Gödel's Incompleteness Theorems

Stefan Hetzl
stefan.hetzl@tuwien.ac.at

Vienna University of Technology

Summer Term 2022

Contents

Introduction

Gödel's incompleteness theorems are among the most important results in mathematical logic. In order to fully appreciate their significance, it is necessary to explain the historical background. At the turn from the 19th to the 20th century, several paradoxes surfaced in the foundations of mathematics, leading to increasing uncertainty concerning the solidity of these foundations. There have been a number of reactions to that situation, the most far-reaching of them was Hilbert's.

At the beginning of the 1920ies, Hilbert put forward a proposal for the foundations of mathematics which is now called "Hilbert's programme". This programme is based on a simple but striking observation which underlies all formalisation efforts, in particular also Russel and Whitehead's Principia Mathematica: in mathematics we talk about infinite sets, real numbers, real-valued functions, operators transforming such functions, etc. in short: about abstract, infinite objects. However, we do so in an inherently finite way; every proof is a finite sequence of symbols, taken from some finite set, every theory is a finite succession of such proofs. What we say and prove about such objects is thus inherently finite. For Hilbert, the part of mathematics which deals with elementary properties of finite sequences of symbols was relying only on a purely intuitive basis. Their elementary properties and relations are immediate and not mediated by logic. Therefore they are not susceptible to the possibility of a contradiction. Elementary statements about such sequences thus form a secure basis for the foundations of mathematics. Hilbert proposed to use this basis for giving an axiomatic formalisation of all of mathematics and to prove this formalisation consistent, i.e., to show that no contradiction can arise based on consideration of finite sequences of symbols alone. Thus, so Hilbert thought, one could justify the use of abstract concepts in mathematics.

However, this hope was shattered by Gödel's incompleteness theorems, which were published in 1931. Informally, they can be stated as follows:

Theorem (First Incompleteness Theorem)**.** *Let T be a consistent and axiomatisable theory "containing arithmetic", then there is a sentence σ s.t. $T \nvdash \sigma$ and $T \nvdash \neg\sigma$.*

Theorem (Second Incompleteness Theorem)**.** *Let T be a consistent and axiomatisable theory "containing arithmetic", then $T \nvdash \mathsf{Con}_T$.*

Without explaining these statements in detail, let us just note that the conditions imposed on T in these two theorems are not identical but, in both cases, encompass all situations envisaged by Hilbert in his programme to prove consistency statements. The second incompleteness theorem destroys Hilbert's programme, for if a theory cannot prove its own consistency, then an even weaker theory, for example one that speaks only about finite sequences of symbols, cannot prove it either. Thus, after publication of the incompleteness theorems, Hilbert's programme had to be given up.

Nevertheless, the investigation of the logical foundations of mathematics that has been carried out since, while not leading to consistency proofs as envisaged by Hilbert, has led to an improvement of our understanding which was sufficient for dissipating doubts about the consistency of mathematical reasoning. Gödel's incompleteness theorems have become a cornerstone of logic (in mathematics, philosophy, and computer science). The proof techniques introduced by Gödel in these results, arithmetisation (also called "Gödelisation") in conjunction with diagonalisation, have become central for many results in mathematical logic.

This course is designed as a second course in mathematical logic, centred around the incompleteness theorems. We are assuming passive and active knowledge of first-order logic, in particular, the syntax and semantics of formulas, proof calculi, models, and the completeness theorem. We will take the incompleteness theorems as central aims of this course. However, we will not proceed there in the most direct way possible. Instead, we take them as occasion to study important notions and results surrounding them, in particular, in computability theory and formal theories of arithmetic.

The proof techniques of arithmetisation and diagonalisation form the backbone of this course. Along this backbone we will proceed with topics of increasing complexity culminating in strong versions of the incompleteness theorems. We will start with studying basic computability theory in Chapter 1. On the one hand, this allows to present these proof techniques in a comparatively simple context. On the other hand, the results of this chapter lay imporant groundwork for later chapters. In the first chapter we will not yet speak about logic in the narrow sense. In particular, formulas will only enter the picture once we start with Chapter 2. Chapter 2 is centred around the question which sets can be defined by (which classes of) arithmetical formulas. Naturally, this leads us to working with formulas, but not yet with proofs. Consequently the only model of interest here will be the standard model $\mathbb{N}$ of the natural numbers. In the main chapter of this course, Chapter 3 on arithmetical theories, we will also work with proofs and non-standard models of arithmetic. We will study several arithmetical theories in order of increasing strength and harvest various versions of the incompleteness theorems as we go. The final Chapter 4 collects some additional results and remarks that provide complementary perspectives on the topics treated in this course.

As further literature, [4] can be recommended as a compact presentation of the incompleteness theorems and [1] as a comprehensive reference on theories of arithmetic. Furthermore, [2] provides a more model-theoretic perspective on theories of arithmetic and [6, 3] are useful for background in computability theory. [5] is a good introduction to provability logic. These lecture notes owe a debt to all of these sources.

Bibliography

[1] Petr Hájek and Pavel Pudlák. *Metamathematics of First-Order Arithmetic*. Springer, 1993.

[2] Richard Kaye. *Models of Peano Arithmetic*, volume 15 of *Oxford Logic Guides*. Clarendon Press, 1991.

[3] Piergiorgio Odifreddi. *Classical Recursion Theory*, volume 125 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., 1989.

[4] Craig Smorynski. The Incompleteness Theorems. In J. Barwise, editor, *Handbook of Mathematical Logic*, pages 821–865. North-Holland, 1977.

[5] Craig Smoryński. Modal logic and self-reference. In D. M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, pages 1–53. Springer, 2004.

[6] Robert I. Soare. *Recursively Enumerable Sets and Degrees*. Perspectives in Mathematical Logic. Springer, 1987.

Chapter 1

Computability

Computability theory is, along with proof theory, set theory, and model theory, one of the four main areas of mathematical logic. The incompleteness theorems are strongly connected, both historically and mathematically, to central notions and techniques of computability theory. We will therefore start this lecture on the former with a brief introduction to the latter. The aim of this chapter is to prove the existence of a recursively enumerable but undecidable set. From this result we will soon be able to obtain a weak version of the first incompleteness theorem as a corollary. As we go along, we pick up some notions, in particular concerning coding, also called arithmetisation or "Gödelisation", that will be useful later on.

1.1 The partial recursive functions

One approach to defining the set of functions which are computable in the intuitive sense is to start "from below": define some functions which are obviously computable, then define closure operators which transform computable functions in computable functions. We will follow this approach here.

Definition 1.1. The *basic functions* are:

1. the constant (nullary function) $0 \in \mathbb{N}$,
2. the *successor function* $\mathrm{S} : \mathbb{N} \to \mathbb{N}, x \mapsto x + 1$,
3. for all $k \geq 1, 1 \leq i \leq k$, the *projection function* $\mathrm{P}^k_i : \mathbb{N}^k \to \mathbb{N} : (x_1, \ldots, x_k) \mapsto x_i$.

All of the basic functions are obviously computable.

Definition 1.2. Let $f : \mathbb{N}^n \to \mathbb{N}$, $g_1 : \mathbb{N}^k \to \mathbb{N}, \ldots, g_n : \mathbb{N}^k \to \mathbb{N}$. Then the function obtained by *composition* of f with $g_1, \ldots, g_n$ is

$$\mathrm{Cn}[f, g_1, \ldots, g_n] : \mathbb{N}^k \to \mathbb{N}, \overline{x} \mapsto f(g_1(\overline{x}), \ldots, g_n(\overline{x})).$$

If $n = 1$, then $\mathrm{Cn}[f, g]$ is usually written as $f \circ g$. If $f, g_1, \ldots, g_n$ are computable, then so is h: in order to compute h, we first compute $y_i = g_i(\overline{x})$ for $i = 1, \ldots, n$ which is possible by assumption and then we compute $f(y_1, \ldots, y_n)$ which is, again, possible by assumption. Another way to put the above definition is to say that, for $k, n \in \mathbb{N}$, Cn^k_n is an operator, transforming functions into functions, i.e., C^k_n is of type $(\mathbb{N}^n \to \mathbb{N}) \times (\mathbb{N}^k \to \mathbb{N})^n \to (\mathbb{N}^k \to \mathbb{N})$.

Definition 1.3. Let $f : \mathbb{N}^k \to \mathbb{N}$ and $g : \mathbb{N}^{k+2} \to \mathbb{N}$. Then the function obtained by *primitive recursion* of f and g is $\mathrm{Pr}[f, g] = h : \mathbb{N}^{k+1} \to \mathbb{N}$ defined by

$$h(\overline{x}, 0) = f(\overline{x}), \text{ and}$$
$$h(\overline{x}, y + 1) = g(\overline{x}, y, h(\overline{x}, y)).$$

If f and g are computable then so is h. Let $\overline{x} \in \mathbb{N}^k$. We argue, informally, by induction on $y \in \mathbb{N}$: if $y = 0$ then, by assumption, $f(\overline{x})$ can be computed and thus $h(\overline{x}, y)$ can be computed. If $y > 0$, say $y = y' + 1$, we can compute $z = h(\overline{x}, y')$ by induction hypothesis and then $h(\overline{x}, y) = g(\overline{x}, y', z)$ from it by assumption.

Definition 1.4. A function $f : \mathbb{N}^k \to \mathbb{N}$ is called *primitive recursive* if it can be obtained from the basic functions by a finite number of applications of the operators composition and primitive recursion.

Example 1.5. Consider the functions $f = \mathrm{P}^1_1 : \mathbb{N} \to \mathbb{N}$ and $g : \mathbb{N}^3 \to \mathbb{N}, (x, y, z) \mapsto z + 1$. Then $g = \mathrm{S} \circ \mathrm{P}^3_3$. By primitive recursion of f and g we obtain the function $h : \mathbb{N}^2 \to \mathbb{N}$ defined by

$$h(x, 0) = \mathrm{P}^1_1(x) = x, \text{ and}$$
$$h(x, y + 1) = g(x, y, h(x, y)) = h(x, y) + 1.$$

In other words, h is the addition of natural numbers which is hence primitive recursive. This fact can also be written as $+ = \mathrm{Pr}[\mathrm{P}^1_1, \mathrm{Cn}[\mathrm{S}, \mathrm{P}^3_3]]$.

Lemma 1.6. *The following functions are primitive recursive*

1. *addition* $(x, y) \mapsto x + y$,
2. *the constant function* $\mathrm{c}^k_z : \mathbb{N}^k \to \mathbb{N}, (x_1, \ldots, x_k) \mapsto z$,
3. *multiplication* $(x, y) \mapsto x \cdot y$
4. *truncated predecessor* $x \mapsto \mathrm{p}(x) = \begin{cases} 0 & \text{if } x = 0 \\ x - 1 & \text{if } x > 0 \end{cases}$
5. *truncated subtraction* $(x, y) \mapsto x \dot{-} y = \begin{cases} 0 & \text{if } x \leq y \\ x - y & \text{if } x > y \end{cases}$
6. *the characteristic function of less than or equal* $(x, y) \mapsto \chi_{\leq}(x, y) = \begin{cases} 1 & \text{if } x \leq y \\ 0 & \text{if } x > y \end{cases}$
7. *the characteristic function of equality* $(x, y) \mapsto \chi_{=}(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$

Proof. 1. has been shown in Example 1.5. For showing 2., first note that $\mathrm{c}^0_z = \mathrm{Cn}[\mathrm{S}, \mathrm{Cn}[\mathrm{S} \cdots \mathrm{Cn}[\mathrm{S}, 0] \cdots]]$. For $k = 1$ we use a trick based on the Pr-operator and define $\mathrm{c}^1_z = \mathrm{Pr}[\mathrm{c}^0_z, \mathrm{P}^2_2]$. Then $\mathrm{c}^1_z(0) = \mathrm{c}^0_z = z$ and $\mathrm{c}^1_z(y + 1) = \mathrm{P}^2_2(y, \mathrm{c}^1_z(y)) = \mathrm{c}^1_z(y) = z$. For $k \geq 2$ we can simply define $\mathrm{c}^k_z = \mathrm{Cn}[\mathrm{c}^1_z, \mathrm{P}^k_1]$. For 3. consider that $x \cdot 0 = 0$ and $x \cdot (y + 1) = x \cdot y + x$, i.e., $\cdot = \mathrm{Pr}[f, g]$ where $f(x) = 0$ and $g(x, y, z) = z + x$, i.e., $f = \mathrm{c}^1_0$ and $g = \mathrm{Cn}[+, \mathrm{P}^3_3, \mathrm{P}^3_1]$. For 4. we can simply define $\mathrm{p} = \mathrm{Pr}[0, \mathrm{P}^2_1]$. For 5. we use a primitive recursive definition based on $x \dot{-} 0 = x$ and $x \dot{-} (y + 1) = \mathrm{p}(x \dot{-} y)$. For 6. observe that $\chi_{\leq}(x, y) = 1 \dot{-} (x \dot{-} y)$. For 7. note that $\chi_{=}(x, y) = \chi_{\leq}(x, y) \cdot \chi_{\leq}(y, x)$. □

At this point one may start to wonder: are the primitive recursive functions all computable functions? did we miss some? The following informal argument shows that there are computable functions which are not primitive recursive. Every primitive recursive function can be defined by a finite string of symbols that conforms to certain simple rules on the arity of the involved functions. Thus all such definitions can be effectively listed. Let f_n be the n-th function in that list and define $g(n) = f_n(n) + 1$. Then g cannot be in this list, for suppose it were, i.e., $g = f_e$, then $g(e) = f_e(e) = f_e(e) + 1$, contradiction. So g is not primitive recursive. However, g is computable in the intuitive sense. This kind of argument, diagonalisation, will reappear at several central places in this course. This argument applies to every set of total functions which can be effectively enumerated. However, diagonalisation is not an obstacle for partial functions, since $f_e(e)$ may simply be undefined. This motivates the following considerations.

Definition 1.7. A *partial function* from $\mathbb{N}^k$ to $\mathbb{N}$, in symbols $f : \mathbb{N}^k \hookrightarrow \mathbb{N}$, is a function $f : D \to \mathbb{N}$ for some $D \subseteq \mathbb{N}^n$.

If $\overline{x} \in D$, we say that f *is defined on* $\overline{x}$ and write $f(\overline{x}) \downarrow$. Analogously, if $\overline{x} \in \mathbb{N}^k \setminus D$, we say that f *is not defined on* $\overline{x}$, in symbols: $f(\overline{x}) \uparrow$. If, for a partial function $f : \mathbb{N}^k \hookrightarrow \mathbb{N}$ and a $k \in \mathbb{N}$, we write $f(\overline{x}) = k$ this includes $f(\overline{x}) \downarrow$. Similarly, given a second partial function $g : \mathbb{N}^k \hookrightarrow \mathbb{N}$, if we write $f = g$, then this includes both the statement that the domain of g is equal to that of f and that f and g have the same value on every element of their domain. The definitions of composition and primitive recursion generalise naturally to partial functions (where a result of a function is only defined if all results required for computing it by the respective operator are defined).

