

Chapter 5

Connections to Logic

In this chapter, we want to connect concepts of computability with concepts of *definability*. The idea of definability comes from logic. Roughly speaking, one can specify a language, and then study what sets or relations might have exact definitions that can be formulated in that language.

In our case, we will take a language for the arithmetic of natural numbers (that is, number theory). One goal will be to show that every computable set is definable in this language.

This connection between computability and definability has some interesting consequences. For one, it will show that the set of true sentences of arithmetic is very far from being a computable set. And for another, we will come to Gödel's (first) incompleteness theorem. This theorem says that starting from any computable set of true axioms, one cannot possibly hope to derive all the true sentences of arithmetic.

Historically, Gödel's incompleteness theorem (1931) preceded by five years the beginnings of computability theory. But there is an advantage to running history backwards, and looking at Gödel's theorem from the point of view of computability theory.

First, however, we want to build on our characterization of recursively enumerable sets as being the Σ_1 sets. The notation " Σ_1 " already suggests that there ought to be a generalization to Σ_2 and then to Σ_n .

Arithmetical hierarchy

We have defined (on page 404) a relation R on the natural numbers to be Σ_1 if for some computable relation Q , we have

$$\vec{t} \in R \iff \exists x Q(\vec{t}, x)$$

for all \vec{t} . That is, a Σ_1 relation might not be computable, but it is only one quantifier away from computability. We now want to extend this measurement of "distance away from computability."

Define R to be Π_1 if for some computable relation Q , we have

$$\vec{t} \in R \iff \forall x Q(\vec{t}, x)$$

for all \vec{t} . For example, the set \overline{K} is a Π_1 set, because

$$e \in \overline{K} \iff \forall y \overline{T}(e, e, y)$$

where T is the ternary relation

$$T(x, v, t) \iff \llbracket x \rrbracket(v) \downarrow \text{ in } \leq t \text{ steps}$$

⁰Chapters 5, 6, and 7 are largely independent, and can read in any order.

which we know is primitive recursive. Another example of a Π_1 set is the set $\{x \mid W_x = \emptyset\}$ of indices of the empty function. This is Π_1 , because

$$W_x = \emptyset \iff \forall v \forall t \text{ not } T(x, v, t)$$

and we know how to collapse $\forall \forall$ into a single \forall :

$$W_x = \emptyset \iff \forall s \text{ not } T(x, (s)_0, (s)_1)$$

In general, we can say that a relation is Π_1 if and only if it is the complement of a Σ_1 relation. This holds because of the principles

$$\text{not } \exists x \iff \forall x \text{ not} \quad \text{and} \quad \text{not } \forall x \iff \exists x \text{ not,}$$

sometimes called De Morgan's laws. (That is, saying that there does not exist a solution for x is equivalent to saying that every x fails to be a solution. And saying that not all x 's have a property is equivalent to saying that there exists some counterexample x lacking the property.) Using these laws, we have for a Σ_1 relation R ,

$$\begin{aligned} \vec{t} \in \bar{R} &\iff \text{not } \exists x Q(\vec{t}, x) \text{ for computable } Q \\ &\iff \forall x \text{ not } Q(\vec{t}, x) \\ &\iff \forall x \bar{Q}(\vec{t}, x) \end{aligned}$$

so that \bar{R} is Π_1 . Similarly, the complement of a Π_1 relation is Σ_1 .

To use yet another Greek letter, say that R is Δ_1 if it is both Σ_1 and Π_1 . Kleene's theorem tells us that R is Δ_1 if and only if it is a computable relation. For example, the set $\{x \mid W_x = \emptyset\}$ is Π_1 by the above, it is not computable by Rice's theorem, and hence it is not Σ_1 .

We now extend these ideas and define Σ_n and Π_n for each n :

classification	defining condition
Σ_1	$\exists x Q(\vec{t}, x)$
Π_1	$\forall x Q(\vec{t}, x)$
Σ_2	$\exists y \forall x Q(\vec{t}, x, y)$
Π_2	$\forall y \exists x Q(\vec{t}, x, y)$
Σ_3	$\exists z \forall y \exists x Q(\vec{t}, x, y, z)$
Π_3	$\forall z \exists y \forall x Q(\vec{t}, x, y, z)$

where Q is a computable relation. And so forth. (It has been estimated that the human mind cannot grasp the meaning of more than five alternating quantifiers.) We further define a relation to be Δ_n if it is both Σ_n and Π_n .

For example, the set Tot of indices of total computable functions on \mathbb{N} is a Π_2 set, because

$$x \in \text{Tot} \iff \forall v \exists t T(x, v, t)$$

where as before T is the ternary relation

$$T(x, v, t) \iff \llbracket x \rrbracket(v) \downarrow \text{ in } \leq t \text{ steps}$$

which we know is primitive recursive. (More generally, $T^{(n)}$ was defined to be the $(n + 2)$ -ary relation

$$\begin{aligned} T^{(n)}(x, \vec{v}, t) &\iff \llbracket x \rrbracket^{(n)}(\vec{v}) \downarrow \text{ in } \leq t \text{ steps} \\ &\iff (\text{snap}^{(n)}(\vec{v}, x, t))_0 \geq \text{lh } x \end{aligned}$$

which we know is primitive recursive.) So the set Tot is at least within two quantifiers of computability.

We can also formulate these definitions by recursion on n . R is Σ_{n+1} if we have

$$\vec{t} \in R \iff \exists x Q(\vec{t}, x)$$

for all \vec{t} , for some Π_n relation Q . Dually, R is Π_{n+1} if we have

$$\vec{t} \in R \iff \forall x Q(\vec{t}, x)$$

for all \vec{t} , where Q is Σ_n . Starting from the concepts of Σ_1 and Π_1 , we can use these clauses to characterize $\Sigma_2, \Pi_2, \Sigma_3, \dots$

Observation: (a) The complement of a Σ_n relation is Π_n .

