers may be considered as having the same "size" as the set of even integers. However no serious investigation into the "sizes" of infinite sets, on comparing them, was undertaken until Cantor, in the second half of the 19^{th} century, created set theory, including both cardinal and ordinal numbers.

The basic definitions about comparing the sizes of sets (including finite) are as follows:

DEFINITION 1.1. (i) $X \sim Y$ (X and Y are *equivalent*, or have the "same number" of elements) iff there is some function mapping X one-to-one onto Y. (ii) $X \preceq Y$ (X has "at most as many" elements as Y) iff there is some function mapping X one-to-one into Y. (iii) $X \prec Y$ (X has strictly "fewer" elements than Y) iff $X \preceq Y$ but not $X \sim Y$.

The following proposition is certainly essential if one is to think of $\preceq$ as some sort of ordering on sets. It was not proved, however, until the end of the 19^{th} century.

PROPOSITION 1.1. *If $X \preceq Y$ and $Y \preceq X$ then $X \sim Y$.*

All the basic facts about size comparisons could be expressed by the above notation but this would be quite clumsy. Instead certain sets, called *cardinal numbers*, are picked out so that for every set X there is *exactly one* cardinal number κ such that $X \sim \kappa$. We then call κ the *cardinality* of X and write $|X| = \kappa$. $|X| \leq |Y|$ means $X \preceq Y$ and $|X| < |Y|$ means $X \prec Y$. Notice that $|\kappa| = \kappa$ if, and only if, κ is a cardinal number.

The first cardinal numbers are defined as follows:

$$
\begin{aligned}
0 &= \emptyset, \\
1 &= \{0\}, \\
2 &= 1 \cup \{1\}, \\
&\vdots \\
n+1 &= n \cup \{n+1\} \\
&\vdots
\end{aligned}
$$

$\omega = \{0, 1, \ldots, n, \ldots\}$ is defined as the smallest set containing 0 and such that if $x \in \omega$ then $x \cup \{x\} \in \omega$.

Notice that $\omega+1 = \omega \cup \{\omega\}$ cannot also be a cardinal number since $\omega \sim \omega \cup \{\omega\}$.

Definition 1.2. (a) X is *finite* iff $X \sim n$ for some $n \in \omega$ ($|X| \in \omega$). (b) X is *countable* iff $X \preceq \omega$ (i.e. $|X| \leq \omega$).

Lemma 1.2. *(i) X is finite iff $|X| < \omega$. (ii) X is countable and infinite iff $|X| = \omega$.*

[The essential content of this lemma is that all cardinals less than ω in fact belong to ω].

One of Cantor's fundamental discoveries is that there are infinite sets which are not equivalent, and in fact that there can be no biggest cardinal number.

Definition 1.3. The *power set* of X, is defined by $\mathcal{P}(X) = \{Y | \ Y \subseteq X\}$.

Theorem 1.3. *For ever X, $X \prec \mathcal{P}(X)$.*

Proof. Obviously, $X \preceq \mathcal{P}(X)$. Suppose that $X \sim \mathcal{P}(X)$, say that h maps X bijectively onto $\mathcal{P}(X)$. Let $D = \{x \in X | \ x \notin h(x)\}$. Then $D = h(d)$ for some $d \in X$. But $d \in D$ iff $d \notin h(d) = D$, which is a contradiction. $\square$

Thus, there must be cardinals κ such that $\omega < \kappa$. We must put off defining them, however, until after we introduce ordinal numbers–we also should admit that we will need the Axiom of Choice (AC) to define cardinal numbers in general. Recall that AC states that if X is a set of non-empty sets, then there is a function f defined on X such that $f(x) \in x$ for every $x \in X$.

It is important to know that many set-theoretic operations lead from countable sets to countable sets.

Definition 1.4. (a) Y_X is the set of all functions f with domain Y and range a subset of X. (b) $\overset{\omega}{\smile}X = \bigcup_{n \in \omega} n_X$ is the set of all finite sequences of elements of X (thinking of a sequence of length n as a function defined on n); an alternate notation is ${}^{\omega>}X$.

Theorem 1.4. *If X is countable then so is $\overset{\omega}{\smile}X$.*

Proof. It suffices to show that $\overset{\omega}{\smile}\omega \preceq \omega$, which follows by using the one-to-one map which sends $(k_0, \ldots, k_{n-1})$ to $2^{k_0+1} \cdot 3^{k_1+1} \cdot \cdots \cdot p_{n-1}^{k_{n-1}+1}$ where p_j is the j^{th} odd prime. $\square$

Corollary 1.5. *If X, Y are countable so are $X \cup Y$ and $X \times Y$.*

THEOREM 1.6. *(AC) If X_n is countable for every $n \in \omega$ then $\bigcup_{n\in\omega} X_n$ is also countable.*

ω is, in fact, the the *smallest* infinite cardinal, although the proof requires the axiom of choice.