Example 1.8. If $f : \mathbb{N} \hookrightarrow \mathbb{N}, x \mapsto \begin{cases} \frac{x}{2} & \text{if } x \text{ is even} \\ \text{undefined} & \text{otherwise} \end{cases}$ and $g : \mathbb{N} \hookrightarrow \mathbb{N}$ is defined by $g = \mathrm{Cn}[\cdot, \mathrm{c}^1_0, f]$, then $g(x) = \begin{cases} 0 & \text{if } x \text{ is even} \\ \text{undefined} & \text{otherwise} \end{cases}$.

In all programming languages there are constructs that allow to start a recursion or an iteration *without* knowing in advance how often it will be repeated. Instead a condition is given which decides when to terminate the recursion/iteration, for example `while`- or `repeat ... until`-loops in imperative programming languages. Functions defined using such loops are clearly computable in the intuitive sense. However, in such constructs we do not have a guarantee that the condition will eventually be met. The computation may not terminate. In case of non-termination the value of the function that is computed is not defined. In our setting of operator terms, this behaviour is modelled with the minimisation operator.

Definition 1.9. Let $f : \mathbb{N}^{k+1} \hookrightarrow \mathbb{N}$, then the function obtained from *minimisation* of f is $\mathrm{Mn}[f] = g : \mathbb{N}^k \hookrightarrow \mathbb{N}$, defined as

$$g(\overline{x}) = \begin{cases} y & \text{if } f(\overline{x}, y) = 0 \text{ and } \forall y' < y \, f(\overline{x}, y') \downarrow \text{ and } f(\overline{x}, y') \neq 0 \\ \text{undefined} & \text{if there is no such } y \end{cases}.$$

If f is computable, then so is g: we compute g by computing $f(\overline{x}, 0), f(\overline{x}, 1), \ldots$ until we find a y with $f(\overline{x}, y) = 0$. If one of the computations $f(\overline{x}, y')$ does not terminate, then the computation of g does not terminate. If all the computations of $f(\overline{x}, y')$ terminate but none of them yields 0, then the computation of g does not terminate.

We will often use the following notation: for an $f : \mathbb{N}^{k+1} \hookrightarrow \mathbb{N}$ we write $\mu y \, f(\overline{x}, y)$ for the function

$$\overline{x} \mapsto \begin{cases} \text{the smallest } y \text{ s.t. } f(\overline{x}, y) = 1 \text{ and } f(\overline{x}, y') = 0 \text{ for all } y' < y & \text{if such a } y \text{ exists} \\ \text{undefined} & \text{otherwise} \end{cases}$$

In particular, this notation will be useful if f is the characteristic function of a relation R. Then $\mu y\, \chi_R(\overline{x}, y)$ is the smallest y s.t. $R(\overline{x}, y)$, if there exists one. Because of this notation, Mn is often also referred to as μ-recursion.

Definition 1.10. A *partial recursive function* is a partial function $f : \mathbb{N}^n \hookrightarrow \mathbb{N}$ that can be obtained from the basic functions by a finite number of applications of the operators of composition, primitive recursion, and minimisation.

A *recursive function* is a partial recursive function which is total.

At this point we can pause again to ask whether we have characterised the set of computable functions (by the set of partial recursive functions). It is now important to observe that this statement cannot be proven mathematically since the notion "computable (in the intuitive sense)" is not mathematical. However, there exists a large number of formalisms for modelling computation which are based on different paradigms for machines or programs which all turn out to be equivalent in the sense that they can compute exactly the partial recursive functions. This situation has led to the *Church-Turing thesis*: a partial function is computable (in the intuitive sense) iff it is partial recursive. We can thus claim with reasonable confidence that we have characterised the computable functions.

We turn back to more technical matters now. A syntactic expression involving 0, S, P^n_k, Cn, Pr, and Mn that is formed according to the rules of Definitions 1.1, 1.2, 1.3, 1.9 is called *operator term*. We write $\mathcal{O}$ for the set of all operator terms and, for $k \in \mathbb{N}$, $\mathcal{O}_k$ for the set of all operator terms defining a k-ary function. For example, $\mathrm{Pr}[\mathrm{P}^1_1, \mathrm{Cn}[S, \mathrm{P}^3_3]] \in \mathcal{O}_2$. The primitive recursive (partial recursive) functions are closed under definition by cases:

Lemma 1.11. *If $g, f_0, \ldots, f_n : \mathbb{N}^k \hookrightarrow \mathbb{N}$ are primitive recursive (partial recursive), then so is $h : \mathbb{N}^k \hookrightarrow \mathbb{N}$ defined by*

$$h(\overline{x}) = \begin{cases} f_0(\overline{x}) & \text{if } g(\overline{x}) = 0 \\ f_1(\overline{x}) & \text{if } g(\overline{x}) = 1 \\ \vdots & \\ f_{n-1}(\overline{x}) & \text{if } g(\overline{x}) = n-1 \\ f_n(\overline{x}) & \text{if } g(\overline{x}) \geq n \\ \text{undefined} & \text{if } g(\overline{x}) \uparrow \end{cases}$$

Proof. We have $h(\overline{x}) = \chi_=(g(\overline{x}), 0) \cdot f_0(\overline{x}) + \cdots + \chi_=(g(\overline{x}), n-1) \cdot f_{n-1}(\overline{x}) + \chi_\geq(g(\overline{x}, n), f_n(\overline{x}))$. □

Example 1.12. $\min, \max : \mathbb{N}^2 \to \mathbb{N}$ are primitive recursive, since

$$\min(x, y) = \begin{cases} x_1 & \text{if } x_1 \leq x_2 \\ x_2 & \text{otherwise} \end{cases}, \text{ and}$$

$$\max(x, y) = \begin{cases} x_1 & \text{if } x_1 \geq x_2 \\ x_2 & \text{otherwise} \end{cases}.$$

1.2 Undecidability

Definition 1.13. A relation $R \subseteq \mathbb{N}^k$ is called *decidable* if $\chi_R : \mathbb{N}^k \to \{0,1\}$ is recursive.

Theorem 1.14. *There are undecidable sets.*

Proof. Every operator term is a finite string of symbols which are taken from a countable set. Therefore, there are only countably many operator terms, hence there are only countably many partial recursive functions, and thus, only countably many decidable relations. On the other hand, there are uncountably many $A \subseteq \mathbb{N}$. $\square$

The above proof is not very satisfactory because it does not give a concrete example of an undecidable set. We will now define the halting problem and prove it undecidable. The halting problem plays an important role in computability theory. In order to do that, we make some preliminary observations first: since there are only countably many operator terms that define partial functions of arity 1, there is a bijection from some subset C of $\mathbb{N}$, the set of "codes", to the set of such operator terms. For $e \in C$ we will write φ_e for the partial recursive function defined by the operator term with code e. For the time being, it is irrelevant which set C and which mapping $e \mapsto \varphi_e$ we pick. In Section 1.4 it will become relevant and we will give a concrete definition of C and the mapping $e \mapsto \varphi_e$.

Definition 1.15. The *halting problem* is $H = \{(e,x) \in C \times \mathbb{N} \mid \varphi_e(x)\downarrow\}$. Moreover, we define $K = \{e \in C \mid \varphi_e(e)\downarrow\}$.

Theorem 1.16. *K is undecidable.*

Proof. Define $f : \mathbb{N} \hookrightarrow \mathbb{N}$ by

$$f(n) = \begin{cases} 0 & \text{if } n \notin K \\ \text{undefined} & \text{if } n \in K \end{cases}$$

Suppose that K is decidable, i.e., χ_K is recursive, then f is partial recursive. Let $e \in \mathbb{N}$ be s.t. $f = \varphi_e$. Then we have

$$e \in K \overset{\text{Def. } f}{\iff} f(e) = \varphi_e(e) \text{ is undefined} \overset{\text{Def. } K}{\iff} e \notin K$$

which is a contradiction. $\square$

Corollary 1.17. *H is undecidable.*

Proof. Suppose $\chi_H : \mathbb{N}^2 \to \mathbb{N}$ would be recursive, then so would be $\chi_K : \mathbb{N} \to \mathbb{N}$ because $\chi_K(x) = \chi_H(x,x)$. $\square$

The main aim of the rest of this chapter is to prove that K is recursively enumerable, i.e., that there is a total recursive function $f : \mathbb{N} \to \mathbb{N}$ s.t. $f(\mathbb{N}) = K$. The existence of a recursively enumerable and undecidable set will then allow to obtain a first, weak, version of the first incompleteness theorem. In order to establish recursive enumerability of K we will have to code operator terms and computations as natural numbers. This technique, arithmetisation or "Gödelisation", in particular when used in conjunction with diagonalisation, is central, not only for the proof of the incompleteness theorems but for many results in mathematical logic.

1.3 Coding pairs, tuples, and trees

We will develop our coding machinery on a sufficiently general level to allow its reuse later when we code formulas and proofs. We start in this section with coding pairs, tuples, and trees. Before we do so, we need some more closure properties of the primitive recursive functions.

Lemma 1.18. *If $f : \mathbb{N}^{k+1} \to \mathbb{N}$ is primitive recursive, then so are:*

1. $(\overline{x}, z) \mapsto \sum_{y=0}^{z} f(\overline{x}, y)$,
2. $(\overline{x}, z) \mapsto \prod_{y=0}^{z} f(\overline{x}, y)$,
3. $(\overline{x}, z) \mapsto \min\{f(\overline{x}, y) \mid 0 \leq y \leq z\}$, *and*
4. $(\overline{x}, z) \mapsto \max\{f(\overline{x}, y) \mid 0 \leq y \leq z\}$.

Assuming in addition that $f : \mathbb{N}^{k+1} \to \{0, 1\}$, so are:

5. $(\overline{x}, z) \mapsto \forall y \leq z\, f(\overline{x}, y) = \begin{cases} 1 & \text{if for all } y \in \{0, \dots, z\}\text{: } f(\overline{x}, y) = 1 \\ 0 & \text{if there is } y \in \{0, \dots, z\} \text{ s.t. } f(\overline{x}, y) = 0 \end{cases}$,
6. $(\overline{x}, z) \mapsto \exists y \leq z\, f(\overline{x}, y) = \begin{cases} 1 & \text{if there is } y \in \{0, \dots, z\} \text{ s.t. } f(\overline{x}, y) = 1 \\ 0 & \text{if for all } y \in \{0, \dots, z\}\text{: } f(\overline{x}, y) = 0 \end{cases}$, *and*
7. $(\overline{x}, z) \mapsto (\mu y \leq z) f(\overline{x}, y) = \begin{cases} \text{the least } y \leq z \text{ s.t. } f(\overline{x}, y) = 1 & \text{if such a } y \text{ exists} \\ 0 & \text{otherwise} \end{cases}$.

Proof. For 1., note that the finite sum can be defined with primitive recursion as

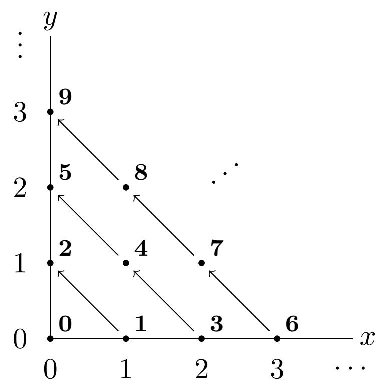
$$\sum_{y=0}^{0} f(\overline{x}, y) = f(\overline{x}, 0)$$
$$\sum_{y=0}^{z+1} f(\overline{x}, y) = \big(\sum_{y=0}^{z} f(\overline{x}, y)\big) + f(\overline{x}, z + 1).$$

For 2., 3., and 4. proceed analogously. If $f : \mathbb{N}^{k+1} \to \{0, 1\}$, then $\forall y \leq z\, f(\overline{x}, y) = \min\{f(\overline{x}, y) \mid 0 \leq y \leq z\}$ and $\exists y \leq z\, f(\overline{x}, y) = \max\{f(\overline{x}, y) \mid 0 \leq y \leq z\}$ which shows 5. and 6. For 7. define $f' : \mathbb{N}^{k+1} \to \mathbb{N}$ as

$$f'(\overline{x}, z) = \begin{cases} 1 & \text{if } f(\overline{x}, z) = 1 \text{ and } \forall z' < z\, f(\overline{x}, z') = 0 \\ 0 & \text{otherwise} \end{cases}$$

and observe that $(\mu y \leq z) f(\overline{x}, y) = \sum_{y=0}^{z} y \cdot f'(\overline{x}, y)$. □

We want to encode a pair of natural numbers as a single natural number. One option for doing that would be, e.g., to code (x, y) as $2^x 3^y$. However, we would like to i) avoid exponentiation and ii) obtain a bijection. Therefore we use the mapping illustrated in the following diagram:



This mapping from $\mathbb{N}^2$ to $\mathbb{N}$ is obviously bijective. Now we want to define it symbolically. To that aim, observe that pairs with the same sum are put on the same chain of arrows. Moreover, there is one pair with sum 0, two pairs with sum 1, etc. In general, there are $i+1$ pairs with sum i. Therefore, there are $\sum_{i=0}^{x+y-1}(i+1) = \sum_{i=1}^{x+y} i$ pairs with a sum less than $x+y$. On a fixed chain of arrows the code of the pair grows as the y-coordinate of the pair does. So we can define this bijection symbolically by

$$\langle x, y\rangle = (\sum_{i=1}^{x+y} i) + y = \frac{(x+y)(x+y+1)}{2} + y.$$

Note that $\langle x_1, y_1\rangle < \langle x_2, y_2\rangle$ iff $x_1 + y_1 < x_2 + y_2$ or ($x_1 + y_1 = x_2 + y_2$ and $y_1 < y_2$). Moreover, observe that $x, y \leq \langle x, y\rangle$ and that, if $(x, y) \notin \{(0,0),(1,0)\}$, then $x < \langle x, y\rangle$ and $y < \langle x, y\rangle$. Another noteworthy feature of this pairing function is that it permits a definition in the usual language of arithmetical theories (which contains addition and multiplication but not exponentiation) as $z = \langle x, y\rangle$ iff $2z = (x+y)(x+y+1)+2y$. We define the inverses of the pairing function $\mathrm{l} : \mathbb{N} \to \mathbb{N}, \langle x, y\rangle \mapsto x$ and $\mathrm{r} : \mathbb{N} \to \mathbb{N}, \langle x, y\rangle \mapsto y$. Based on this pairing function we can now proceed to code tuples.

Definition 1.19. For $k \geq 3$ define $\langle \cdot, \dots, \cdot\rangle : \mathbb{N}^k \to \mathbb{N}$ as $\langle x_1, \dots, x_k\rangle = \langle x_1, \langle x_2, \dots, x_k\rangle\rangle$. For $k = 1$ define $\langle \cdot\rangle : \mathbb{N} \to \mathbb{N}$ as the identity function.