(b) The complement of a Π_n relation is Σ_n .

Proof: We use induction on n . We have already seen the argument for $n = 1$.

Suppose, as the inductive hypothesis, both (a) and (b) hold when $n = k$, and consider a Σ_{k+1} relation R

$$\vec{t} \in R \iff \exists x Q(\vec{t}, x)$$

where Q is Π_k . Then we have

$$\begin{aligned} \vec{t} \in \overline{R} &\iff \text{not } \exists x Q(\vec{t}, x) \\ &\iff \forall x \text{ not } Q(\vec{t}, x) \text{ by De Morgan's laws} \\ &\iff \forall x \overline{Q}(\vec{t}, x) \end{aligned}$$

and by the inductive hypotheses, \overline{Q} is Σ_k , and hence \overline{R} is Π_{k+1} .

Similarly, the complement of a Π_{k+1} relation is Σ_{k+1} . So by induction, the observation holds for all $n \geq 1$. \dashv

For example, the set $\overline{\text{Tot}}$ of indices of non-total functions is Σ_2 :

$$e \in \overline{\text{Tot}} \iff \exists v \forall t \overline{T}(e, v, t)$$

Any Π_1 relation is also Σ_2 and Π_2 , because we can use “vacuous” quantifiers:

$$\begin{aligned} \forall y Q(\vec{x}, y) &\iff \exists z \forall y Q_1(\vec{x}, y, z) \\ &\iff \forall z \exists y Q_2(\vec{x}, y, z) \end{aligned}$$

where

$$Q_1(\vec{x}, y, z) \iff Q(\vec{x}, y) \quad \text{and} \quad Q_2(\vec{x}, y, z) \iff Q(\vec{x}, z).$$

Extending the use of vacuous quantifiers, we come to the following result.

- Proposition:** (a) Any Σ_n relation is also Δ_{n+1} .
 (b) Any Π_n relation is also Δ_{n+1} .

Thus, letting the noun Σ_k denote the collection of all Σ_k relations, we have the chains:

$$\begin{aligned}\Sigma_1 &\subseteq \Sigma_2 \subseteq \Sigma_3 \subseteq \cdots \\ \Pi_1 &\subseteq \Pi_2 \subseteq \Pi_3 \subseteq \cdots\end{aligned}$$

We say that these chains define the *arithmetical hierarchy*. But there will be many relations that fall outside this hierarchy (i.e., some relations are not Σ_n or Π_n for any n).

The following proposition supplies closure results under union, intersection, and substitution of total computable functions.

Proposition: Assume that Q and R are k -ary relations on the natural numbers.

(a) If Q and R are both Σ_n relations, then both $Q \cup R$ and $Q \cap R$ are also Σ_n relations.

(b) If Q and R are both Π_n relations, then both $Q \cup R$ and $Q \cap R$ are also Π_n relations.

Further assume that f_1, \dots, f_k are m -place total computable functions.

(c) If R is a Σ_n relation, then $\{\vec{x} \mid R(f_1(\vec{x}), \dots, f_k(\vec{x}))\}$ is also a Σ_n relation.

(d) If R is a Π_n relation, then $\{\vec{x} \mid R(f_1(\vec{x}), \dots, f_k(\vec{x}))\}$ is also a Π_n relation.

Like the earlier observation concerning complements, this proposition can be verified by using induction on n . But in place of De Morgan's laws, we employ the following quantifier manipulation rules:

$$\begin{aligned}\exists x M(x) \text{ and } \exists y N(y) &\iff \exists x \exists y [M(x) \text{ and } N(y)] \\ \exists x M(x) \text{ or } \exists y N(y) &\iff \exists z [M(z) \text{ or } N(z)] \\ \forall x M(x) \text{ or } \forall y N(y) &\iff \forall x \forall y [M(x) \text{ or } N(y)] \\ \forall x M(x) \text{ and } \forall y N(y) &\iff \forall z [M(z) \text{ and } N(z)]\end{aligned}$$

To see the correctness of the third rule, think about how the condition on the left side, $\forall x M(x) \text{ or } \forall y N(y)$, could *fail*. It fails if and only if there is both some counterexample x^* for which not $M(x^*)$ and some counterexample y^* for which not $N(y^*)$. Under what situations does the condition on the right side, $\forall x \forall y [M(x) \text{ or } N(y)]$, fail? It fails if and only if there is some counterexample $\langle x^*, y^* \rangle$ for which the condition in brackets fails, so that neither $M(x^*)$ nor $N(y^*)$. And that is exactly the same situation under which the left side failed.

Proof: Parts (c) and (d) follow from known substitution rules. To prove parts (a) and (b), we use induction on n . Suppose, as the inductive hypothesis, that part (b) holds when $n = k$, and consider two Σ_{k+1} relations Q and R :

$$\begin{aligned}\vec{t} \in R &\iff \exists x M(\vec{t}, x) \text{ where } M \text{ is } \Pi_k \\ \vec{t} \in Q &\iff \exists y N(\vec{t}, y) \text{ where } N \text{ is } \Pi_k\end{aligned}$$

Then we have

$$\begin{aligned}
\vec{t} \in R \cup Q &\iff \vec{t} \in R \text{ or } \vec{t} \in Q \\
&\iff \exists x M(\vec{t}, x) \text{ or } \exists y N(\vec{t}, y) \\
&\iff \exists z [M(\vec{t}, z) \text{ or } N(\vec{t}, z)] \\
&\iff \exists z [\langle \vec{t}, z \rangle \in M \cup N].
\end{aligned}$$

By the inductive hypothesis, $M \cup N$ is Π_k , and hence $R \cup Q$ is Σ_{k+1} . A similar argument shows that $R \cap Q$ is also Σ_{k+1} . Thus part (a) holds for $n = k + 1$.