PROPOSITION 1.7. *(AC) If X is infinite then $\omega \preceq X$.*

The analogues of $+$ and $\cdot$ trivialize on infinite cardinals because of the preceding corollary, but exponentiation is important.

NOTATION 1. If κ, λ are cardinal numbers then κ^λ is the cardinal $|\lambda_\kappa|$.

LEMMA 1.8. *For any X, $\mathcal{P}(X) \sim X_2$.*

Hence from Cantor's Theorem we see the following corollary.

COROLLARY 1.9. *If $|X| = \kappa$ then $|\mathcal{P}(X)| = 2^\kappa$, and so $\kappa < 2^\kappa$ for every cardinal κ.*

However, increasing the base does not yield still larger cardinals.

LEMMA 1.10. *$2^\omega = n^\omega = \omega^\omega = (2^\omega)^\omega$, any $n \in \omega$.*

PROOF. It suffices to show $(2^\omega)^\omega \leq 2^\omega$, which follows since

$$\omega_{(\omega_2)} \sim (\omega \times \omega)_2 \sim \omega_2.$$

$\square$

Without proof we list some facts about (uncountable) cardinalities, all depending on AC.

(1) If X is infinite then $|X| = |{}^{\underset{\smile}{\omega}}X|$.
(2) If X, Y are infinite then $|X \cup Y| = |X \times Y| = max(|X|, |Y|)$.
(3) If $|I| \leq \kappa$ and $|X_i| \leq \kappa$ for all $i \in I$, then $|\bigcup_{i\in I} X_i| \leq \kappa$, for $\kappa \geq \omega$.
(4) $(\mu^\kappa)^\lambda = \mu^{max(\kappa,\lambda)}$, for $\kappa, \lambda \geq \omega$, $\mu \geq 2$.
(5) For any cardinal κ there is a unique *next* cardinal called κ^+, but there is no set X such that $\kappa \preceq X \preceq \kappa^+$.
(6) If X is a non-empty set of cardinal numbers, then $\bigcup X$ is a cardinal number and it is the first cardinal $\leq$ all cardinals in X.
(7) $(\kappa^+)^\lambda = max(\kappa^\lambda, \kappa^+)$ for $\kappa, \lambda \geq \omega$.
(8) For any sets X, Y either $X \preceq Y$ or $Y \preceq X$, hence for any cardinals κ, λ either $\kappa \leq \lambda$ or $\lambda \leq \kappa$.

Some notation, based on (5), is the following which we will extend in the next section: $\omega_1 = \omega^+, \omega_{n+1} = \omega_n^+, \omega_\omega = \bigcup_{n\in\omega} \omega_n$–writing also $\omega_0 = \omega$. An alternate notation is to use the Hebrew letter "aleph"–thus $\aleph_0, \aleph_1, \ldots, \aleph_\omega, \ldots$.

Note that $\omega_1 \leq 2^\omega$ and, in general, $\kappa^+ \leq 2^\kappa$ for each $\kappa \geq \omega$. It is natural to enquire about whether equality holds or not.

CONJECTURE 1.1. **Continuum Hypothesis** (CH): $2^\omega = \omega_1$

CONJECTURE 1.2. **Generalized Continuum Hypothesis** (GCH): For every infinite cardinal κ, $2^\kappa = \kappa^+$.

CH and GCH are consistent with, but independent of, the usual axiioms of set theory. In fact, each of the following is consistent with the usual axioms:

$$2^\omega = \omega_1, 2^\omega = \omega_2, 2^\omega = \omega_n \quad \text{for any} n \in \omega$$
$$2^\omega = \omega_\omega)+, 2^\omega = (\omega_\omega)^{++}, \ldots$$

We can, however, prove that $2^\omega \neq \omega_\omega$ since we have $(\omega_\omega)^\omega > \omega_\omega$.

Some further facts about cardinals will be presented at the end of the next section.

2. Ordinals and Induction

The principles of proof by induction and definition by recursion on the natural numbers are consequences just of the fact that ω is well-ordered by the usual order.