For fixed $k \geq 1$, the function $\langle \cdot, \dots, \cdot\rangle$ is bijective. The union over these functions is *not* bijective, consider, e.g., $0 = \langle 0, 0\rangle = \langle 0, 0, 0\rangle = \cdots$

Lemma 1.20. *The following functions are primitive recursive:*

1. *for* $k \geq 1$*:* $\langle \cdot, \dots, \cdot\rangle : \mathbb{N}^k \to \mathbb{N}, (x_1, \dots, x_k) \mapsto \langle x_1, \dots, x_k\rangle$
2. $\pi : \mathbb{N}^3 \to \mathbb{N}, (k, i, x) \mapsto \begin{cases} x_i & \text{if } k \geq 1,\ 1 \leq i \leq k, \text{ and } x = \langle x_1, \dots, x_k\rangle \\ 0 & \text{if } k = 0,\ i = 0, \text{ or } i > k \end{cases}$

Proof. We first show 1. If $k = 1$, then $\langle \cdot\rangle = \mathrm{P}^1_1$ is primitive recursive. If $k \geq 2$, observe that, for an even number z, $\frac{z}{2} = (\mu z_0 \leq z)\, 2 \cdot z_0 = z$. Therefore the pairing function $\langle \cdot, \cdot\rangle : \mathbb{N}^2 \to \mathbb{N}$ is primitive recursive. For $k \geq 3$, we obtain $\langle \cdot, \dots, \cdot\rangle : \mathbb{N}^k \to \mathbb{N}$ by composing the pairing function with itself a suitable number of times.

For 2., note that

$$l(z) = (\mu x \leq z)(\exists y \leq z)\, \langle x, y\rangle = z \text{ and}$$
$$r(z) = (\mu y \leq z)(\exists x \leq z)\, \langle x, y\rangle = z.$$

So both l and r are primitive recursive. Therefore, also $(j, z) \mapsto r^j(z)$ is primitive recursive. We have

$$\pi(k, i, x) = \begin{cases} r^{i-1}(x) & \text{if } k \geq 1 \text{ and } i = k \\ l(r^{i-1}(x)) & \text{if } k \geq 1 \text{ and } 1 \leq i < k \\ 0 & \text{otherwise} \end{cases}$$

and therefore also π is primitive recursive. $\square$

Now that we have primitive recursive tuples we can show another useful closure property: the primitive recursive functions are closed under course-of-value recursion. To that aim define first:

Definition 1.21. Let $h : \mathbb{N}^{k+1} \to \mathbb{N}$. The *history function* $\hat{h} : \mathbb{N}^{k+1} \to \mathbb{N}$ of h is defined as $\hat{h}(\overline{x}, y) = \langle h(\overline{x}, y), \ldots, h(\overline{x}, 0)\rangle$.

Lemma 1.22. *If $f : \mathbb{N}^k \to \mathbb{N}$ and $g : \mathbb{N}^{k+2} \to \mathbb{N}$ are primitive recursive, then so is the function $h : \mathbb{N}^{k+1} \to \mathbb{N}$ defined by:*

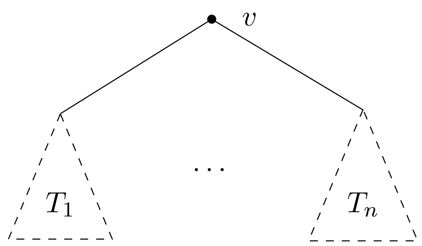
$$h(\overline{x}, 0) = f(\overline{x}) \text{ and}$$
$$h(\overline{x}, y + 1) = g(\overline{x}, y, \hat{h}(\overline{x}, y)).$$

Proof. It suffices to show that $\hat{h}$ is primitive recursive because $h(\overline{x}, y) = \pi(y + 1, 1, \hat{h}(\overline{x}, y))$. To this aim, note that

$$\hat{h}(\overline{x}, 0) = h(\overline{x}, 0) = f(\overline{x}) \text{ and}$$
$$\hat{h}(\overline{x}, y + 1) = \langle h(\overline{x}, y + 1), \hat{h}(\overline{x}, y)\rangle = \langle g(\overline{x}, y, \hat{h}(\overline{x}, y)), \hat{h}(\overline{x}, y)\rangle.$$

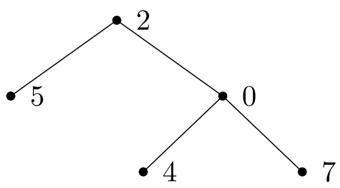
$\square$

As a next step, we want to encode finite ordered trees, i.e., the order of subtrees is significant, whose vertices are labelled by natural numbers. Each such tree T will be encoded as a natural number $\#T$. We use our tuple encoding and define the code of a tree by induction on the structure of the tree: a tree of the form $T =$



is encoded as $\#T = \langle v, n, \#T_1, \ldots, \#T_n\rangle$ where $\#T_i$ is the code of the subtree T_i. Note that this definition includes the case $\langle v, 0\rangle$ for a leaf. Also note that # is just a function from trees to natural numbers without any a priori connection to primitive recursion or computability theory.

Example 1.23. The code of the ordered labelled tree



is $\langle 2, 2, \langle 5, 0\rangle, \langle 0, 2, \langle 4, 0\rangle, \langle 7, 0\rangle\rangle\rangle$ which is the 84-digit natural number

$$120443650830443822950654392810134061331537938301945868395455743743602923276498173998.$$

If the natural number $m = \langle v, n, m_1, \ldots, m_n\rangle$ is given, then $v = l(m)$ and $n = l(r(m))$. Therefore, v, n, m_1, …, m_n are determined uniquely by m. Furthermore, $n > 0$ implies that $m_i < m$ for $i = 1, \ldots, n$. So, by induction, we can conclude that the mapping # from ordered labelled trees to $\mathbb{N}$ is bijective. Moreover, the functions $m \mapsto n$ and $(m, i) \mapsto m_i$ are primitive recursive.

Lemma 1.24 (Tree recursion)**.** *If $f : \mathbb{N} \to \mathbb{N}$ and $g : \mathbb{N}^4 \to \mathbb{N}$ are primitive recursive, then so is $h : \mathbb{N} \to \mathbb{N}$, defined by*

$$h(\langle v, n, x_1, \ldots, x_n\rangle) = \begin{cases} f(v) & \text{if } n = 0 \\ g(v, n, \langle x_1, \ldots, x_n\rangle, \langle h(x_1), \ldots, h(x_n)\rangle) & \text{if } n > 0 \end{cases}$$

Proof. First note that $v, n, x_1, \ldots, x_n$ are all well-defined since # from ordered trees to $\mathbb{N}$ is injective. Moreover, they can be computed by primitive recursive functions from $\langle v, n, x_1, \ldots, x_n\rangle$. Since # is also surjective, h is a total function. Furthermore, note that, for $n > 0$, $x_i < \langle v, n, x_1, \ldots, x_n\rangle$ for all $i \in \{1, \ldots, n\}$. Therefore $\langle h(x_1), \ldots, h(x_n)\rangle$ can be computed by a primitive recursive function, based on the projection π, from the value $\hat{h}(\langle v, n, x_1, \ldots, x_n\rangle)$ of the history function of h. So, by course-of-values recursion, h is primitive recursive. $\square$

1.4 The enumeration theorem

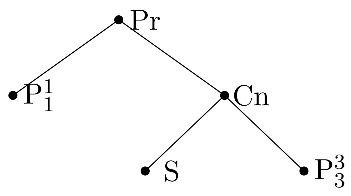
The key to our proof of the recursive enumerability of the halting set K is the enumeration theorem. The enumeration theorem shows the existence of a universal partial recursive function, i.e., a single partial recursive function that, given the code of any partial recursive function f and some input $\overline{x}$ for f can compute $f(\overline{x})$. Our proof will proceed via Kleene's normal form theorem which also entails that a single use of the minimisation operator is enough to compute any partial recursive function. In order to prove these results, we first have to code operator terms. To that aim, given that we know how to code labelled trees, it is sufficient to assign unique codes to the operators.

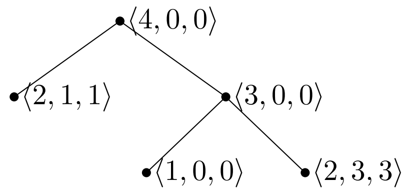
Definition 1.25. We assign codes to operators as follows:

$$\begin{aligned} \#0 &= \langle 0, 0, 0\rangle & \#\mathrm{Cn} &= \langle 3, 0, 0\rangle \\ \#\mathrm{S} &= \langle 1, 0, 0\rangle & \#\mathrm{Pr} &= \langle 4, 0, 0\rangle \\ \#\mathrm{P}^k_i &= \langle 2, k, i\rangle & \#\mathrm{Mn} &= \langle 5, 0, 0\rangle \end{aligned}$$

The code of an operator term is given by a function $\# : \mathcal{O} \to \mathbb{N}$ and is defined, by induction on the structure of $t \in \mathcal{O}$, as the code of the tree whose labels are determined by the operators.

Example 1.26. In Example 1.5 we have seen that $+ = \mathrm{Pr}[\mathrm{P}^1_1, \mathrm{Cn}[\mathrm{S}, \mathrm{P}^3_3]]$. As a tree this is

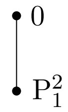




with operators on the left and their codes on the right. The code of this tree is the natural number

$$\langle\langle 4,0,0\rangle,2,\langle\langle 2,1,1\rangle,0\rangle,\langle\langle 3,0,0\rangle,2,\langle\langle 1,0,0\rangle,0\rangle,\langle\langle 2,3,3\rangle,0\rangle\rangle\rangle.$$

The mapping $\# : \mathcal{O} \to \mathbb{N}$ is injective since $t \in \mathcal{O}$ uniquely determines its tree. However, it is not surjective anymore since there are operator-labelled trees which do not correspond to an operator term, e.g.,



We can now make precise the set of codes C of unary functions and the injective function $e \mapsto \varphi_e$ that was mentioned above. For C we simply take $\#\mathcal{O}_1$.

Definition 1.27. For $e \in \#\mathcal{O}_k$ we write φ_e for the partial recursive function from $\mathbb{N}^k$ to $\mathbb{N}$ defined by the operator term $\#^{-1}(e)$.

Lemma 1.28. *The characteristic function* $\chi_{\#\mathcal{O}} : \mathbb{N} \to \{0,1\}$*, the function* $(k,m) \mapsto \chi_{\#\mathcal{O}_k}(m)$*, and the function* $\mathrm{ar} : \mathbb{N} \to \mathbb{N}, m \mapsto \begin{cases} k & \textit{if } m \in \#\mathcal{O}_k \\ 0 & \textit{if } m \notin \#\mathcal{O} \end{cases}$ *are primitive recursive.*

Note that $\mathrm{ar}(m) = 0$ is an ambiguous case since it is not clear whether $m \in \#\mathcal{O}_0$ or $m \notin \#\mathcal{O}$. Nevertheless the function ar is quite natural as it computes the arity of an operator term. The ambiguity is resolved by using ar only on such m where we have checked that $\chi_{\#\mathcal{O}}(m) = 1$ before. This is analogous to the use of bounded minimisation together with the bounded existential quantifier.

Proof. It suffices to show that $\mathrm{ar}' : \mathbb{N} \to \mathbb{N}, m \mapsto \begin{cases} k+1 & \text{if } m \in \#\mathcal{O}_k \\ 0 & \text{if } m \notin \#\mathcal{O} \end{cases}$ is primitive recursive because $\chi_{\#\mathcal{O}}(m) = \begin{cases} 1 & \text{if } \mathrm{ar}'(m) \geq 1 \\ 0 & \text{otherwise} \end{cases}$, $\chi_{\#\mathcal{O}_k}(m) = \begin{cases} 1 & \text{if } \mathrm{ar}'(m) = k+1 \\ 0 & \text{otherwise} \end{cases}$, and $\mathrm{ar}(m) = \mathrm{p}(\mathrm{ar}'(m))$. The function ar' is defined via tree recursion from the following $f : \mathbb{N} \to \mathbb{N}$:

$$v \mapsto \begin{cases} 1 & \text{if } v = \#0 \\ 2 & \text{if } v = \#\mathrm{S} \\ k+1 & \text{if } v = \#\mathrm{P}^k_i \\ 0 & \text{otherwise} \end{cases}$$

and $g : \mathbb{N}^4 \to \mathbb{N}$:

$$(v,n,\langle m_1,\ldots,m_n\rangle,\langle \mathrm{ar}'(m_1),\ldots,\mathrm{ar}'(m_n)\rangle) \mapsto \begin{cases} k+1 & \text{if } v = \#\mathrm{Cn},\ \mathrm{ar}'(m_1) = n, \text{ and} \\ & \mathrm{ar}'(m_2) = \cdots = \mathrm{ar}'(m_n) = k+1 \\ \mathrm{ar}'(m_1)+1 & \text{if } v = \#\mathrm{Pr},\ n = 2, \text{ and} \\ & \mathrm{ar}'(m_1)+2 = \mathrm{ar}'(m_2) \\ \mathrm{ar}'(m_1)-1 & \text{if } v = \#\mathrm{Mn},\ n = 1, \text{ and} \\ & \mathrm{ar}'(m_1) \geq 2 \\ 0 & \text{otherwise} \end{cases}.$$

$\square$

The next, and for this chapter: final, type of objects we want to encode are computation trees.

Definition 1.29. Let $k \in \mathbb{N}$, $t \in \mathcal{O}_k$, $x_1, \dots, x_k \in \mathbb{N}$, and $y \in \mathbb{N}$. We will encode the equation $t(x_1, \dots, x_k) = y$ as the tuple $\langle \#t, x_1, \dots, x_k, y \rangle$. A *computation tree* for an equation is a tree whose vertices are codes of equations subject to the following conditions:

1. A computation tree for $0 = 0$ consists of the single node

$$\bullet \ \langle \#0, 0 \rangle$$

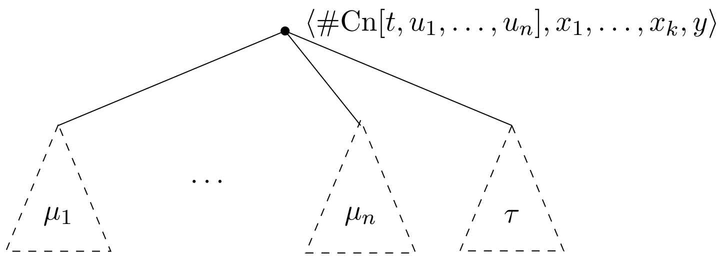
2. A computation tree for $\mathrm{S}(x) = x + 1$ consists of the single node

$$\bullet \ \langle \#\mathrm{S}, x, x+1 \rangle$$

3. A computation tree for $\mathrm{P}^k_i(x_1, \dots, x_k) = x_i$ consists of the single node

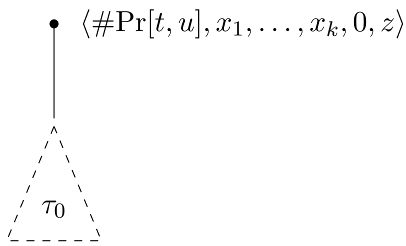
$$\bullet \ \langle \#\mathrm{P}^k_i, x_1, \dots, x_k, x_i \rangle$$

4. A computation tree for $\mathrm{Cn}[t, u_1, \dots, u_n](x_1, \dots, x_k) = y$ is of the form



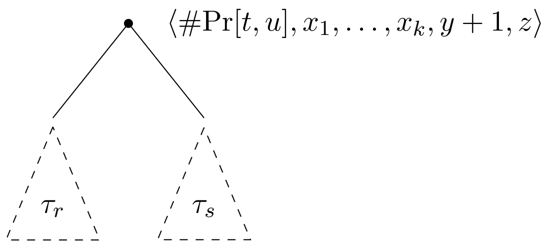
where μ_i is a computation tree for $u_i(\overline{x}) = y_i$ for some $y_i \in \mathbb{N}$ and τ is a computation tree for $t(y_1, \dots, y_n) = y$.