Similarly, supposing that part (a) holds when $n = k$, we find that part (b) holds for $n = k + 1$.

What about the basis for the induction? The easiest approach is to take both “ Σ_0 ” and “ Π_0 ” to mean simply *computable*. We know that the class of computable relations is closed under union and intersection, so parts (a) and (b) hold when $n = 0$. \dashv

Part (c) is already familiar in the case $n = 1$; see page 410. Looking at complements will give us part (d) when $n = 1$. Then induction on n gives us parts (c) and (d) in general.

Corollary: Let A and B be sets of numbers with $A \leq_m B$.

- (a) If B is Σ_n , then A is also Σ_n .
- (b) If B is Π_n , then A is also Π_n .

Proof: Assume that $A \leq_m B$ under f . Then $A = \{x \mid f(x) \in B\}$. Apply parts (c) and (d) of the preceding proposition. \dashv

This corollary provides us with a method for showing that a set is *not* Σ_n or that it is *not* Π_n . We already know that to show that a set B is not r.e., one possible strategy is to try showing that $\bar{K} \leq_m B$. (Don’t we?) The corollary extends the method. Whenever we have a set S that is known not to be Σ_n , then we can show that another set B is not Σ_n if we can obtain $S \leq_m B$.

But to apply this method, we first need that initial set S that is known not to be Σ_n . Read on.

For the Σ_1 relations, we have from page 404 the “normal form” result: Whenever R is an n -ary Σ_1 relation, then R is the domain $W_e^{(n)}$ of some computable partial function $\llbracket e \rrbracket^{(n)}$. Hence for this e ,

$$R = W_e^{(n)} = \{\vec{x} \mid \exists t T^{(n)}(e, \vec{x}, t)\}.$$

We want to extend this idea. First, consider a Π_1 relation R . Its complement \bar{R} is Σ_1 , so we can say that for some e ,

$$\begin{aligned}
\vec{x} \in R &\iff \vec{x} \notin \bar{R} \\
&\iff \text{not } \exists t T^{(n)}(e, \vec{x}, t) \text{ by the above} \\
&\iff \forall t \text{ not } T^{(n)}(e, \vec{x}, t) \text{ by De Morgan} \\
&\iff \forall t \bar{T}^{(n)}(e, \vec{x}, t).
\end{aligned}$$

We conclude that any Π_1 relation R can be written, for some number e , in the form

$$R = \{\vec{x} \mid \forall t \bar{T}^{(n)}(e, \vec{x}, t)\}$$

(and of course conversely any relation of this form is Π_1).

Next, consider a Π_2 relation R . We know that

$$\vec{x} \in R \iff \forall y Q(\vec{x}, y)$$

for some Σ_1 relation Q . Using our preceding normal form for Σ_1 , we see that for some e ,

$$\vec{x} \in R \iff \forall y \exists t T^{(n+1)}(e, \vec{x}, y, t).$$

We thus obtain the normal form for a Π_2 relation R :

$$R = \{\vec{x} \mid \forall y \exists t T^{(n+1)}(e, \vec{x}, y, t)\}$$

for some number e .

Keep going. For a Σ_2 relation R , we have

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

for some Π_1 relation Q . Using our preceding normal form for Π_1 , we see that for some e ,

$$\vec{x} \in R \iff \exists y \forall t \bar{T}^{(n+1)}(e, \vec{x}, y, t).$$

For a Σ_3 relation R , we have

$$\vec{x} \in R \iff \exists z Q(\vec{x}, z)$$

for some Π_2 relation Q . Using our normal form for Π_2 , we see that for some e ,

$$\vec{x} \in R \iff \exists z \forall y \exists t T^{(n+2)}(e, \vec{x}, y, z, t).$$

One more. For a Π_3 relation R , we have

$$\vec{x} \in R \iff \forall z Q(\vec{x}, z)$$

for some Σ_2 relation Q . Using our normal form for Σ_2 , we see that for some e ,

$$\vec{x} \in R \iff \forall z \exists y \forall t \bar{T}^{(n+2)}(e, \vec{x}, y, z, t).$$

Let's collect what we have in a table:

Σ_1	$\{\vec{x} \mid \exists t T^{(n)}(e, \vec{x}, t)\}$
Π_1	$\{\vec{x} \mid \forall t \bar{T}^{(n)}(e, \vec{x}, t)\}$
Σ_2	$\{\vec{x} \mid \exists y \forall t \bar{T}^{(n+1)}(e, \vec{x}, y, t)\}$
Π_2	$\{\vec{x} \mid \forall y \exists t T^{(n+1)}(e, \vec{x}, y, t)\}$
Σ_3	$\{\vec{x} \mid \exists z \forall y \exists t T^{(n+2)}(e, \vec{x}, y, z, t)\}$
Π_3	$\{\vec{x} \mid \forall z \exists y \forall t \bar{T}^{(n+2)}(e, \vec{x}, y, z, t)\}$

And so forth and so on. (This table emphasizes that for any fixed k and n , there are only *countably* many Σ_n k -ary relations, and only countably many Π_n k -ary relations. The last line in the table shows that each Π_3 relation has the form $\{\vec{x} \mid \forall z \exists y \forall t \bar{T}^{(n+2)}(e, \vec{x}, y, z, t)\}$ for some e . Taking the various possible values of e

$$e = 0, \quad e = 1, \quad e = 2, \quad \dots$$

we get a complete list, with repetitions, of all the Π_3 k -ary relations. Moreover, because a countable union of countable sets is countable, we can go a step further and say that only countably many relations can be in the arithmetical hierarchy at all. Because $\mathcal{P}\mathbb{N}$ is uncountable, there is a sense in which “most” relations fall outside the arithmetical hierarchy.)