DEFINITION 2.1. $(X, \leq)$ is a well-ordering iff $\leq$ is a linear order of X and every non-empty subset $Y \subseteq X$ contains a least element, i.e. there is some $a_0 \in Y$ such that $a_0 \leq a$ for all $a \in Y$.

THEOREM 2.1. *(Proof by Induction) Let $(X, \leq)$ be a well-ordering. Let $A \subseteq X$ have the property that for every $a \in X$, if $b \in A$ for all $b < a$ then $a \in A$. Then $A = X$.*

PROOF. If not, consider $Y = X - A$ and obtain a contradiction to the definition. □

The way this is used if one wants to prove that all elements of X have property P is to let A be the set of all elements of X having property P.

In a similar vein, we see:

THEOREM 2.2. *(Definition by Recursion) Let $(X, \leq)$ be a well-ordering. Let Y be any non-empty set and let g be a function from $\mathcal{P}(Y)$ into Y. Then there is a unique function f from X into Y such that for every $a \in X$,*

$$f(a) = g(\{f(x)|\ x \in X, x < a\}).$$

[Less formally, this just says that $f(a)$ is defined in terms of the $f(x)$'s for $x < a$.]

As in the previous section, we wish to pick out particular well-orderings, called *ordinal numbers*, such that each well-ordering is isomorphic to *exactly one* ordinal number. We do this so that the well order $\leq$ of the ordinal is as natural as possible–that is, is give by $\in$. The precise definition we obtain is as follows:

DEFINITION 2.2. A set X is an ordinal number iff (i) $x \in y \in X \Rightarrow x \in X$ (equivalently, $y \in X \Rightarrow y \subseteq X$), and (ii) X is well-ordered by the relation $\leq$ defined by: $a \leq b$ iff $a \in b$ or $a = b$.

Condition (i) is frequently expressed by saying "X is transitive" and condition (ii) is loosely expressed by saying "$\in$ well-orders X." Note that technically X is not a well-ordering, but $(X, \leq)$ is–however condition (ii) determines $\leq$ completely from X. Notice, of course, that most sets aren't even *linearly* ordered by $\in$–in fact, one of the usual (but somewhat technical) axioms of set theory implies that if X is linearly ordered by $\in$, then in fact it is well-ordered by $\in$. Thus the conditions in (ii) could be expanded to read: (ii)*: $x \in y, y \in z, z \in X \Rightarrow x \in z$, $x, y \in X \Rightarrow x = y \vee x \in y \vee y \in x$. $(x \notin x)$ follows by the usual axioms.

Notice that the finite cardinal numbers and ω, as defined in the previous section, are also ordinal numbers. The following lemma gives some of the basic properties of ordinals. By convention, we normally use Greek letters $\alpha, \beta, \ldots$ to stand for ordinals.

LEMMA 2.3. *(1) If α is an ordinal and $x \in \alpha$ then x is an ordinal. (2) If α, β are ordinals then either $\alpha \in \beta$ or $\alpha = \beta$ or $\beta \in \alpha$. (3) If α is an ordinal then $\alpha + 1 = \alpha \cup \{\alpha\}$ is an ordinal. (4) If X is a set of ordinals then $\bigcup X$ is an ordinal.*

NOTATION 2. If α, β are ordinals we write $\alpha < \beta$ for $\alpha \in \beta$. Part (1) of the lemma states that if α is an ordinal then $\alpha = \{\beta |\ \beta \text{ is an ordinal}, \beta < \alpha\}$. The ordinal $\alpha + 1$ is the *immediate successor* of α–that is, $\alpha < \alpha + 1$ and there is no ordinal β such that $\alpha < \beta < \alpha + 1$. Similarly, $\bigcup X$ is the least upper bound of the set X of ordinals.

The class of all ordinals is not a set, but we can still think of it as well-ordered by $\leq$. Further, we can prove things about the class of all ordinals by induction, and define functions on ordinals by recursion.

Finally we note that ordinals do have the property for which we introduced them.

THEOREM 2.4. *Let $(X, \leq)$ be a well-ordering. Then there is exactly one ordinal α such that $(X, \leq) \cong (\alpha, \leq)$.*

We distinguish between two types of non-zero ordinals as follows:

DEFINITION 2.3. α is a successor ordinal iff $\alpha = \beta + 1$ for some ordinal β; α is a limit ordinal iff $\alpha \neq 0$ and α is not a successor ordinal.

Note that α is a limit ordinal iff $\alpha \neq 0$ and $\bigcup \alpha = \alpha$. If X is any non-empty set of ordinals not containing a largest ordinal, then $\bigcup X$ is a limit ordinal.