5. A computation tree for $\mathrm{Pr}[t, u](x_1, \dots, x_k, 0) = z$ is of the form



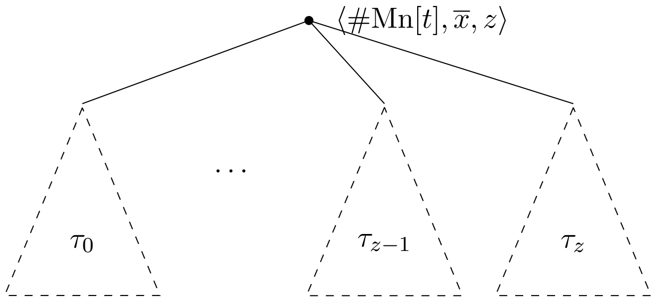
where τ_0 is a computation tree for $t(x_1, \dots, x_k) = z$.

A computation tree for $\mathrm{Pr}[t, u](x_1, \dots, x_k, y+1) = z$ is of the form



where τ_r is a computation tree for $\mathrm{Pr}[t,u](\overline{x},y) = z'$ for some $z' \in \mathbb{N}$ and τ_s is a computation tree for $u(\overline{x},y,z') = z$.

6. A computation tree for $\mathrm{Mn}[t](\overline{x}) = z$ is of the form



where, for $i \in \{0,\dots,z\}$, τ_i is a computation tree of $t(\overline{x},i) = y_i$ for some $y_i \in \mathbb{N}$, $y_z = 0$ and, for $i \in \{0,\dots,z-1\}$, $y_i > 0$.

It is straightforward to verify that $t(\overline{x}) = y$ iff there is a computation tree for $t(\overline{x}) = y$. Note that, if $t(\overline{x}) \uparrow$, then some Mn-node cannot finish its computation based on finitely many subtrees. Since all our trees are finite, the existence of a computation tree implies termination of the computation.

Lemma 1.30. *The set $T = \{n \in \mathbb{N} \mid n$ is code of a computation tree$\}$ is primitive recursive.*

Proof. The characteristic function $\chi_T : \mathbb{N} \to \{0,1\}$ is obtained from tree recursion as

$$\chi_T(\langle v,n,x_1,\dots,x_n\rangle) = \begin{cases} f(v) & \text{if } n = 0 \\ g(v,n,\langle x_1,\dots,x_n\rangle, \langle \chi_T(x_1),\dots,\chi_T(x_n)\rangle) & \text{if } n > 0 \end{cases}$$

from primitive recursive functions $f : \mathbb{N} \to \mathbb{N}$ and $g : \mathbb{N}^4 \to \mathbb{N}$. The function f returns 1 if v is a label of a leaf according to Definition 1.29/1.-3. and 0 otherwise. The function g first makes a case distinction: if there is an $i \in \{1,\dots,n\}$ s.t. $\chi_T(x_i) = 0$ then return 0. If $\chi_T(x_1) = \dots = \chi_T(x_n) = 1$, then g returns 1 if v is label of the root of a computation tree with immediate subtrees $x_1,\dots,x_n$ according to Definition 1.29/4.-6. and 0 otherwise. □

Theorem 1.31 (Kleene's normal form theorem)**.** *There is a primitive recursive function $P : \mathbb{N} \to \mathbb{N}$ and, for each $k \geq 0$, a primitive recursive predicate $T_k \subseteq \mathbb{N}^{k+2}$ s.t. for all $e \in \#\mathcal{O}_k$ and all $x_1,\dots,x_k \in \mathbb{N}$:*

$$\varphi_e(x_1,\dots,x_k) = P(\mu y\, T_k(e,x_1,\dots,x_k,y)).$$

Proof. Let

$$T_k = \{(e,x_1,\dots,x_k,y) \in \mathbb{N}^{k+2} \mid e = \#t \text{ for some } t \in \mathcal{O}_k \text{ and } y \text{ is code of a computation tree of } t \text{ on input } x_1,\dots,x_k\}$$

and observe that T_k is primitive recursive. P first computes $l = \mathrm{ar}(e)$ and then obtains the $l+2$-nd element of the label of the root of its input y. This is a definition of a primitive recursive function. When applied to a y s.t. $(e,x_1,\dots,x_k,y) \in T_k$, this yields the value of φ_e on input $x_1,\dots,x_k$. □

Corollary 1.32 (Enumeration theorem)**.** *For every $k \geq 0$ there is a partial recursive function $U_k : \mathbb{N}^{k+1} \hookrightarrow \mathbb{N}$ s.t. for all $e \in \#\mathcal{O}_k$ and all $x_1, \ldots, x_k \in \mathbb{N}$:*

$$\varphi_e(x_1, \ldots, x_k) = U_k(e, x_1, \ldots, x_k).$$

The enumeration theorem states one of the central properties of the partial recursive functions: the existence of a universal function, i.e., a function capable of computing the value of any partial recursive function (with the right arity) on any input. There are (at least) two perspectives on this result: mathematically, this is a uniformity property. From the point of view of computer science, a universal function is just an interpreter: a program that executes another program.

1.5 Recursively enumerable sets

Definition 1.33. A relation $R \subseteq \mathbb{N}^k$ is called *recursively enumerable (r.e.)* if R is the domain of a partial recursive function.

The terminology "recursively enumerable" is explained by the following property:

Lemma 1.34. *Let $A \subseteq \mathbb{N}$, then the following are equivalent:*

1. *A is r.e.*
2. *A is the range of a partial recursive function*
3. *$A = \emptyset$ or A is the range of a primitive recursive function*

Proof. For 1. $\Rightarrow$ 2. let $f : \mathbb{N} \hookrightarrow \mathbb{N}$ with $\mathrm{dom}(f) = A$ and define $g : \mathbb{N} \hookrightarrow \mathbb{N}$ by the operator term corresponding to $x \mapsto x + 0 \cdot f(x)$, i.e., $g = \mathrm{Cn}[+, \mathrm{P}^1_1, \mathrm{Cn}[\cdot, \mathrm{c}^1_0, f]]$. Then $g(x)\downarrow$ iff $f(x)\downarrow$ and in that case: $g(x) = x$. Therefore $\mathrm{rng}(g) = \mathrm{dom}(f) = A$.

For 2. $\Rightarrow$ 3. let $f : \mathbb{N} \hookrightarrow \mathbb{N}$ with $\mathrm{rng}(f) = A$. If $A = \emptyset$ we are done. So let $A \neq \emptyset$, let $a \in A$. Remember from the proof of the normal form theorem that the relation

$$T_1 = \{(e, x, y) \in \mathbb{N}^3 \mid e \in \#\mathcal{O}_1 \text{ and } y \text{ is code of a computation tree of } \#^{-1}(e) \text{ on input } x\}$$

and the output function P are primitive recursive. Let $c \in \#\mathcal{O}_1$ be a code of (an operator term computing) f. Define $g : \mathbb{N} \to \mathbb{N}$ by

$$g(\langle x, k\rangle) = \begin{cases} U((\mu y \leq k)\, \chi_{T_1}(c, x, y)) & \text{if } (\exists y \leq k)\, \chi_{T_1}(c, x, y) \\ a & \text{otherwise} \end{cases}$$

which satisfies $\mathrm{rng}(g) = \mathrm{rng}(f) = A$ and is primitive recursive.

For 3. $\Rightarrow$ 1. observe that $\emptyset$ is the domain of the partial recursive function that is defined nowhere. If $A \neq \emptyset$, let $A = \mathrm{rng}(f)$ for a primitive recursive $f : \mathbb{N} \to \mathbb{N}$. Define $g : \mathbb{N} \hookrightarrow \mathbb{N}, y \mapsto \mu x\, f(x) = y$ and observe that $\mathrm{dom}(g) = A$. $\square$

In the previous lemma we have restricted our attention to subsets A of $\mathbb{N}$. As the following observation shows, this is not a significant restriction.

Definition 1.35. For $R \subseteq \mathbb{N}^k$ we write $\langle R\rangle$ for the set $\{\langle x_1, \ldots, x_k\rangle \mid (x_1, \ldots, x_k) \in R\} \subseteq \mathbb{N}$.

Lemma 1.36. *$R \subseteq \mathbb{N}^k$ is primitive recursive (decidable, r.e.) iff $\langle R \rangle$ is.*

Proof. This follows immediately from the observation that

$$\chi_R(x_1, \dots, x_k) = \chi_{\langle R \rangle}(\langle x_1, \dots, x_k \rangle) \quad \text{and} \quad \chi_{\langle R \rangle}(x) = \chi_R(\pi(k, 1, x), \dots, \pi(k, k, x)).$$

□

A fundamental property of r.e. sets is the following

Lemma 1.37. *$R \subseteq \mathbb{N}^k$ is decidable iff both R and $\mathbb{N}^k \setminus R$ are r.e.*

Proof. For the left-to-right direction note that, if $\chi_R : \mathbb{N}^k \to \{0, 1\}$ is recursive, then so are $\chi_R^+ : \mathbb{N}^k \hookrightarrow \mathbb{N}, \overline{x} \mapsto \begin{cases} 0 & \text{if } \chi_R(\overline{x}) = 1 \\ \text{undefined} & \text{if } \chi_R(\overline{x}) = 0 \end{cases}$ and $\chi_R^- : \mathbb{N}^k \hookrightarrow \mathbb{N}, \overline{x} \mapsto \begin{cases} \text{undefined} & \text{if } \chi_R(\overline{x}) = 1 \\ 0 & \text{if } \chi_R(\overline{x}) = 0 \end{cases}$. Moreover, $R = \mathrm{dom}(\chi_R^+)$ and $\mathbb{N}^k \setminus R = \mathrm{dom}(\chi_R^-)$.
For the right-to-left direction we can, by Lemma 1.36, assume that $k = 1$. So $R \subseteq \mathbb{N}$ and there are recursive functions $f^+, f^- : \mathbb{N} \to \mathbb{N}$ s.t. $\mathrm{rng}(f^+) = R$ and $\mathrm{rng}(f^-) = \mathbb{N} \setminus R$. Define $g(x) = \mu y\, (f^+(y) = x \text{ or } f^-(y) = x)$ and note that $g : \mathbb{N} \to \mathbb{N}$ is total. Moreover

$$\chi_R(x) = \begin{cases} 1 & \text{if } f^+(g(x)) = x \\ 0 & \text{if } f^-(g(x)) = x \end{cases}$$

which is a well-formed case distinction because, for every $x \in \mathbb{N}$, exactly one of $f^+(g(x)) = x$ and $f^-(g(x)) = x$ is true. □

In particular, the above lemma entails that every decidable set is r.e. The converse is not true. In Section 1.2 we have defined the halting set $K = \{e \in C \mid \varphi_e(e) \downarrow\}$ without defining either C nor $e \mapsto \varphi_e$ concretely. Later we have made the mapping $e \mapsto \varphi_e$ concrete as the partial recursive function defined by the operator term $\#^{-1}(e)$ where $\#$ is our specific coding. We have also made C concrete as $C = \#\mathcal{O}_1$, so

$$K = \{e \in \#\mathcal{O}_1 \mid \varphi_e(e) \downarrow\}$$

and we obtain

Corollary 1.38. *K is recursively enumerable and undecidable.*

Proof. Undecidability has already been shown in Theorem 1.16. For recursive enumerability, consider the universal function $U_1 : \mathbb{N}^2 \hookrightarrow \mathbb{N}, (e, x) \mapsto \varphi_e(x)$. U_1 is partial recursive by the enumeration theorem, thus so is $V : \mathbb{N} \hookrightarrow \mathbb{N}, e \mapsto U_1(e, e) = \varphi_e(e)$ and $K = \mathrm{dom}(V)$. □

Chapter 2

Arithmetical definability

2.1 The arithmetical hierarchy

We will now start to consider formulas in first-order predicate logic. The language of arithmetic is $L_A = \{0, s, +, \cdot, \leq\}$. An L_A formula is also called *arithmetical formula*. We work in first-order logic with equality. We will often write $\varphi(x_1, \ldots, x_k)$ for a formula whose free variables are among $\{x_1, \ldots, x_k\}$. Unless otherwise stated, we do not assume that all x_i occur in φ. For terms $t_1, \ldots, t_k$ which do not contain any variable bound in φ we then write $\varphi(t_1, \ldots, t_k)$ for the result of substituting all x_i by t_i in parallel. We use an analogous notational convention for terms writing $t(x_1, \ldots, x_k)$ and $t(t_1, \ldots, t_n)$. For terms s and t we write $s \neq t$ as abbreviation of $\neg s = t$. For a formula $\varphi(x, \overline{z})$ we write $\exists! x\, \varphi(x, \overline{z})$ as an abbreviation for $\exists x\, (\varphi(x, \overline{z}) \land \forall y\, (\varphi(y, \overline{z}) \rightarrow x = y))$. We write $\equiv$ for syntactic equality of formulas and terms. For $i \in \mathbb{N}$ and a term t we define the term $s^i(t)$ by $s^0(t) :\equiv t$ and $s^{i+1}(t) :\equiv s(s^i(t))$. For $n \in \mathbb{N}$ the *numeral* $\underline{n}$ is defined as the term $s^n(0)$. If $\mathcal{M}$ is an L structure, where L is some first-order language, then $\mathrm{Th}(\mathcal{M}) = \{\sigma \text{ is an } L \text{ sentence} \mid \mathcal{M} \models \sigma\}$. In particular, $\mathrm{Th}(\mathbb{N})$ is the set of L_A sentences which are true in $\mathbb{N}$.

Definition 2.1. Let $R \subseteq \mathbb{N}^k$, an arithmetical formula $\varphi(x_1, \ldots, x_k)$ *defines* R if

$$\mathbb{N} \models \varphi(\underline{n_1}, \ldots, \underline{n_k}) \text{ iff } (n_1, \ldots, n_k) \in R.$$

A relation $R \subseteq \mathbb{N}^k$ is called *arithmetically definable* if there is a formula $\varphi(x_1, \ldots, x_k)$ which defines R.