An advantage to having such “normal form” results is that we can diagonalize out of them. Recall that when we wanted a Σ_1 set that was not Π_1 , we used the set K defined by the condition

$$x \in K \iff \llbracket x \rrbracket(x) \downarrow \iff \exists t T(x, x, t).$$

Imitating this construction, define the Π_2 set S by the condition

$$x \in S \iff \forall y \exists t T^{(2)}(x, x, y, t).$$

Is it possible that this set S is also Σ_2 ? If so, then by our normal form results, it would have to be, for some number e , the set

$$V_e = \{x \mid \exists y \forall t \bar{T}^{(2)}(e, x, y, t)\}.$$

But S and V_e cannot be the same set, because they differ at the number e :

$$\begin{aligned} e \notin V_e &\iff \text{not } \exists y \forall t \bar{T}^{(2)}(e, e, y, t) \\ &\iff \forall y \exists t \text{not } \bar{T}^{(2)}(e, e, y, t) \\ &\iff \forall y \exists t T^{(2)}(e, e, y, t) \\ &\iff e \in S \end{aligned}$$

so e belongs to one and only one of the two sets S and V_e .

We conclude that S is Π_2 but not Σ_2 . So its complement \bar{S} is Σ_2 but not Π_2 . We can generalize the construction of S to obtain the following result.

Hierarchy theorem: For each positive integer n , there is some set that is Σ_n but not Π_n , and there is some set that is Π_n but not Σ_n .

Example: We know that the set Tot of indices of total functions is Π_2 . We can now show that it is *not* Σ_2 . Take S to be the above set that is Π_2 but not Σ_2 . By Exercise 3, we have $S \leq_m \text{Tot}$. Now apply the earlier corollary: Tot cannot be Σ_2 , lest S be Σ_2 .

Exercises

- (a) Show that $\{x \mid W_x \text{ is infinite}\}$ is Π_2 .
(b) Show that $\{x \mid \overline{W}_x \text{ is infinite}\}$ is Π_3 .
- Show that $\{x \mid W_x \text{ is a computable set}\}$ is Σ_3 .
- Show that every Π_2 set of natural numbers is many-one reducible to Tot. *Suggestion:* For a set $\{x \mid \forall u \exists v R(x, u, v)\}$, look at the function $\langle u, x \rangle \mapsto \mu v R(x, u, v)$ and apply the parameter theorem.
- Show that the binary relation $\{\langle x, y \rangle \mid W_x \subseteq W_y\}$ is a Π_2 relation.
- Let Z be the set of indices for the function that is constantly zero:

$$Z = \{t \mid \llbracket t \rrbracket(x) = 0 \text{ for all } x\}$$

- Show that Z is Π_2 .
- Show that Z is not Π_1 .
- Show that Z is not even Σ_2 .

Definability in arithmetic

A number is prime if it is greater than 1 and is not the product of two smaller numbers. That sentence reflects a certain property of the set of primes: The set of primes is *definable in arithmetic*. What other sets are definable in arithmetic? What sets are not?

Before tackling either of these questions, we need to be more explicit about what counts as “arithmetic.” We want to establish a certain language, so that we can then consider what is expressible in that language and what is not.

The study of definability in formal languages is an important part of logic. What we do here is to take an initial look at one such situation.

The language we want incorporates the following seven elements.

- A symbol 0 to name zero. We need to start somewhere.
- A symbol S for the successor function (that is, the function $S(x) = x + 1$). The string $S0$ names 1, the string $SS0$ names 2, and so forth. For each natural number n , we have a *numeral* $SS \cdots S0$ naming n ; call this numeral \bar{n} .
- Symbols for addition, multiplication, and exponentiation. (Everyday notation uses $+$ and \times for addition and multiplication, but lacks a symbol for exponentiation. The practice of writing x^y curiously avoids having a symbol for the exponentiation operation.)
- Symbols for comparing numbers: $=$, $<$, \leq .
And then some infrastructure.

- Variables x_1, x_2, x_3, \dots and u, v, w, \dots . There are enough variables that we will never run out. (This is only part of the story. Actually, it is important to make the total supply of symbols *finite*. So what the language really has is one or two variables, and a prime symbol $'$. That way we can make all the variables x', x'', x''', \dots we need, with just a few basic symbols.)
- Connective words ‘and,’ ‘or,’ ‘not,’ ‘if ... then,’ and ‘if and only if.’ Also parentheses, so we don’t get confused.
- Quantifiers over \mathbb{N} : $\forall v$ and $\exists v$ (for a variable v of our choice), to express “for every natural number” and “for some natural number.”

And that is all.

Example: In the language we can say

$$S0 < x_1 \text{ and not } \exists u \exists v (u < x_1 \text{ and } v < x_1 \text{ and } u \cdot v = x_1)$$

which expresses “ x_1 is prime.” If we are told what number the variable x_1 names, then we can try to say whether this expression—call it $\pi(x_1)$ —is true or false. Or better, if we *replace* the variable x_1 by a numeral $SS \cdots S0$, we get a sentence that is either true or false. That is, $\pi(SSS0)$ is true, but $\pi(SSSS0)$ is false. Or to use the abbreviations for numerals, $\pi(\bar{3})$ is true, but $\pi(\bar{4})$ is false.

Example: Fermat’s Last Theorem can be written as a sentence in the language.

Non-example: The language does not incorporate a way to say “for every set of natural numbers.” It has no way to refer to real numbers in general, or to points and lines. It talks only of natural numbers, their sums, their products, and so forth.

Definition: A set S of natural numbers is *definable in arithmetic* by an expression $\alpha(x_1)$ (of the language of arithmetic) if the following conditions hold for each number n :

- (i) If $n \in S$ then $\alpha(\bar{n})$ is a true sentence.
- (ii) If $n \notin S$ then $\alpha(\bar{n})$ is a false sentence.