It is frequently more convenient to break proofs by induction, or definitions by recursion, into cases according to whether an ordinal is a successor or a limit ordinal. For example, the recursive definition of ordinal addition is as follows:

if $\beta = 0$ then $\alpha + \beta = \alpha$,
if $\beta = \gamma + 1$ then $\alpha + \beta = (\alpha + \gamma) + 1$,
if β is a limit then $\alpha + \beta = \bigcup \{\alpha + \gamma |\ \gamma < \beta\}$.

While most linear orderings $(X, \leq)$ are not well-orderings, there is no restriction on the sets X in well-orderings, by the next theorem. This means that proof by induction can (in principle) be applied to any set.

THEOREM 2.5. *(AC) For every set X there is some $\leq$ which well-orders X.*

As an immediate consequence of the two preceeding theorems we have:

COROLLARY 2.6. *(AC) For every set X there is some ordinal α such that $X \sim \alpha$.*

The ordinal α is not unique unless $\alpha < \omega$, since if $\omega \leq \alpha$ then $\alpha \sim \alpha + 1$, but the *least* such ordinal will be the *cardinality* of X.

DEFINITION 2.4. κ is a cardinal number iff κ is an ordinal number and for every $\alpha < \kappa$ we have $\alpha \prec \kappa$ (equivalently, for every ordinal α such that $\alpha \sim \kappa$ we have $\kappa \leq \alpha$).

This fills the lacuna in the preceding section. Note that the cardinal numbers are well-ordered by $\leq$, and $<$ is $\in$ on cardinal numbers.

The way we will customarily use proof by induction on an arbitrary set X is as follows: let $|X| = \kappa$ so there is some one-to-one function h mapping κ onto X. Write x_α for $h(\alpha)$. Then $X = \{x_\alpha | \ \alpha < \kappa\}$ and we prove what we want about x_α by induction on $\alpha < \kappa$. Note that for each $\alpha < \kappa$ we have $|\{x_\beta | \ \beta < \alpha\}| = \alpha < \kappa$.

The class of infinite cardinals can be indexed by the class of ordinals by using the following definition by recursion:

$\omega(0) = \omega$,
$\omega(\gamma + 1) = (\omega(\gamma))^+$,
β a limit $\Rightarrow \omega(\beta) = \bigcup\{\omega(\gamma) | \ \gamma < \beta\}$.

We normally write ω_γ instead of $\omega(\gamma)$.

We finally need to introduce the concept of cofinality in order to make the important distinction between regular and singular cardinals.

Definition 2.5. Let α, β be limit ordinals. Then α is cofinal in β iff there is a strictly increasing function $f \in \alpha_\beta$ such that $\bigcup\{f(\gamma) | \ \gamma < \alpha\} = \beta$.

Certainly β is confinal in β. ω is cofinal in every countable limit ordinal, but ω is not cofinal in ω_1.

Definition 2.6. Let β be a limit ordinal. Then the cofinality of β is $cf(\beta)$ equals the least α such that α is confinal in β.

Lemma 2.7. *For any limit ordinal β, $cf(\beta) \leq \beta$ and $cf(\beta)$ is a cardinal.*

Definition 2.7. Let κ be an infinite cardinal. Then κ is regular iff $\kappa = cf(\kappa)$. κ is singular iff $cf(\kappa) < \kappa$.

Definition 2.8. κ is a successor cardinal iff $\kappa = \lambda^+$ for some cardinal λ, i.e. $\kappa = \omega_{\beta+1}$ for some β.

Definition 2.9. κ is a limit cardinal iff $\kappa \geq \omega$ and κ is not a successor cardinal, i.e., $\kappa = \omega_\alpha$ for some limit ordinal α.

The division of infinite cardinals into regular and singular is almost the same as the division into successor and limit.

Theorem 2.8. *(1) Every successor cardinal is regular. (2) if $\kappa = \omega_\alpha$ is a limit cardinal, then $cf(\kappa) = cf(\alpha)$–hence if κ is regular then $\kappa = \omega_\kappa$.*

Regular limit cardinals are called *inaccessible cardinals*–their existence cannot be proved from the usual axioms of set theory.

With cofinalities we can state a few more laws of cardinal computation, continuing the list from the previous section.

(9) $\kappa^{cf(\kappa)} > \kappa$ for every cardinal $\kappa \geq \omega$.
(10) Assume that $|I| < cf(\kappa)$ and for every $i \in I$, $|X_i| < \kappa$.
Then $|\bigcup_{i \in I} X_i| < \kappa$.