Example 2.2. The set of even numbers is defined by the arithmetical formula

$$\mathsf{Even}(x) \quad \equiv \quad \exists y\, y \cdot \underline{2} = x.$$

The set of prime numbers is defined by the arithmetical formula

$$\mathsf{Prime}(x) \quad \equiv \quad \forall y\, (\exists z\, z \cdot y = x \rightarrow y = 1 \lor y = x) \land x \neq \underline{1}.$$

The set of prime numbers can also be defined by the arithmetical formula

$$\varphi(x) \quad \equiv \quad \forall y_1 \forall y_2\, (\exists z\, z \cdot x = y_1 \cdot y_2 \rightarrow (\exists z\, z \cdot x = y_1 \lor \exists z\, z \cdot x = y_2)) \land x \neq \underline{1} \land x \neq 0.$$

So we see that a single set can be defined by different formulas. Thus there is a certain arbitrariness in fixing a *particular* formula as a definition, much as there is in picking a particular operator term for computing a function. In order to distinguish between a set or relation and the particular arithmetical formula we pick for defining it, we will use sans-serif font for the formula. We say that two formulas $\varphi_1(\overline{x})$ and $\varphi_2(\overline{x})$ are *equivalent* if they define the same relation, i.e., if $\mathbb{N} \models \forall \overline{x}\,(\varphi_1(\overline{x}) \leftrightarrow \varphi_2(\overline{x}))$. We say that $\varphi_1(\overline{x})$ and $\varphi_2(\overline{x})$ are *logically equivalent* if the formula $\forall \overline{x}\,(\varphi_1(\overline{x}) \leftrightarrow \varphi_2(\overline{x}))$ is valid.

Definition 2.3. If t is a term which does not contain x and φ a formula, we define $\exists x \leq t\, \varphi$ as abbreviation for $\exists x\,(x \leq t \wedge \varphi)$ and $\forall x \leq t\, \varphi$ as abbreviation for $\forall x\,(x \leq t \rightarrow \varphi)$. $\exists x \leq t$ and $\forall x \leq t$ are called *bounded quantifiers.*

Occasionally we will also write $x < y$ which is an abbreviation for the formula $x \leq y \wedge x \neq y$. Correspondingly, $\exists x < t\, \varphi$ is an abbreviation for $\exists x \leq t\,(x \neq t \wedge \varphi)$ and $\forall x < t\, \varphi$ is an abbreviation of $\forall x \leq t\,(x \neq t \rightarrow \varphi)$.

Definition 2.4. A formula is called *bounded* if all its quantifiers are bounded. We define $\Sigma_0 = \Pi_0 =$ the set of bounded formulas. For $n \geq 0$ we define the sets of formulas $\Sigma_{n+1} = \{\exists x\, \varphi \mid \varphi \in \Pi_n\}$ and $\Pi_{n+1} = \{\forall x\, \varphi \mid \varphi \in \Sigma_n\}$.

Definition 2.5. Let $n \geq 0$ and $R \subseteq \mathbb{N}^k$. Then R is called Σ_n-definable if there is a Σ_n-formula which defines R and Π_n-definable if there is a Π_n-formula which defines R. R is called Δ_n-definable if it is both Σ_n- and Π_n-definable.

We also refer to the bounded formulas as Δ_0 formulas. The following lemma entails, in particular, that every L_A formula is equivalent to a Σ_n formula for some $n \in \mathbb{N}$ as well as to a Π_m formula for some $m \in \mathbb{N}$.

Lemma 2.6. *If $n \geq 0$, then:*

1. *If $R \subseteq \mathbb{N}^k$ is Σ_n-definable (Π_n-definable), then $\mathbb{N}^k \setminus R$ is Π_n-definable (Σ_n-definable).*
2. *The Δ_n-definable relations are closed under complementation.*
3. *The Σ_{n+1}-definable relations are closed under existential quantification.*
4. *The Π_{n+1}-definable relations are closed under universal quantification.*
5. *The Σ_n-, Π_n-, and Δ_n-definable relations are closed under union and intersection.*
6. *The Σ_n-, Π_n-, and Δ_n-definable relations are closed under bounded quantification.*

Proof. 1. follows from the observation that, for any Σ_n formula φ, the formula $\neg\varphi$ is logically equivalent to a Π_n formula and vice versa. 2. is an immediate corollary of 1.

We prove 3. and 4. simultaneously by induction on n. Let $\exists z\, \varphi(\overline{x}, y, z)$ be Σ_{n+1}, i.e., $\varphi(\overline{x}, y, z)$ is Π_n. Then $\exists y \exists z\, \varphi(\overline{x}, y, z)$ is equivalent to

$$\psi(\overline{x}) \quad \equiv \quad \exists u \forall y \forall z\,(u = \langle y, z\rangle \rightarrow \varphi(\overline{x}, y, z))$$

as well as to

$$\psi_{\mathrm{b}}(\overline{x}) \quad \equiv \quad \exists u \forall y \leq u \forall z \leq u\,(u = \langle y, z\rangle \rightarrow \varphi(\overline{x}, y, z))$$

where $u = \langle y, z\rangle$ is an abbreviation for $\underline{2} \cdot u = (y+z) \cdot (y+z+1) + \underline{2} \cdot z$. If $n = 0$, then $\varphi(\overline{x}, y, z)$ is Π_0 and $\psi_{\mathrm{b}}(\overline{x})$ is a Σ_1 formula. If $n > 0$, then $\varphi(\overline{x}, y, z)$ is a Π_n formula and thus $\psi(\overline{x})$ is equivalent, due to the quantifier shifts in predicate logic and the induction hypothesis, to a Σ_{n+1} formula. For 4. let $\varphi(\overline{x}, y)$ be a Π_{n+1} formula, then $\forall y\, \varphi(\overline{x}, y)$ defines a relation $R \subseteq \mathbb{N}^k$ which is also defined by $\neg\exists y \neg\varphi(\overline{x}, y)$. Now, as in 1., $\neg\varphi(\overline{x}, y)$ is equivalent to a Σ_{n+1} formula, so by the case for Σ_{n+1}, $\exists y \neg\varphi(\overline{x}, y)$ is equivalent to a Σ_{n+1} formula, so, again as in 1., $\neg\exists y\neg\varphi(\overline{x}, y)$ is equivalent to a Π_{n+1}-formula.

For 5., first observe that the statement is trivial for $n = 0$, so let $n > 0$. If $\exists y\, \varphi(\overline{x}, y)$ and $\exists z\, \psi(\overline{x}, z)$ are Σ_n formulas, then $\exists y\, \varphi(\overline{x}, y) \wedge \exists z\, \psi(\overline{x}, z)$ is logically equivalent to $\exists y \exists z\, (\varphi(\overline{x}, y) \wedge \psi(\overline{x}, z))$ which is equivalent to a Σ_n formula by 3. Similarly, $\exists y\, \varphi(\overline{x}, y) \vee \exists z\, \varphi(\overline{x}, z)$ is logically equivalent to $\exists y\, (\varphi(\overline{x}, y) \vee \psi(\overline{x}, y))$ which is Σ_n too. The cases for Π_n are analogous. The cases for Δ_n follow from those of Σ_n and Π_n.

For 6., proceed by induction on n. The case $n = 0$ is trivial. For $n \geq 1$ first note that $\exists y \leq t \exists z\, \varphi(\overline{x}, y, z)$ is logically equivalent to $\exists z \exists y \leq t\, \varphi(\overline{x}, y, z)$ and similarly for two universal quantifiers. Let $\varphi_0(x_1, \ldots, x_k, y, z)$ be Π_n, then $\exists z\, \varphi_0(x_1, \ldots, x_k, y, z)$ is a Σ_{n+1} formula and we claim that

$$\varphi(x_1, \ldots, x_k) \quad \equiv \quad \forall y \leq t(x_1, \ldots, x_k)\, \exists z\, \varphi_0(x_1, \ldots, x_k, y, z)$$

is equivalent to

$$\psi(x_1, \ldots, x_k) \quad \equiv \quad \exists w\, \forall y \leq t(x_1, \ldots, x_k)\, \exists z \leq w\, \varphi_0(x_1, \ldots, x_k, y, z).$$

For proving the claim first observe that $\forall x_1 \cdots \forall x_k\, (\psi(x_1, \ldots, x_k) \to \varphi(x_1, \ldots, x_k))$ is valid. For the other direction, let $\mathbb{N} \models \varphi(\underline{n_1}, \ldots, \underline{n_k})$ and let $m \in \mathbb{N}$ s.t. $\mathbb{N} \models t(\underline{n_1}, \ldots, \underline{n_k}) = \underline{m}$. Then, for every $i \in \{0, \ldots, m\}$, there is a $q_i \in \mathbb{N}$ s.t. $\mathbb{N} \models \varphi_0(\underline{n_1}, \ldots, \underline{n_k}, \underline{i}, \underline{q_i})$. Let $q = \max\{q_i \mid 0 \leq i \leq n\}$, then $\mathbb{N} \models \forall y \leq t(\underline{n_1}, \ldots, \underline{n_k}) \exists z \leq \underline{q}\, \varphi_0(\underline{n_1}, \ldots, \underline{n_k}, y, z)$, i.e., $\mathbb{N} \models \psi(\underline{n_1}, \ldots, \underline{n_k})$. By induction hypothesis, $\forall y \leq t(x_1, \ldots, x_k) \exists z \leq w\, \varphi_0(x_1, \ldots, x_k, y, z)$ is equivalent to a Π_n formula and therefore $\psi(x_1, \ldots, x_k)$ and hence $\varphi(x_1, \ldots, x_k)$ are equivalent to a Σ_{n+1} formula.

For the remaining case, let $\varphi_0(x_1, \ldots, x_k, y, z)$ be Σ_n, then $\forall z\, \varphi_0(x_1, \ldots, x_k, y, z)$ is Π_{n+1} and $\varphi(x_1, \ldots, x_k) \equiv \exists y \leq t(x_1, \ldots, x_k)\, \forall z\, \varphi_0(x_1, \ldots, x_k, y, z)$ is logically equivalent to $\neg\varphi'(x_1, \ldots, x_k)$ where $\varphi'(x_1, \ldots, x_k) \equiv \forall y \leq t(x_1, \ldots, x_k)\, \exists z\, \neg\varphi_0(x_1, \ldots, x_k, y, z)$ and φ' is Σ_{n+1} by the previous case. Then, by 6., φ is equivalent to a Π_{n+1}-formula.

Since both, the Σ_n- and the Π_n-definable relations are closed under both bounded quantifiers, so is Δ_n. $\square$

The Σ_n, Π_n, and Δ_n sets form the arithmetical hierarchy, see Figure 2.1 for a graphical representation. We will later show that the arithmetical hierarchy is strict, i.e., each two nodes represent different sets. An arrow from a node X to a node Y indicates that all X sets are Y sets but not vice versa, i.e., $X \subset Y$. Subset inclusions that follow from transitivity are left implicit. The arithmetical hierarchy is intimately tied to computability theory through Post's theorem. We will neither formulate nor prove this result here but instead continue this course in its direction towards logic. We merely need a part of a special case of Post's theorem here, the result that the Σ_1-definable sets are the r.e. sets. This will entail directly that the Δ_1-definable sets are the decidable sets.

2.2 Definability and computability

We proceed to study the relationship between arithmetical definability and computability. Our main results in this section will be that a relation is Σ_1-definable iff it is r.e. and decidable iff it is Δ_1. To that aim we first observe:

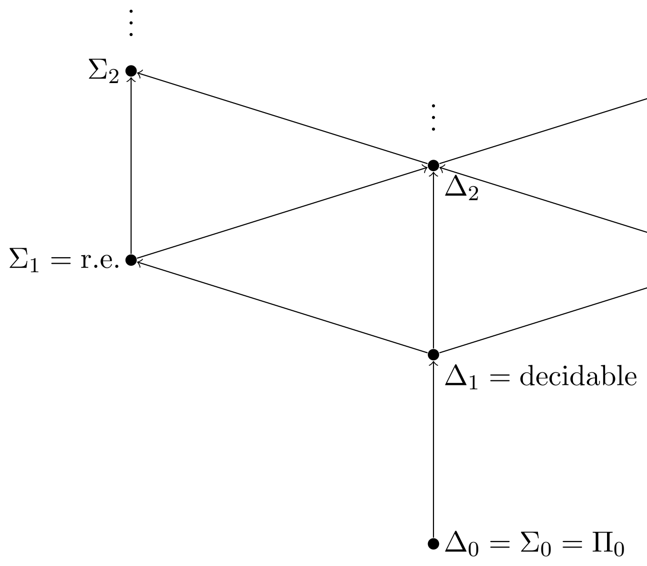


Figure 2.1: The arithmetical hierarchy

Lemma 2.7. *If $\varphi(x_1, \ldots, x_k)$ is a Δ_0 formula, then the function $\chi_\varphi : \mathbb{N}^k \to \mathbb{N}$ defined by*

$$\chi_\varphi(n_1, \ldots, n_k) = \begin{cases} 1 & \text{if } \mathbb{N} \models \varphi(\underline{n_1}, \ldots, \underline{n_k}) \\ 0 & \text{if } \mathbb{N} \not\models \varphi(\underline{n_1}, \ldots, \underline{n_k}) \end{cases}$$

is primitive recursive.

Proof. By induction on the logical complexity of φ. W.l.o.g. φ does not contain implications. If φ is an atom $s(x_1, \ldots, x_k) = t(x_1, \ldots, x_k)$ or $s(x_1, \ldots, x_k) \leq t(x_1, \ldots, x_k)$, then χ_φ is primitive recursive because 0, S, $+$, and $\cdot$, as well as $\chi_=$ and $\chi_\leq$ are primitive recursive. For the connectives $\wedge$ and $\vee$ it suffices to observe that

$$\chi_{\varphi_1 \wedge \varphi_2}(n_1, \ldots, n_k) = \min\{\chi_{\varphi_1}(n_1, \ldots, n_k), \chi_{\varphi_2}(n_1, \ldots, n_k)\} \text{ and}$$
$$\chi_{\varphi_1 \vee \varphi_2}(n_1, \ldots, n_k) = \max\{\chi_{\varphi_1}(n_1, \ldots, n_k), \chi_{\varphi_2}(n_1, \ldots, n_k)\}.$$

because min and max are primitive recursive. If $\varphi = \neg\varphi_0$ we have $\chi_\varphi(n_1, \ldots, n_k) = 1 \dot{-} \chi_{\varphi_0}(n_1, \ldots, n_k)$ and both the constant 1-function as well as $\dot{-}$ are primitive recursive. For the bounded quantifiers, let $\varphi(\overline{x})$ be $Qy \leq t(\overline{x})\, \varphi_0(\overline{x}, y)$ and note that, since, by induction hypothesis, $\chi_{\varphi_0} : \mathbb{N}^{k+1} \to \mathbb{N}$ is primitive recursive, so is $\chi_\varphi : \mathbb{N}^k \to \mathbb{N}$ by Lemma 1.18/5. and 6. □

A crucial ingredient for the clarification of the relation between arithmetical definability and computability is a definition of sequences of arbitrary length by an arithmetical formula. We already have the pairing function $\langle \cdot, \cdot \rangle$ which allows to express $z = \langle x, y \rangle$ as the arithmetical formula $\underline{2} \cdot z = (x + y) \cdot (x + y + 1) + \underline{2} \cdot y$. Iterating this allows to give, for every $k \geq 2$, an arithmetical formula φ_k which defines the codes of k-tuples. What we want, however, are two formula $\mathsf{Seq}(w, v)$ and $\varphi(w, u, x)$ which uniformly, i.e., independently of k, define "w is a sequence of length v" and "the u-th element of w is x". In order to simplify the notation we will notate $\varphi(w, u, x)$ as $(w)_u = x$. There are such formulas, even Δ_0 formulas, but they require a subtle construction. More precisely:

Lemma 2.8. *There are Δ_0 formulas $\mathsf{Seq}(w,v)$ and $(w)_u = x$ s.t. the following formulas are true in $\mathbb{N}$:*

$$\mathsf{Seq}(w,v) \to \forall u < v \exists! x\, (w)_u = x \tag{S1}$$
$$\exists w\, \mathsf{Seq}(w,0) \tag{S2}$$
$$\mathsf{Seq}(w,v) \to \forall x \exists w' \,(\mathsf{Seq}(w', s(v)) \wedge \forall u < v \forall y\, ((w')_u = y \leftrightarrow (w)_u = y) \wedge (w')_v = x) \tag{S3}$$

Keep in mind that that $(w)_u = x$ is not an equation, it is merely a suggestive notation for a formula $\varphi(w,u,x)$ with the free variables w,u,x. (S1), (S2), and (S3) can be considered axioms of a theory of lists. (S1) ensures that $\mathsf{Seq}(w,v)$ and $(w)_u = x$ have their intended interpretations. (S2) asserts the existence of the empty sequence. (S3) ensures that we can append an element to a sequence. In particular, (S2) and (S3) entail the existence of all finite lists. There are several different Δ_0 encodings of sequences that satisfy the above properties. We will construct one in the exercises and prove Lemma 2.8 there.