Example: The set of primes is definable in arithmetic, by the expression $\pi(x_1)$ we have just seen.

Example: The set of odd numbers is defined in arithmetic by the expression $\exists y x_1 = y + y + S0$. (There are several claims being made here. First, this expression is indeed in the language of arithmetic; it employs only features from our given list. Secondly, for any odd number n , the result of replacing x_1 by \bar{n} is a true sentence. And thirdly, for any even number n , the result of replacing x_1 by \bar{n} is a false sentence.)

Non-example: There must be many sets that are *not* definable in arithmetic. There are uncountably many subsets of \mathbb{N} , by Cantor’s theorem. But only countably many can be definable in arithmetic. This is because there can be

only countably many defining expressions. Each expression is a finite string of symbols, drawn from a finite alphabet, and there are only countably many such strings. (See the appendix for a summary of facts about countable sets.)

The definability concept extends naturally to relations on \mathbb{N} .

Definition: A k -ary relation R on natural numbers is *definable in arithmetic* by an expression $\alpha(x_1, \dots, x_k)$ (of the language of arithmetic) if the following conditions hold for each k -tuple of numbers $\langle n_1, \dots, n_k \rangle$:

- (i) If $\langle n_1, \dots, n_k \rangle \in R$ then $\alpha(\bar{n}_1, \dots, \bar{n}_k)$ is a true sentence.
- (ii) If $\langle n_1, \dots, n_k \rangle \notin R$ then $\alpha(\bar{n}_1, \dots, \bar{n}_k)$ is a false sentence.

Example: The divisibility relation (which is a binary relation) is defined in arithmetic by the expression $\exists y x_1 \cdot y = x_2$. Call this expression $\delta(x_1, x_2)$. Then $\delta(7, 91)$, which is the sentence $\exists y 7 \cdot y = 91$, is true, because we can take $y = 13$.

Example: Let A be the binary relation of being “adjacent primes.” That is, $\langle p, q \rangle \in A \Leftrightarrow$ both p and q are prime and $p < q$ and there is no prime in between. (For example, $\langle 3, 7 \rangle \notin A$ and $\langle 13, 17 \rangle \in A$.) Then A is defined in arithmetic by the expression:

$$\pi(x_1) \text{ and } \pi(x_2) \text{ and } x_1 < x_2 \text{ and not } \exists z(\pi(z) \text{ and } x_1 < z < x_2)$$

where $\pi(x_1)$ is the earlier expression defining the primes.

Our goal is to show that every relation that is Σ_n or Π_n (for any n) is definable in arithmetic. A more immediate goal is to show that the graph of every primitive recursive function is definable in arithmetic.

Example: The graph of the one-place function $f(t) = \lfloor t/2 \rfloor$ is a binary relation, and is defined in arithmetic by the expression

$$x_2 + x_2 = x_1 \text{ or } x_2 + x_2 + S0 = x_1.$$

The initial functions present no difficulties:

1. The k -place function that is constantly 0 has a graph that is defined in arithmetic by the expression $x_{k+1} = 0$.
2. The successor function (which is 1-place) has a graph that is defined in arithmetic by the expression $x_2 = Sx_1$.
3. The projection function I_m^k (where $1 \leq m \leq k$) has a graph that is defined in arithmetic by the expression $x_{k+1} = x_m$

Now for a more serious matter.

Theorem: The class of functions with graphs definable in arithmetic is closed under composition. That is, if f and g_1, \dots, g_k all have graphs definable in arithmetic and if h is given by the equation $h(\vec{t}) = f(g_1(\vec{t}), \dots, g_k(\vec{t}))$, then the graph of h is also definable in arithmetic.

Proof (for 2-place functions): Assume the following:

The graph of f is defined in arithmetic by the expression $\varphi(x_1, x_2, x_3)$.

The graph of g_1 is defined in arithmetic by the expression $\gamma_1(x_1, x_2, x_3)$.

The graph of g_2 is defined in arithmetic by the expression $\gamma_2(x_1, x_2, x_3)$.

And let $h(p, q) = f(g_1(p, q), g_2(p, q))$. We claim that h is defined in arithmetic by the following expression:

$$\exists y_1 \exists y_2 [\gamma_1(x_1, x_2, y_1) \text{ and } \gamma_2(x_1, x_2, y_2) \text{ and } \varphi(y_1, y_2, x_3)]$$

Call this expression $\sigma(x_1, x_2, x_3)$. If $h(a, b) = c$, then $\sigma(\bar{a}, \bar{b}, \bar{c})$ is a true sentence, because we can assign $g_1(a, b)$ to y_1 and $g_2(a, b)$ to y_2 .

Conversely, suppose that $\sigma(\bar{a}, \bar{b}, \bar{c})$ is a true sentence. So there must be numbers assigned to y_1 and y_2 making the expression true. The number assigned to y_1 must have been $g_1(a, b)$ to make $\gamma_1(\bar{a}, \bar{b}, y_1)$ true. Similarly, the number assigned to y_2 must have been $g_2(a, b)$ to make $\gamma_2(\bar{a}, \bar{b}, y_2)$ true. Consequently c must be $f(g_1(a, b), g_2(a, b))$ to make $\varphi(y_1, y_2, \bar{c})$ true. \dashv

It remains to show closure under primitive recursion. Towards that end, we will employ the following two lemmas regarding two specific primitive recursive functions.

Lemma: The graph of the function $t \mapsto p_t$ (where p_t is the $(t + 1)$ st prime number) is definable in arithmetic.

Proof: We already have an expression $\delta(x_1, x_2)$ defining in arithmetic the divisibility relation, and an expression $\alpha(x_1, x_2)$ defining in arithmetic the relation of being adjacent primes.