It is frequently tempting to assume GCH because it simplifies many computations, e.g.: Assuming GCH we have, for any cardinals $\kappa, \lambda \geq \omega$, $\kappa^\lambda = \kappa$ if $\lambda < cf(\kappa)$, $\kappa^\lambda = \kappa^+$ if $cf(\kappa) \leq \lambda \leq \kappa$, $\kappa^\lambda = \lambda^+$ if $\kappa \leq \lambda$.

Appendix B: Notes on Validities and Logical Consequence

1. Some Useful Validities of Sentential Logic

1) Excluded Middle
$\models \phi \vee \neg\phi$
$\models \neg(\phi \wedge \neg\phi)$

2) Modus Ponens
$\phi, \phi \to \psi \models \psi$

3) Conjunction
$\phi, \psi \models \phi \wedge \psi$

4) Transitivity of Implication
$\phi \to \psi, \psi \to \theta \models \phi \to \theta$

5) Plain Ol' True as Day
$\phi \wedge \psi \models \phi$
$\phi \models \phi \vee \psi$
$\phi \to (\psi \to \theta), \phi \to \psi \models \phi \to \theta$
$\phi \models \psi \to \phi$
$\neg\psi \models \psi \to \phi$
$\phi \vdash\dashv \neg\neg\phi$

6) Proof by Contradiction
$\neg\phi \to (\psi \wedge \neg\psi) \models \phi$
$\neg\phi \to \phi \models \phi$

7) Proof by Cases
$\phi \to \psi, \theta \to \psi, \models (\phi \vee \theta) \to \psi$
$\phi \to \psi, \neg\phi \to \psi \models \psi$

8) De Morgan's Laws
$\neg(\phi \vee \psi) \vdash\dashv \neg\phi \wedge \neg\psi$
$\neg(\phi \wedge \psi) \vdash\dashv \neg\phi \vee \neg\psi$

9) Distributive Laws
$\phi \wedge (\psi \vee \theta) \vdash\dashv (\phi \wedge \psi) \vee (\phi \wedge \theta)$

$\phi \vee (\psi \wedge \theta) \vdash\dashv (\phi \vee \psi) \wedge (\phi \vee \theta)$

10) Contraposition
$\phi \to \psi \vdash\dashv \neg\psi \to \neg\phi$

11) The connectives $\wedge$ and $\vee$ are both commutative and associative.

2. Some Facts About Logical Consequence

1) $\Sigma \cup \{\phi\} \models \psi$ iff $\Sigma \models \phi \to \psi$

2) If $\Sigma \models \phi_i$ for each $i = 1, \ldots, n$ and $\{\phi_1, \ldots, \phi_n\} \models \psi$ then $\Sigma \models \psi$.

3) $\Sigma \models \phi$ iff $\Sigma \cup \{\neg\phi\}$ is not satisfiable.

Appendix C: Gothic Alphabet

a	$\mathfrak{A}\mathfrak{a}$	b	$\mathfrak{B}\mathfrak{b}$	c	$\mathfrak{C}\mathfrak{c}$	d	$\mathfrak{D}\mathfrak{d}$
e	$\mathfrak{E}\mathfrak{e}$	f	$\mathfrak{F}\mathfrak{f}$	g	$\mathfrak{G}\mathfrak{g}$	h	$\mathfrak{H}\mathfrak{h}$
i	$\mathfrak{I}\mathfrak{i}$	j	$\mathfrak{J}\mathfrak{j}$	k	$\mathfrak{K}\mathfrak{k}$	l	$\mathfrak{L}\mathfrak{l}$
m	$\mathfrak{M}\mathfrak{m}$	n	$\mathfrak{N}\mathfrak{n}$	o	$\mathfrak{O}\mathfrak{o}$	p	$\mathfrak{P}\mathfrak{p}$
q	$\mathfrak{Q}\mathfrak{q}$	r	$\mathfrak{R}\mathfrak{r}$	s	$\mathfrak{S}\mathfrak{s}$	t	$\mathfrak{T}\mathfrak{t}$
u	$\mathfrak{U}\mathfrak{u}$	v	$\mathfrak{V}\mathfrak{v}$	w	$\mathfrak{W}\mathfrak{w}$	x	$\mathfrak{X}\mathfrak{x}$
y	$\mathfrak{Y}\mathfrak{y}$	z	$\mathfrak{Z}\mathfrak{z}$				

Bibliography

Index