Example 2.9. Applying (S2) and (S3) to the finite sequence $2,3,5,7$ yields an $m \in \mathbb{N}$ which, by (S1), satisfies:

$$\begin{aligned}
&\mathbb{N} \models \mathsf{Seq}(\underline{m}, \underline{4}),\\
&\mathbb{N} \models (\underline{m})_0 = x \leftrightarrow x = \underline{2},\\
&\mathbb{N} \models (\underline{m})_{\underline{1}} = x \leftrightarrow x = \underline{3},\\
&\mathbb{N} \models (\underline{m})_{\underline{2}} = x \leftrightarrow x = \underline{5}, \text{and}\\
&\mathbb{N} \models (\underline{m})_{\underline{3}} = x \leftrightarrow x = \underline{7}.
\end{aligned}$$

Definition 2.10. Let $f : \mathbb{N}^k \hookrightarrow \mathbb{N}$. The *graph of f* is the set $\Gamma_f = \{(x_1,\dots,x_k,y) \in \mathbb{N}^{k+1} \mid f(x_1,\dots,x_k) = y\}$. We say that an arithmetical formula $\varphi(x_1,\dots,x_k,y)$ *defines* f if it defines Γ_f.

Theorem 2.11. *$f : \mathbb{N}^k \hookrightarrow \mathbb{N}$ is recursive iff f is Σ_1-definable.*

Proof. For the right-to-left direction, assume that $\exists z\, \varphi(x_1,\dots,x_k,y,z)$ is a Σ_1 definition of Γ_f, i.e., φ is Σ_0 and

$$f(n_1,\dots,n_k) = m \text{ iff } \mathbb{N} \models \exists z\, \varphi(\underline{n_1},\dots,\underline{n_k},\underline{m},z).$$

Let $g : \mathbb{N}^k \hookrightarrow \mathbb{N}, (n_1,\dots,n_k) \mapsto (\mu u)\, \chi_\varphi(n_1,\dots,n_k,\pi(2,1,u),\pi(2,2,u))$. Then $f(n_1,\dots,n_k) = \pi(2,1,g(n_1,\dots,n_k))$. Since φ is Σ_0, by Lemma 2.7, χ_φ is primitive recursive and so are the other constructors except the μ-recursion. Therefore f is partial recursive.

For the left-to-right direction, we will show that the set of partial functions whose graph has a Σ_1 definition contains the basic functions and is closed under composition, primitive recursion, and minimisation. The graph of the nullary zero function is defined by $\varphi(y) \equiv y = 0$, the graph of the successor function by $\varphi(x,y) \equiv y = s(x)$, and the graph of the projection function P^k_i by $\varphi(x_1,\dots,x_k,y) \equiv y = x_i$.

If the graphs of $f : \mathbb{N}^n \hookrightarrow \mathbb{N}$ and $g_1,\dots,g_n : \mathbb{N}^k \hookrightarrow \mathbb{N}$ have Σ_1 definitions $\varphi(y_1,\dots,y_n,z)$ and $\psi_i(x_1,\dots x_k,y_i)$ for $1 \le i \le n$, then the graph of $\mathrm{Cn}[f,g_1,\dots,g_n]$ is defined by

$$\exists y_1 \cdots \exists y_n \,(\bigwedge_{i=1}^{n} \psi_i(x_1,\dots,x_n,y_i) \wedge \varphi(y_1,\dots,y_n,z))$$

which is equivalent to a Σ_1 formula.

If $h = \mathrm{Pr}[f, g] : \mathbb{N}^{k+1} \hookrightarrow \mathbb{N}$ where the graphs of $f : \mathbb{N}^k \hookrightarrow \mathbb{N}$ and $g : \mathbb{N}^{k+2} \hookrightarrow \mathbb{N}$ have Σ_1 definitions $\varphi(x_1, \ldots, x_k, y)$ and $\psi(x_1, \ldots, x_k, y, z, w)$ respectively, then we define the graph of h via the existence of the finite sequence $h(\overline{x}, 0), h(\overline{x}, 1), \ldots, h(\overline{x}, y)$ of intermediate results in the computation of $h(\overline{x}, y)$. By Lemma 2.8 this is expressed by the formula

$$\begin{aligned}\chi(\overline{x}, y, z) \equiv \exists w \, \big(\mathsf{Seq}(w, y+1) \wedge \exists v \, ((w)_0 = v \wedge \varphi(\overline{x}, v)) \wedge (w)_y = z \wedge \\ \forall u < y \exists r \exists r' \, ((w)_u = r \wedge (w)_{s(u)} = r' \wedge \psi(\overline{x}, u, r, r'))\big)\end{aligned}$$

which, by Lemma 2.6, is equivalent to a Σ_1 formula.

If $g = \mathrm{Mn}[f] : \mathbb{N}^k \hookrightarrow \mathbb{N}$ where the graph of $f : \mathbb{N}^{k+1} \hookrightarrow \mathbb{N}$ has a Σ_1 definition $\varphi(\overline{x}, y, z)$, then the graph of g is defined by

$$\psi(\overline{x}, y) \equiv \varphi(\overline{x}, y, 0) \wedge \forall u < y \exists z \, (\varphi(\overline{x}, u, z) \wedge z \neq 0).$$

By Lemma 2.6, $\psi(\overline{x}, y)$ is equivalent to a Σ_1 formula. □

Theorem 2.11 can be extended from functions to sets as follows.

Corollary 2.12. *$R \subseteq \mathbb{N}^k$ is r.e. iff R is Σ_1-definable.*

Proof. For the left-to-right direction let $R \subseteq \mathbb{N}^k$ be r.e. Then there is $f : \mathbb{N}^k \hookrightarrow \mathbb{N}$ s.t. $\mathrm{dom}(f) = R$. By Theorem 2.11 there is a Σ_1 formula $\psi(x_1, \ldots, x_k, y)$ that defines Γ_f. Therefore $\varphi(x_1, \ldots, x_k) \equiv \exists y \, \psi(x_1, \ldots, x_k, y)$ is equivalent to a Σ_1 formula that defines R.

For the right-to-left direction, let $\varphi(x_1, \ldots, x_k)$ be a Σ_1 definition of an $R \subseteq \mathbb{N}^k$. Define

$$f : \mathbb{N}^k \hookrightarrow \mathbb{N}, (n_1, \ldots, n_k) \mapsto \begin{cases} 0 & \text{if } \mathbb{N} \models \varphi(\underline{n_1}, \ldots, \underline{n_k}) \\ \text{undefined} & \text{otherwise} \end{cases}$$

then $\mathrm{dom}(f) = R$. Furthermore $\psi(x_1, \ldots, x_k, y) \equiv y = 0 \wedge \varphi(x_1, \ldots, x_k)$ is equivalent to a Σ_1 formula defining f and so, by Theorem 2.11, f is partial recursive and hence R is r.e. □

Corollary 2.13. *$R \subseteq \mathbb{N}^k$ is decidable iff R is Δ_1-definable.*

Proof. $R \subseteq \mathbb{N}^k$ is decidable iff both R and $\mathbb{N}^k \setminus R$ are r.e. iff both R and $\mathbb{N}^k \setminus R$ are Σ_1-definable, i.e., R is Σ_1-definable and Π_1-definable, i.e., Δ_1-definable. □

Corollary 2.14. *There is a Σ_1-definable set that is not Δ_1-definable.*

Proof. The halting set K is r.e. but not decidable, i.e., Σ_1-definable but not Δ_1-definable. □

Theorem 2.11 and Corollaries 2.12 and 2.14 can be strengthened considerably based on the following famous result, the MRDP theorem, named after Y. Matiyasevič, J. Robinson, M. Davis, and H. Putnam.

Theorem 2.15. *For every arithmetical Σ_1 formula $\varphi(x_1, \ldots, x_k)$ there is an equivalent formula*

$$\exists y_1 \cdots \exists y_n \, \psi(x_1, \ldots, x_k, y_1, \ldots, y_n)$$

where ψ is quantifier-free.

The crucial point of this result is that ψ does not even contain bounded quantifiers. A proof of this theorem is beyond the scope of this course. The last part of its proof was completed in 1970 thus providing an answer to Hilbert's 10th problem which was posed in 1900: does there exist an algorithm which, given a Diophantine equation, i.e., an equation of the form $p(x_1, \dots, x_k) = 0$ where $p \in \mathbb{Z}[x_1, \dots, x_n]$ is a polynomial in the variables $x_1, \dots, x_n$, determines whether it has an integer solution, i.e., whether there are $a_1, \dots, a_k \in \mathbb{Z}$ s.t. $p(a_1, \dots, a_k) = 0$. If there was such an algorithm, then it could be modified to decide an arbitrary r.e. set, in particular the halting set.

2.3 Coding formulas

Just as we have considered operator terms that receive (codes of) operator terms as input in Chapter 1, we now want to consider formulas that talk about (codes of) formulas. To that aim, we will develop an encoding of formulas. We code formulas in a language L having for each $n \geq 0$ the n-ary function symbols $f_0^n, f_1^n, \dots$ and the n-ary relation symbols $R_0^n, R_1^n, \dots$. The only propositional connectives are $\neg$ and $\to$, the only quantifier is $\forall$. The other connectives and the existential quantifier are considered to be abbreviations. The variables appearing in formulas are taken from the fixed set $\{x_i \mid i \in \mathbb{N}\}$. For coding formulas we essentially proceed as we did for operator terms: by using trees. We write $\mathcal{T}(L)$ for the set of terms in the language L and $\mathcal{F}(L)$ for the set of formulas in the language L.

Definition 2.16. We assign codes to variables and to function symbols of L as follows.

$$x_i \mapsto \langle 0, i \rangle \qquad\qquad f_i^n \mapsto \langle n+1, i \rangle$$

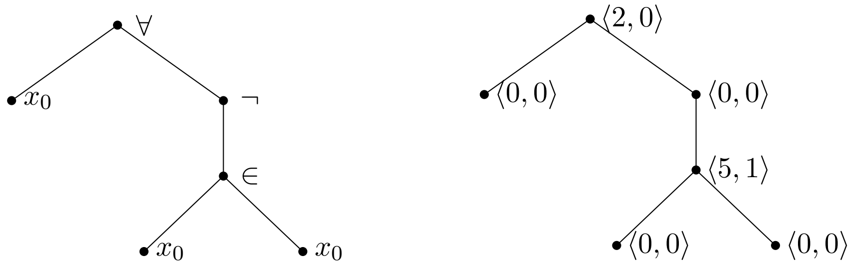
The code of a term is given by a function $\# : \mathcal{T}(L) \to \mathbb{N}$ which is defined as code of the tree whose labels are determined by variables and function symbols.

We assign codes to logical symbols and predicate symbols of L as follows.

$$\neg \mapsto \langle 0, 0 \rangle \qquad \to \mapsto \langle 1, 0 \rangle \qquad \forall \mapsto \langle 2, 0 \rangle \qquad R_i^n \mapsto \langle n+3, i \rangle$$

The code of a formula is given by a function $\# : \mathcal{F}(L) \to \mathbb{N}$ which is defined as code of the tree whose labels are determined by the logical connectives. A universal quantifier induces a node with two children: the first being the variable, the second the formula. An atom with predicate symbol R_i^n induces a node with n children, the i-th being the tree representing the i-th term.

Example 2.17. For the language $L = \{= /2, \in /2\} = \{R_0^2, R_1^2\}$ of set theory, the L-formula $\forall x_0\, \neg x_0 \in x_0$ is encoded as:



Example 2.18. For $L_A = \{0, s, +, \cdot, \leq\} = \{f_0^0, f_0^1, f_0^2, f_1^2, R_0^2\}$ the first two numerals are encoded as

$$\bullet\ 0 \quad = \quad \bullet\ f_0^0 \quad = \quad \bullet\ \langle 1, 0 \rangle$$

which has the code $\langle\langle 1,0\rangle,0\rangle = 1$, i.e., $\#\underline{0} = 1$, and

$$\begin{tikzcd} \bullet\ s \arrow[d, dash] \\ \bullet\ 0 \end{tikzcd} \quad = \quad \begin{tikzcd} \bullet\ f_0^1 \arrow[d, dash] \\ \bullet\ f_0^0 \end{tikzcd} \quad = \quad \begin{tikzcd} \bullet\ \langle 2,0\rangle \arrow[d, dash] \\ \bullet\ \langle 1,0\rangle \end{tikzcd}$$

which has the code $\langle\langle 2,0\rangle,1,\langle\langle 1,0\rangle,0\rangle\rangle = 32$, i.e., $\#\underline{1} = 32$.

Definition 2.19. A set of terms (formulas) is said to be *recursively enumerable*, *decidable*, *primitive recursive* if its set of codes is.

The set of (codes of) numerals is primitive recursive (just use tree recursion to check if the encoded term has the required form). The function which maps a formula to its set of free variables (based on the primitive recursive coding of finite sets developed in the exercises) is primitive recursive. The set of L-formulas is primitive recursive, the set of L-sentences is primitive recursive, just check if the set of free variables of the given formula φ is $\emptyset$. Checking whether a formula is Σ_n (Π_n), for any $n \geq 0$, is primitive recursive.