First, consider the relation Q for which

$$\langle b, c \rangle \in Q \iff b \text{ is prime and } c = 2^0 3^1 5^2 \dots b^\square$$

where \square is the number for which $b = p_\square$. For example, $\langle 5, 75 \rangle \in Q$, because $75 = 2^0 3^1 5^2$. Here are the first four members of Q :

$$Q = \{ \langle 2, 1 \rangle, \langle 3, 3 \rangle, \langle 5, 75 \rangle, \langle 7, 25725 \rangle, \dots \}$$

In general, we can say that $\langle b, c \rangle \in Q$ if and only if b is prime and

- (i) $2 \nmid c$,
- (ii) for any adjacent primes q and r with $q < r \leq b$, we have

$$q^j \mid c \iff r^{j+1} \mid c$$

for all j , and

- (iii) no prime larger than b divides c .

Translating these conditions into the language of arithmetic, we obtain an expression defining Q :

$$\begin{aligned} & \pi(x_1) \text{ and not } \delta(SS0, x_2) \text{ and} \\ & \forall u \forall v [\text{if } (\alpha(u, v) \text{ and } v \leq x_1) \text{ then } \forall w (\delta(u^w, x_2) \text{ if and only if } \delta(v^{S^w}, x_2))] \\ & \text{and not } \exists z (\pi(z) \text{ and } x_1 < z \text{ and } \delta(z, x_2)) \end{aligned}$$

Call this expression $\theta(x_1, x_2)$.

Secondly, observe that $p_a = b$ if and only if b is prime and, where c is the unique number for which $\langle b, c \rangle \in Q$, we have $b^a \mid c$ and $b^{a+1} \nmid c$. (For example, $p_2 = 5$ because 5 is prime, $5^2 \mid 75$, and $5^3 \nmid 75$.) Thus the expression

$$\pi(x_2) \text{ and } \exists y[\theta(x_2, y) \text{ and } \delta(x_2^{x_1}, y) \text{ and not } \delta(x_2^{Sx_1}, y)]$$

defines the graph of the function $t \mapsto p_t$ in arithmetic. \dashv

Lemma: The graph of the decoding function $\langle s, t \rangle \mapsto (s)_t$ is definable in arithmetic.

Proof: The key fact is that

$$(s)_t = \begin{cases} 0 & \text{if } s = 0 \\ 0 & \text{if } p_t \nmid s \\ \text{the } e \text{ for which } p_t^{e+1} \mid s \text{ and } p_t^{e+2} \nmid s & \text{otherwise.} \end{cases}$$

Using the expression $\delta(x_1, x_2)$ for divisibility and the expression $\psi(x_1, x_2)$ for the graph of $t \mapsto p_t$, we can make the expression

$$\begin{aligned} & (x_1 = 0 \text{ and } x_3 = 0) \text{ or} \\ & \exists y(\psi(x_2, y) \text{ and not } \delta(y, x_1) \text{ and } x_3 = 0) \text{ or} \\ & \exists y(\psi(x_2, y) \text{ and } \delta(y^{Sx_3}, x_1) \text{ and not } \delta(y^{SSx_3}, x_1)) \end{aligned}$$

which defines the graph of $\langle s, t \rangle \mapsto (s)_t$ in arithmetic. (Every number divides 0, so there is no danger that the three clauses might overlap.) \dashv

Theorem: The class of functions with graphs definable in arithmetic is closed under primitive recursion. That is, if f and g have graphs definable in arithmetic, and if h is given by the recursion equations

$$h(\vec{r}, 0) = f(\vec{r}) \quad \text{and} \quad h(\vec{r}, t + 1) = g(h(\vec{r}, t), \vec{r}, t)$$

then the graph of h is also definable in arithmetic.

Proof: The key fact is that $h(\vec{r}, t) = q$ if and only if there exists some number s with the following three properties:

- (i) $(s)_0 = f(\vec{r})$.
- (ii) $(s)_{j+1} = g((s)_j, \vec{r}, j)$ for each j , unless $j \geq t$.
- (iii) $(s)_t = q$.

(In one direction, if we have the equation $h(\vec{r}, t) = q$, then taking

$$s = [h(\vec{r}, 0), h(\vec{r}, 1), \dots, h(\vec{r}, t)],$$

we see that (i), (ii), and (iii) all hold. In the other direction, suppose that s is a number satisfying (i), (ii), and (iii). Then by induction of j , we see that $(s)_j = h(\vec{r}, j)$ for $j \leq t$. In particular, $(s)_t = q = h(\vec{r}, t)$.)

For notational simplicity, suppose that \vec{r} is a single number r . We are given an expression $\varphi(x_1, x_2)$ defining the graph of f and an expression $\gamma(x_1, x_2, x_3, x_4)$

defining the graph of g . From the foregoing lemma, we have an expression $\beta(x_1, x_2, x_3)$ defining the graph of the decoding function $\langle s, t \rangle \mapsto (s)_t$. Then the expression

$$\begin{aligned} & \exists z[\exists y(\varphi(x_1, y) \text{ and } \beta(z, 0, y)) \text{ and} \\ & \forall u[x_2 \leq u \text{ or } \exists v \exists w(\beta(z, u, v) \text{ and } \beta(z, Su, w) \text{ and } \gamma(v, x_1, u, w))] \\ & \text{and } \beta(z, x_2, x_3)] \end{aligned}$$

defines the graph of h in arithmetic. (Translation hints: The variable z will be assigned a number s meeting (i)–(iii). The variable y will be assigned $f(r)$.)
 \dashv

Corollary: The graph of any primitive recursive function is definable in arithmetic.

Corollary: Every primitive recursive relation is definable in arithmetic.

Proof: For any k -ary primitive recursive relation R , the graph of its characteristic function is defined in arithmetic by some expression $\rho(x_1, \dots, x_k, x_{k+1})$. Then R is defined in arithmetic by the expression $\rho(x_1, \dots, x_k, S0)$.
 \dashv

Corollary: Every Σ_1 relation is definable in arithmetic.

Proof: We showed back on page 403 that any Σ_1 relation had the form $\{\vec{s} \mid \exists t Q(\vec{s}, t)\}$ for a primitive recursive relation Q . We know that Q is definable; add one more quantifier to define $\{\vec{s} \mid \exists t Q(\vec{s}, t)\}$ in arithmetic.
 \dashv

Digression: Work by Martin Davis, Yuri Matiyacevich, Hilary Putnam, and Julia Robinson has shown that any Σ_1 relation is definable in arithmetic by an expression

$$\exists y_1 \cdots \exists y_k \theta$$

where θ contains no quantifiers at all. And it gets even better than that; θ can actually be a polynomial equation. In particular, while θ uses multiplication and addition, it does not need exponentiation. We have included exponentiation in our language in order to simplify the proofs.

But all of these corollaries are mere preliminaries for the following result.

Theorem: Any relation that is Σ_n or Π_n (for any n) is definable in arithmetic.

Proof: Keep adding quantifiers. \dashv

And that is where this string of results stops. Although we will not go into the topic here, the converse to the theorem also holds: The Σ_n and Π_n relations are the *only* relations that are definable in arithmetic.

The complexity of truth

We now know that for any Σ_{99} set S , there is an expression $\alpha(x_1)$ of arithmetic such that

$$n \in S \iff \alpha(\bar{n}) \text{ is a true sentence.}$$

That is, any Σ_{99} set is reducible, in a sense, to the set of true sentences in arithmetic. And the same holds for any set that is Σ_{999} or elsewhere in the arithmetical hierarchy. This “reducibility” will be seen to demonstrate that the set of true sentence is a very complicated set. In particular, it will be seen that the set of true sentences is *undecidable*. It is not even semi-decidable.

In order to describe the situation more precisely, we need to convert the set of true sentences of arithmetic to a set of numbers. That is, to each expression ε of arithmetic (which is a string symbols), we can assign its *Gödel number* $\#\varepsilon$, much as we assigned Gödel numbers to register machine programs. Of course, different expressions receive different Gödel numbers.

Moreover, for any fixed expression $\alpha(x_1)$, we expect the function

$$n \mapsto \#\alpha(\bar{n})$$

to be a computable function. (This function needs to go through the expression, and replace occurrences of the variable x_1 by the numeral \bar{n} .)

Rather than to go into the specifics of Gödel numbering, let’s take it for granted that the Gödel numbers can be assigned in such a way that $n \mapsto \#\alpha(\bar{n})$ is always a computable function. (This is not so unreasonable. Expressions of arithmetic are words over a certain finite language. We can code such words by numbers. Logic textbooks, such as the one cited in the *References*, carry out Gödel numbering and verify that substituting a numeral for a variable is a computable procedure.)

Define True to be the set of Gödel numbers of true sentences of arithmetic:

$$\text{True} = \{\#\tau \mid \tau \text{ is a true sentence of arithmetic}\}$$

For example, the number

$$\#\forall x \forall y \forall z \forall n (n \leq 2 \text{ or not } x^n + y^n = z^n)$$

belongs to the set True. An indication of the complexity of the set True is given by the following result.

Proposition: For any set S that is definable in arithmetic, we have $S \leq_m$ True.

Proof: Say S is defined in arithmetic by the expression $\alpha(x_1)$. Thus

$$n \in S \iff \#\alpha(\bar{n}) \in \text{True}$$

for each number n . Since $n \mapsto \#\alpha(\bar{n})$ is a computable function, we conclude that $S \leq_m$ True under this function. \dashv

Corollary: For any set S that is either Σ_n or Π_n for some n , we have $S \leq_m \text{True}$.

Proof: Any such set is definable in arithmetic; apply the preceding proposition. \dashv

Tarski's theorem: The set True is not Σ_n or Π_n for any n .

Proof: We will show that True is not Σ_{99} . Let S be a set that is Π_{99} but not Σ_{99} . (We know there are such sets.) By the above corollary, we have $S \leq_m \text{True}$. Therefore True cannot be Σ_{99} , lest S be Σ_{99} . \dashv

In particular, True is not a computable set. That is, *truth in arithmetic is undecidable*. We have here an unsolvability result for a problem that is not formulated in terms of computability concepts (as was the halting problem). Even for arithmetic, sometimes regarded as one the simpler branches of mathematics, the set of true sentences is not a decidable set. (Of course, if instead of calling it “arithmetic,” we call it “number theory,” then its undecidability comes as less of a surprise.)

Digression: Suppose that we omit exponentiation from our language, and we apply it not to \mathbb{N} but to \mathbb{R} . That is, now $\forall x$ means “for all real numbers x ,” and $\exists x$ means “there exists a real number x .” Then we get a language for talking about the real number line. For example, a subset of \mathbb{R} that is a finite union of intervals (open or closed) with rational endpoints is definable in this language. Perhaps surprisingly, the set True in this situation turns out to be computable! This follows from a different theorem of Tarski, from 1939.

Moreover, in addition to showing that True is not computable, we have shown that True is not Σ_1 ; that is, it is not recursively enumerable. (More informally, truth in arithmetic is not semi-decidable.) This fact, although lacking the full strength of Tarski's theorem, is relevant to *axiomatic theories* in arithmetic, such as might be studied in a logic course.

Imagine, then, that we want to develop an axiomatic theory for arithmetic. So we need two components. First, we need to adopt some true sentences as our *axioms*. (There is a set of sentences called the *first-order Peano axioms* that is a popular choice.) Secondly, we need to adopt rules for what is an acceptable *proof*. Here there is not so much latitude; logicians have succeeded in nailing down the concept of a proof from axioms very precisely.