Definition 2.20. Let x_i be a variable, t be a term. The application of the substitution $[x_i\backslash t]$ to a term is defined by:

$$f_j^n(t_1,\ldots,t_n)[x_i\backslash t] = f_j^n(t_1[x_i\backslash t],\ldots,t_n[x_i\backslash t]) \qquad x_j[x_i\backslash t] = \begin{cases} t & \text{if } i=j \\ x_j & \text{otherwise} \end{cases}$$

Definition 2.21. Let x_i be a variable, t be a term, and φ a formula s.t. t does not contain a variable that occurs bound in φ. Then $\varphi[x\backslash t]$, the application of the substitution $[x\backslash t]$ to φ, is defined by:

$$R_j^n(t_1,\ldots,t_n)[x_i\backslash t] = R_j^n(t_1[x_i\backslash t],\ldots,t_n[x_i\backslash t]) \qquad (\neg\psi)[x_i\backslash t] = \neg\psi[x_i\backslash t]$$

$$(\psi\to\chi)[x_i\backslash t] = \psi[x_i\backslash t]\to\chi[x_i\backslash t] \qquad (\forall x_j\,\psi)[x_i\backslash t] = \begin{cases} \forall x_j\,\psi & \text{if } i=j \\ \forall x_j\,\psi[x_i\backslash t] & \text{if } i\neq j \end{cases}$$

Note that these definitions are primitive recursive. Therefore there is a Σ_1 formula $\mathsf{Subst}(x,y,z,u)$ which defines substitution, i.e., $\mathbb{N} \models \mathsf{Subst}(\underline{m},\underline{n},\underline{k},\underline{l})$ iff $m = \#\varphi$, $n = \#x_i$, $k = \#t$, and $l = \#\varphi[x_i\backslash t]$. It is often useful to abbreviate $\underline{\#a}$, the numeral of the code of some object a, as $\ulcorner a\urcorner$. For example, we have $\mathbb{N} \models \mathsf{Subst}(\ulcorner\varphi\urcorner,\ulcorner x_i\urcorner,\ulcorner t\urcorner,\ulcorner\varphi[x_i\backslash t]\urcorner)$. Note that $\underline{\#a}$, the numeral of the code of some object a, is different from $\#\underline{n}$, the code of the numeral of some $n \in \mathbb{N}$.

Remark 2.22. It is possible to define substitution without the condition of t not containing any variable that is bound in φ. However, this definition, requiring renaming of bound variables, is a little more cumbersome and the present one suffices for our purposes. Since we want to work with the formalised definition we will therefore use this simpler one.

Based on our coding of formulas and the existence of a recursively enumerable but undecidable set, we are now in a position to prove our first result with some impact on the foundations of mathematics.

Theorem 2.23. Th($\mathbb{N}$) *is not recursively enumerable.*

Proof. Let A be r.e. but undecidable, then A has a Σ_1 definition $\varphi(x)$, i.e., $A = \{n \in \mathbb{N} \mid \mathbb{N} \models \varphi(\underline{n})\}$. Let $B = \{n \in \mathbb{N} \mid \mathbb{N} \not\models \varphi(\underline{n})\} = \{n \in \mathbb{N} \mid \mathbb{N} \models \neg\varphi(\underline{n})\}$, then $A \uplus B = \mathbb{N}$. Suppose Th($\mathbb{N}$) is r.e., then also B is r.e. (just check whether the value of the recursive enumeration function of Th($\mathbb{N}$) is of the form $\neg\varphi(\underline{n})$), so, by Lemma 1.37, A would be decidable. Contradiction. $\square$

This result is a semantic, and thus weaker, variant of the first incompleteness theorem. It impacts the foundations of mathematics in that it shows that truth in the natural numbers cannot be axiomatised in a reasonably simple way, or, put differently, every reasonably simple attempt at an axiomatisation of the natural numbers is incomplete. What does "attempt at an axiomatisation" mean here? At the very least we would like our set of axioms to be sound, i.e., true in $\mathbb{N}$. What does "reasonably simple" mean here? For the following argument it is enough to make the modest request that the set of axioms shall be recursively enumerable. Then, if A is a r.e. set of true arithmetical sentences, then $\{\sigma \mid A \vdash \sigma\}$ is incomplete, i.e., there is a true arithmetical sentence τ s.t. $A \nvdash \tau$. For suppose, for the sake of contradiction, that $A \vdash \tau$ for all true arithmetical sentences τ, then, since $\mathbb{N} \models A$, we would even have $\{\sigma \mid A \vdash \sigma\} = \mathrm{Th}(\mathbb{N})$. Since A is r.e. there is a recursive enumeration of all proofs from axioms of A and hence of all formulas provable from A, i.e., of the set $\{\sigma \mid A \vdash \sigma\} = \mathrm{Th}(\mathbb{N})$, contradicting Theorem 2.23.

For the foundations of mathematics this result has the consequence that, at least in principle, we have to consider a third possibility when we deal with a mathematical statement σ based on some fixed axiomatisation T. In addition to being provable in T and to being refutable in T, i.e., $\neg\sigma$ being provable in T, it may be the case that neither σ nor $\neg\sigma$ are provable in T. A famous example for such a situation is the independence of the continuum hypothesis, the statement that there is no set whose cardinality is strictly between that of the natural numbers and that of the real numbers, from ZFC, Zermelo Fraenkel set theory with the axiom of choice. In how far this incompleteness phenomenon impacts the daily work of mathematicians is still a subject of current research.

Beyond the result of Theorem 2.23 itself, also its proof is of considerable interest to us. We will encounter variants of this proof repeatedly in this course. It transfers our main result on computability, the existence of a set which is r.e. but not decidable, to a logical context in order to obtain a negative result there. The first incompleteness theorem that we will see later is stronger than Theorem 2.23 which still has the following weaknesses: on the one hand it is non-constructive in the sense that it does not yield a particular sentence which is true but unprovable. On the other hand it is semantic in the sense that it talks about the standard model. We will see later that incompleteness is a very general phenomenon that also occurs in unsound theories. But also on the purely semantic layer a much stronger result than Theorem 2.23 is true: not only is there no Σ_1 definition of $\mathrm{Th}(\mathbb{N})$, there is no arithmetical formula whatsoever that defines $\mathrm{Th}(\mathbb{N})$. However, for showing this stronger result, carrying out a diagonalisation on the level of the halting problem is not enough. Instead we have to diagonalise in the setting of arithmetical formulas.

2.4 On the definability of truth

In this section we will study the arithmetical definability of the set $\mathrm{Th}(\mathbb{N})$. The main result will be Tarski's theorem: $\mathrm{Th}(\mathbb{N})$ is not arithmetically definable. However, it will turn out that for all $n \in \mathbb{N}$ the set $\{\sigma \in \mathrm{Th}(\mathbb{N}) \mid \sigma \text{ is a } \Sigma_n \text{ formula}\}$ is arithmetically definable. For showing Tarski's theorem we will prove a first, semantic, version of the fixed point lemma, or diagonal lemma, which will later also play a central role in the proofs of the incompleteness theorems. The version we prove now is restricted in that it only applies to truth in the standard model $\mathbb{N}$. Later we will prove an extension to a large class of formal theories of arithmetic.

Lemma 2.24 (Fixed point lemma). *Let $\varphi(x)$ be an arithmetical formula. Then there is an arithmetical sentence σ s.t. $\mathbb{N} \models \sigma \leftrightarrow \varphi(\ulcorner\sigma\urcorner)$. Moreover, if $\varphi(x)$ is Σ_n for some $n \geq 1$, then σ can be chosen to be Σ_n.*

The sentence σ is a fixed point of the mapping $\chi \mapsto \varphi(\ulcorner\chi\urcorner)$ modulo equivalence in $\mathbb{N}$, hence the name of the lemma. Note that σ refers to itself in the sense that, up to equivalence in $\mathbb{N}$, it states: "I have the property φ".

Proof. The key to this result is the definition of a formula $\psi(x)$ that acts like $\varphi(x)$ on any formula $\chi(x)$ except that it applies its argument to itself first, i.e.,

$$\mathbb{N} \models \psi(\ulcorner\chi(x)\urcorner) \leftrightarrow \varphi(\ulcorner\chi(\ulcorner\chi(x)\urcorner)\urcorner) \tag{*}$$

for all formulas $\chi(x)$. Then, applying $\psi(x)$ to itself, we obtain

$$\mathbb{N} \models \psi(\ulcorner\psi(x)\urcorner) \leftrightarrow \varphi(\ulcorner\psi(\ulcorner\psi(x)\urcorner)\urcorner)$$

thus using the duplication ability of the outer ψ on the inner ψ (on the left-hand side) to reproduce ψ applied to itself (on the right-hand side). Therefore, by letting $\sigma \equiv \psi(\ulcorner\psi(x)\urcorner)$, we have

$$\mathbb{N} \models \sigma \leftrightarrow \varphi(\ulcorner\sigma\urcorner).$$

It remains to define ψ and to show (*). To that aim first define $f : \mathbb{N} \to \mathbb{N}$ by

$$n \mapsto \begin{cases} \#\chi(\ulcorner\chi(x)\urcorner) & \text{if } n = \#\chi(x) \text{ for a formula } \chi(x) \\ 0 & \text{otherwise} \end{cases}$$

and note that f is primitive recursive (on input n, check whether $n = \#\chi$ and $\mathrm{FV}(\chi) = \{x\}$ for some formula χ and some variable x, if yes return $\chi[x\backslash\underline{n}]$). So, by Theorem 2.11, there is a Σ_1 definition $F(x, y)$ of f. We define $\psi(x)$ as, or, if necessary, as a Σ_n formula equivalent to, $\exists y\,(F(x, y) \land \varphi(y))$ and obtain

$$\begin{aligned} \mathbb{N} \models \psi(\ulcorner\chi(x)\urcorner) &\leftrightarrow \exists y\,(F(\ulcorner\chi(x)\urcorner, y) \land \varphi(y)) \\ &\leftrightarrow \exists y\,(y = \underline{f(\#\chi(x))} \land \varphi(y)) \\ &\leftrightarrow \exists y\,(y = \ulcorner\chi(\ulcorner\chi(x)\urcorner)\urcorner \land \varphi(y)) \\ &\leftrightarrow \varphi(\ulcorner\chi(\ulcorner\chi(x)\urcorner)\urcorner). \end{aligned}$$

$\square$

Example 2.25. Applying the fixed point theorem to $\mathsf{Even}(x)$ we obtain a sentence σ s.t. $\mathbb{N} \models \sigma \leftrightarrow \mathsf{Even}(\ulcorner\sigma\urcorner)$, i.e., up to equivalence in $\mathbb{N}$, σ states "My code is an even number.". This does not tell us whether σ is true in $\mathbb{N}$, it merely tells us that σ is true in $\mathbb{N}$ iff $\#\sigma$ is even.

Typically we will apply the fixed point theorem to properties of sentences rather than numbers. Then the meaning of the sentence does not refer to codes (explicitly).

Theorem 2.26 (Undefinability of truth (Tarski))**.** $\mathrm{Th}(\mathbb{N})$ *is not arithmetically definable.*

Proof. Suppose that $\mathrm{Tr}(x)$ is an arithmetical formula that defines the true arithmetical sentences, i.e., $\mathbb{N} \models \mathrm{Tr}(\underline{n})$ iff $n = \#\sigma$ for some arithmetical sentence σ with $\mathbb{N} \models \sigma$. So, for every sentence σ, $\mathbb{N} \models \sigma \leftrightarrow \mathrm{Tr}(\ulcorner\sigma\urcorner)$. Then, by the fixed point lemma applied to $\neg\mathrm{Tr}(x)$, there is a sentence τ s.t. $\mathbb{N} \models \tau \leftrightarrow \neg\mathrm{Tr}(\ulcorner\tau\urcorner)$, i.e., τ expresses "I am not true". Then $\mathbb{N} \models \mathrm{Tr}(\ulcorner\tau\urcorner) \leftrightarrow \neg\mathrm{Tr}(\ulcorner\tau\urcorner)$, contradiction. $\square$

Even though, as we have just seen, a (complete) truth definition is impossible, partial truth definitions are possible in the sense that the truth of Σ_n, or Π_n, sentences is arithmetically definable for all $n \geq 0$. In order to show this, we start by quickly recalling the definition of the satisfaction relation. If $\mathcal{M} = (M, I)$ is an L structure, φ is an L formula, and v is a variable evaluation for $\mathcal{M}$ and φ, i.e., v is a mapping from variables to elements of M with $\mathrm{dom}(v) = \mathrm{FV}(\varphi)$, then $(M, I, v) \models \varphi$ is defined. In particular, if φ starts with a quantifier, we have:

$$(M, I, v) \models \forall x\, \psi \text{ iff for all } m \in M\text{: } (M, I, v[x \backslash m]) \models \psi$$
$$(M, I, v) \models \exists x\, \psi \text{ iff there is an } m \in M \text{ s.t. } (M, I, v[x \backslash m]) \models \psi$$

Our strategy for obtaining a partial truth definition will be to follow this inductive definition of the satisfaction relation $\models$. Since we are interested in arithmetical truth, (M, I) is fixed to $\mathbb{N}$, so we have to define a binary relation on codes of formulas and codes of variable evaluations for $\mathbb{N}$. We have already developed an encoding of formulas in Section 2.3. A primitive recursive encoding of variable evaluations is sufficient for our purposes and can be developed in a quite straightforward way by relying on the existing encoding of variables and considering a variable evaluation for $\mathbb{N}$ as a finite set of variable/number-pairs. From now on, we assume a fixed such coding of variable evaluations for $\mathbb{N}$.

Definition 2.27. For $n \geq 0$ we define the relation $\mathrm{Sat}_{\Sigma,n} \subseteq \mathbb{N} \times \mathbb{N}$ as follows:

$$\begin{aligned}(k, l) \in \mathrm{Sat}_{\Sigma,n} \text{ iff } & k = \#\varphi \text{ for some } \Sigma_n \text{ formula } \varphi,\\ & l = \#v \text{ for some variable evaluation } v \text{ for } \mathbb{N} \text{ and } \varphi, \text{ and}\\ & (\mathbb{N}, v) \models \varphi.\end{aligned}$$

The relation $\mathrm{Sat}_{\Pi,n} \subseteq \mathbb{N} \times \mathbb{N}$ is defined analogously.

Theorem 2.28. *For all $n \geq 1$: $\mathrm{Sat}_{\Sigma,n}$ is Σ_n-definable and $\mathrm{Sat}_{\Pi,n}$ is Π_n-definable.*

We will exhibit formulas $\mathsf{Sat}_{\Sigma,n}$ and $\mathsf{Sat}_{\Pi,n}$ which define $\mathrm{Sat}_{\Sigma,n}$ and $\mathrm{Sat}_{\Pi,n}$ respectively. To this aim it suffices to follow the usual inductive definition of the satisfaction relation as outlined above.

Proof. In Lemma 2.7 we have shown that, if $\psi(\overline{x})$ is a Σ_0 formula, then χ_ψ is primitive recursive by an induction on the logical complexity of $\psi(\overline{x})$. By repeating this inductive proof as definition of a recursive algorithm we obtain a primitive recursive function

$$f : \mathbb{N} \to \mathbb{N}, k \mapsto \begin{cases} e & \text{if } k = \#\psi(\overline{x}),\ \psi \text{ is a } \Delta_0 \text{ formula},\ \varphi_e = \chi_\psi \\ 0 & \text{otherwise} \end{cases}$$

where φ_e is primitive recursive. Now,

$$\begin{aligned}(k, l) \in \mathrm{Sat}_{\Sigma,0} \text{ iff } & k = \#\psi(x_1, \ldots, x_n) \text{ for a } \Sigma_0 \text{ formula } \psi,\\ & l = \#v \text{ for a variable evaluation } v = [x_1 \backslash m_1, \ldots, x_n \backslash m_n] \text{ for } m_1, \ldots, m_n \in \mathbb{N}, \text{ and}\\ & (\mathbb{N}, v) \models \psi(x_1, \ldots, x_n).\end{aligned}$$

and $(\mathbb{N}, v) \models \psi(x_1, \ldots, x_n)$ iff $U_n(f(k), m_1, \ldots, m_n) = 1$ where U_n is the n-argument universal function. Therefore $\mathrm{Sat}_{\Sigma,0} = \mathrm{Sat}_{\Pi,0}$ is decidable and hence Δ_1-definable.