But there is one additional feature we expect: The binary relation

$$\{\langle \pi, \sigma \rangle \mid \pi \text{ is a proof of } \sigma\}$$

must be a *decidable* relation. That is, it is not acceptable to take simply all true sentences as axioms, because then we could not effectively tell an axiom from a non-axiom. (The set True is not computable.) A key feature of a *proof* is that it should be *effectively verifiable*. It must be possible—in principle—for a hard-working graduate student (or a referee) to check a proof line by line and

verify its correctness. We cannot demand that the student contribute brilliant insights. Nor can we demand that the student spend an infinite amount of time, checking an infinite number of cases. What we can insist on is that the student or the referee must eventually either conclude that the proof is correct, or conclude that it is not yet acceptable. We need to be able to distinguish between proofs and non-proofs.

Using Church's Thesis and Gödel numbers, this demand can be translated as follows: The binary relation

$$\{\langle p, s \rangle \mid p \text{ is the Gödel number of a proof of the sentence with Gödel number } s\}$$

must be a computable relation. This has the following consequence:

$$\{s \mid \exists p(p \text{ is the Gödel number of a proof of the sentence with Gödel number } s)\}$$

is recursively enumerable. (In fact, this set would be r.e. even if the binary relation, instead of being computable, were merely r.e.) That is, in an axiomatic theory of arithmetic, the set of Gödel numbers of provable sentences is recursively enumerable. Therefore this set cannot be the same as True, which is not recursively enumerable. The best that an axiomatic theory can hope to generate is some recursively enumerable subset T of True.

Gödel incompleteness theorem (1931): For any recursively enumerable subset T of True, we can find a true sentence σ with $\#\sigma \notin T$.

Thus, for an axiomatic theory of arithmetic, we can find a true sentence not provable in that theory.

First proof: T is a subset of True. These two sets cannot be equal, because the first is r.e. and the second is not. So there must be something in True not in T . \neg

But it will be more interesting if we can actually get some idea of what that sentence σ might express. Let's retrace the argument.

Second proof: We know that True is not r.e., because the non- Σ_1 set \overline{K} is many-one reducible to True. That is, there is an expression $\kappa(x_1)$ that defines \overline{K} in arithmetic, and

$$n \in \overline{K} \iff \#\kappa(\bar{n}) \in \text{True}.$$

For the given r.e. subset T of True, let

$$J = \{n \mid \#\kappa(\bar{n}) \in T\}.$$

Thus J is the set of numbers that T "knows" are in \overline{K} . The set J is r.e. (because we have $J \leq_m T$ under the function $n \mapsto \#\kappa(\bar{n})$). So we have $J = W_j$ for some number j . Moreover, $J \subseteq \overline{K}$ (because

$$n \in J \Rightarrow \#\kappa(\bar{n}) \in T \Rightarrow \kappa(\bar{n}) \text{ is true}$$

for each n). Therefore J is a proper subset of \overline{K} , because J is r.e. and \overline{K} is not. So there is a number in \overline{K} that is not in J . In fact, j is such a number, by the proposition on page 405. Thus the sentence

$$\kappa(\bar{j})$$

is true (because $j \in \overline{K}$) but its Gödel number is not in T (because $j \notin J$). So here is a specific sentence witnessing Gödel's incompleteness theorem. \dashv

And what might this sentence $\kappa(\bar{j})$ say? Literally, it speaks of numbers and their sums and products—dullsville. But we can give it a more interesting translation:

$$\begin{array}{ll} \kappa(\bar{j}) & \text{says } j \in \overline{K} \\ & \text{i.e., } j \notin W_j \\ & \text{i.e., } j \notin J \\ & \text{i.e., } \# \kappa(\bar{j}) \notin T \end{array}$$

That is, our witness (our true unprovable sentence) asserts, in a sense, that it is itself not in the axiomatic theory that yields T . It is saying (under this rather free translation), “I am unprovable in this axiomatic theory.”

Digression: In 1931, Gödel did not have the development of computability theory available to him. Instead, he proceeded directly to an ingenious construction of a sentence that could be freely translated as saying “I am unprovable in this axiomatic theory.” This sentence had to be true (if it were false, we would have a provable falsehood), and hence unprovable in the axiomatic theory. Even better, Gödel worked not from the concept of a true sentence, but from the concept (from logic) of a consistent theory. This led to a result called the *second* incompleteness theorem, which cannot be explored here.

Emil Post, in a seminal 1944 paper, defined a concept he called a *creative* set; the set K was an example of such a set. (See Exercise 8.) He gave, much as is done here, a version of the Gödel incompleteness theorem. He then added: “The conclusion is unescapable that even for such a fixed, well defined body of mathematical propositions, mathematical thinking is, and must remain, essentially creative.”

Exercises

6. Call a set S of natural numbers *productive* if there is a computable partial function f (a *productive function* for S) such that whenever $W_x \subseteq S$ then $f(x)$ is defined and belongs to S but not to W_x . (Thus $f(x)$ witnesses the fact that W_x is not all of S .) For example, \overline{K} is productive, and the identity function is a productive function for \overline{K} . Clearly, a productive set cannot be recursively enumerable. Show that if $A \leq_m B$ and A is productive, then B is also productive.

7. (a) Show that the set True is productive.
 (b) Show that its complement, $\overline{\text{True}}$, is also productive.

(c) Show that the set Tot is productive.

(d) Show that its complement, $\overline{\text{Tot}}$, is also productive.

8. Call a set *creative* if it is recursively enumerable and its complement is productive. For example, the set K is creative. Show that any m-complete r.e. set (i.e., any r.e. set such that all other r.e. sets are many-one reducible to it) is creative.