We proceed by induction on n and observe that

$$\begin{aligned}(k,l)\in \mathrm{Sat}_{\Pi,n+1} \text{ iff } & k=\#\forall x\,\varphi \text{ for some } \Sigma_n \text{ formula } \varphi,\\ & l=\#v \text{ for some variable evaluation } v \text{ for } \mathbb{N} \text{ and } \forall x\,\varphi, \text{ and}\\ & \text{for all } m\text{: if } l'=l[x\backslash m], \text{ then } (\#\varphi,l')\in \mathrm{Sat}_{\Sigma,n}.\end{aligned}$$

This can be written as the formula

$$\begin{aligned}\chi(k,l)\equiv \exists u\leq k\exists k'\leq k\,\big(&\mathsf{UniQ}(u,k',k)\wedge \Sigma_n\mathsf{Formula}(k')\wedge \mathsf{VarEvalFor}(l,k)\\ &\forall m\forall l'\,(\mathsf{VarEvalAdd}(l,u,m,l')\to \mathsf{Sat}_{\Sigma.n}(k',l'))\big)\end{aligned}$$

where $\mathsf{UniQ}(u,v',v)$ iff $u=\#x$ for some variable x, $v'=\#\varphi$ for some formula φ and $v=\#\forall x\,\varphi$, etc. All of these predicates are decidable and hence Δ_1-definable. Therefore, since $\mathsf{Sat}_{\Sigma,n}$ is a Σ_n formula by induction hypothesis, $\chi(k,l)$ is equivalent to a Π_{n+1} formula which we call $\mathsf{Sat}_{\Pi,n+1}(k,l)$. The proof for $\mathsf{Sat}_{\Sigma,n+1}$ is analogous. $\square$

Corollary 2.29. *For all $n\geq 1$: the set of true Σ_n sentences is Σ_n-definable and the set of true Π_n sentences is Π_n-definable.*

Proof. σ is a true Σ_n sentence iff $(\#\sigma,\#\emptyset)\in \mathrm{Sat}_{\Sigma,n}$ where $\emptyset$ denotes the empty variable evaluation. Therefore $\mathsf{Tr}_{\Sigma,n}(x)\equiv \mathsf{Sat}_{\Sigma,n}(x,\ulcorner\emptyset\urcorner)$ is a Σ_n definition of the set of true Σ_n sentences. The proof for Π_n sentences is analogous. $\square$

On the other hand, the set of true Σ_n sentences is not Π_n-definable. In order to show this it suffices to repeat the proof of Tarski's result of the undefiablity of truth on every level of the arithmetical hierachy.

Theorem 2.30. *Let $n\geq 1$. Then the set of true Σ_n sentences is not Π_n-definable.*

Proof. Suppose that there is a Π_n formula $\varphi(x)$ s.t. $\mathbb{N}\models\varphi(x)\leftrightarrow \mathsf{Tr}_{\Sigma,n}(x)$. Then $\neg\varphi(x)$ is equivalent to a Σ_n formula. So, by the fixed point lemma, there is a Σ_n sentence σ s.t. $\mathbb{N}\models\neg\varphi(\ulcorner\sigma\urcorner)\leftrightarrow\sigma$. But then $\mathbb{N}\models\neg\mathsf{Tr}_{\Sigma,n}(\ulcorner\sigma\urcorner)\leftrightarrow\sigma$ and, since $\mathsf{Tr}_{\Sigma,n}(x)$ is a definition of the true Σ_n sentences we have $\mathbb{N}\models \mathsf{Tr}_{\Sigma,n}(\ulcorner\sigma\urcorner)\leftrightarrow\sigma$, and thus $\mathbb{N}\models\neg\mathsf{Tr}_{\Sigma,n}(\ulcorner\sigma\urcorner)\leftrightarrow\mathsf{Tr}_{\Sigma,n}(\ulcorner\sigma\urcorner)$, contradiction. $\square$

Corollary 2.31. *The arithmetical hierachy is strict.*

Proof. Let $n\geq 1$. Theorems 2.28 and 2.30 show that there is an $A_n\in\Sigma_n\setminus\Pi_n$. Let $B_n=\mathbb{N}\setminus A_n$. Then $B_n\in\Pi_n\setminus\Sigma_n$. Since $\Sigma_{n-1}\subseteq\Pi_n$ we have $A_n\in\Sigma_n\setminus\Sigma_{n-1}$ and, symmetrically, $B_n\in\Pi_n\setminus\Pi_{n-1}$. The sets Δ_m are closed under complement, i.e., $\{\mathbb{N}\setminus X\mid X\in\Delta_m\}=\Delta_m$ but Σ_n and Π_n are not, hence $\Sigma_n\neq\Delta_m$ and $\Pi_n\neq\Delta_m$ for all $m\geq 0$. Moreover, $A_n\in\Sigma_n\subseteq\Delta_{n+1}$ but $A_n\notin\Delta_n\subseteq\Pi_n$ so $\Delta_n\subset\Delta_{n+1}$. Finally $\Delta_0\subset\Delta_1$ because there is a decidable set which is not Δ_0-definable. $\square$

Chapter 3

Arithmetical theories

3.1 Theories

We start this chapter by recalling some standard notions about first-order logic. A *sentence* is a formula without free variables. A *theory* is a set of sentences. The sentences which comprise a theory are considered as the axioms of that theory. For a theory T and a formula φ we write $T \vdash \varphi$ if φ is provable from T and $T \models \varphi$ if φ is true in all models of T. We are working in first-order logic with equality, i.e., $=$ is considered a logical symbol and we assume that every theory T contains the axioms $\forall x\, x = x$, $\forall x \forall y\, (x = y \to y = x)$, $\forall x \forall y \forall z\, (x = y \to y = z \to x = z)$, as well as $\forall \overline{x} \forall \overline{y}\, (\bigwedge_{i=1}^{k} x_i = y_i \to f(\overline{x}) = f(\overline{y}))$ for every n-ary function symbol f in the language of T and $\forall \overline{x} \forall \overline{y}\, (\bigwedge_{i=1}^{k} x_i = y_i \to R(\overline{x}) \to R(\overline{y}))$ for every n-ary relation symbol R in the language of T. These axioms for equality will henceforth not be mentioned explicitly when defining a theory. We assume familiarity with proofs and models in first-order logic as well as knowledge of the following two results and their proofs.

Theorem 3.1 (Soundness). *If $T \vdash \varphi$, then $T \models \varphi$.*

Theorem 3.2 (Completeness). *If $T \models \varphi$, then $T \vdash \varphi$.*

Definition 3.3. Let T be a theory. T is called *complete* if for every sentence σ: $T \vdash \sigma$ or $T \vdash \neg\sigma$. T is called *consistent* if there is no sentence σ s.t. $T \vdash \sigma$ and $T \vdash \neg\sigma$.

If a theory T is inconsistent, then it proves every sentence: Assume $T \vdash \sigma$ and $T \vdash \neg\sigma$ and let τ be an arbitrary sentence, then, since $\sigma \to \neg\sigma \to \tau$ is a tautology, $T \vdash \tau$. Since an inconsistent T proves every sentence, also $T \vdash \bot$. In the other direction, if $T \vdash \bot$, then T proves every sentence (ex falso quodlibet), so T is inconsistent. Therefore T is inconsistent iff $T \vdash \bot$.

A theory T is consistent and complete iff for every sentence σ, T proves exactly one of σ and $\neg\sigma$. For a structure $\mathcal{M}$ we write $\mathrm{Th}(\mathcal{M}) = \{\sigma \text{ sentence} \mid \mathcal{M} \models \sigma\}$. A theory of the form $\mathrm{Th}(\mathcal{M})$ is consistent and complete since every σ has a uniquely determined truth value which is the negation of the truth value of $\neg\sigma$. On the other hand, if T is a consistent theory, then T has a model, for suppose T would not have a model, then every $\mathcal{M}$ with $\mathcal{M} \models T$ would also make $\mathcal{M} \models \bot$, hence $T \models \bot$ and, by the completeness theorem, $T \vdash \bot$, i.e., T would be inconsistent. If T is both consistent and complete then this $\mathcal{M}$ even makes $\mathrm{Th}(\mathcal{M}) = \{\sigma \text{ sentence} \mid T \vdash \sigma\}$. So we see that the theories that are consistent and complete are exactly the theories of the form $\mathrm{Th}(\mathcal{M})$.

Let T be a theory which is consistent but incomplete, then there is a sentence σ s.t. $T \nvdash \sigma$ and $T \nvdash \neg\sigma$. Then both $T + \sigma$ and $T + \neg\sigma$ are consistent, for assume, say, $T + \sigma$ would be inconsistent, then $T + \sigma \vdash \bot$, so $T \vdash \neg\sigma$ which contradicts the assumption that $T \nvdash \neg\sigma$. For $T + \neg\sigma$ we can proceed analogously. Therefore both, $T + \sigma$ and $T + \neg\sigma$ have models.

Definition 3.4. Let T be a theory in a language L, then a theory T' in a language L' is called *extension of* T if $L' \supseteq L$ and, for every L-formula φ, $T \vdash \varphi$ implies $T' \vdash \varphi$.

Example 3.5. Let $L_M = \{e/0, \circ/2\}$ and let T_M be the following set of axioms (writing $\circ$ in infix notation):

$$\forall x \forall y \forall z\, x \circ (y \circ z) = (x \circ y) \circ z$$
$$\forall x\, (x \circ e = x \wedge e \circ x = x)$$

Then T_M is the theory of monoids. Let $L_G = L_M \cup \{\cdot^{-1}/1\}$ and let $T_G = T_M \cup \{I\}$ where I is the following axiom (writing the unary function symbol $\cdot^{-1}$ as superscript):

$$\forall x\, (x \circ x^{-1} = e \wedge x^{-1} \circ x = e).$$

Then T_G is the theory of groups which is an extension of T_M.

Often we would like to relate two theories which are not as similar. To that aim, theory interpretations are a central tool.

Definition 3.6. Let L, L' be languages, let T be an L theory and let T' be an L' theory. An *interpretation of* L *in* T' is given by:

1. an L' formula $\chi(x)$ s.t. $T' \vdash \exists x\, \chi(x)$

The formula $\chi(x)$ will serve as a definition of the domain of T in T'.

2. for each n-ary predicate symbol P of L an L' formula $\psi_P(x_1, \ldots, x_n)$

3. for each n-ary function symbol f of L an L' formula $\psi_f(x_1, \ldots, x_n, y)$ s.t.

$$T' \vdash \bigwedge_{i=1}^{n} \chi(x_i) \to \exists! y\, (\chi(y) \wedge \psi_f(x_1, \ldots, x_n, y)).$$

An interpretation of L in T' induces a mapping $* : \mathcal{F}(L) \to \mathcal{F}(L')$ as follows: first, for each L term t with free variables $x_1, \ldots, x_n$ we define an L' formula $\psi_t(x_1, \ldots, x_n)$ by induction as follows:

1. If $t = f(t_1, \ldots, t_k)$ with free variables $\overline{x}$, then $\psi_t(\overline{x}, y) \equiv \exists \overline{z}\, (\psi_f(\overline{z}, y) \bigwedge_{i=1}^{k} \psi_{t_i}(\overline{x}, z_i))$.

2. If $t = c$ then $\psi_c(y) \equiv y = c$.

3. If $t = x$ then $\psi_x(x, y) \equiv y = x$.

It is then easy to show by induction on t that

$$T' \vdash \bigwedge_{i=1}^{n} \chi(x_i) \to \exists! y\, (\chi(y) \wedge \psi_t(x_1, \ldots, x_n, y))$$

The translation of a formula with free variables $\overline{x}$ is then defined by

$$\begin{aligned}
(P(t_1,\dots,t_k))^* &\equiv \exists y_1 \cdots \exists y_k \, (\psi_{t_1}(\overline{x}, y_1) \wedge \cdots \wedge \psi_{t_k}(\overline{x}, y_k) \wedge \psi_P(y_1,\dots,y_k)),\\
(t_1 = t_2)^* &\equiv \exists y \, (\psi_{t_1}(\overline{x}, y) \wedge \psi_{t_2}(\overline{x}, y)),\\
(\varphi \to \psi)^* &\equiv \varphi^* \to \psi^*,\\
(\forall x \, \varphi)^* &\equiv \forall x \, (\chi(x) \to \varphi^*),
\end{aligned}$$

and similarly for the other logical symbols. We say that an interpretation of L in T' is an *interpretation of* T *in* T' if $T' \vdash \sigma^*$ for every axiom $\sigma \in T$.

If T' is an extension of T then there is a straightforward interpretation of T in T'.

Lemma 3.7. *Let T be an L theory, let T' be an L' theory, let $\mathcal{I}$ be an interpretation of T in T' and let $*$ be the formula translation induced by $\mathcal{I}$. Then, for every L formula φ, $T \vdash \varphi$ implies $T' \vdash \varphi^*$.*

Proof Sketch. By induction on the length of a T-proof of φ, applying $*$ line by line and showing that $*$ transforms logical axioms into valid formulas of first-order logic, theory axioms into provable sentences, and rule applications into rule applications. $\square$

We say that T' *contains* T if there is an interpretation of T in T'. In general there are different interpretations of T in T'. However, as a notational convention, when we say "T' contains T" we consider this interpretation to be fixed and do not write $*$ explicitly. Where to add $*$ is clear from the context, i.e., the language of the involved formula. So the above lemma can be stated more succinctly as follows: if T' contains T and $T \vdash \varphi$, then $T' \vdash \varphi$.

A theory T is called *arithmetical* if the language of T is the language of arithmetic $L_A = \{0, s, +, \cdot, \leq\}$.

Definition 3.8. The arithmetical theory Q consists of the universal closures of the following formulas:

$$s(x) \neq 0 \tag{Q1}$$
$$s(x) = s(y) \to x = y \tag{Q2}$$
$$x \neq 0 \to \exists y \, x = s(y) \tag{Q3}$$
$$x + 0 = x \tag{Q4}$$
$$x + s(y) = s(x + y) \tag{Q5}$$
$$x \cdot 0 = 0 \tag{Q6}$$
$$x \cdot s(y) = (x \cdot y) + x \tag{Q7}$$
$$x \leq y \leftrightarrow \exists z \, z + x = y \tag{Q8}$$

Example 3.9. Zermelo-Fraenkel set theory ZF is a theory in the language $L' = \{\in/2\}$ which interprets Q as follows:

$$\chi(x) \equiv x \in \omega,$$

where, as usual in set-theoretic notation, ω is the least non-zero limit ordinal,

$$\begin{aligned}
\psi_0(y) &\equiv y = \emptyset,\\
\psi_s(x, y) &\equiv y = x \cup \{x\},
\end{aligned}$$

corresponding to the usual von Neumann definition of the natural numbers in set theory. Addition and multiplication on elements of ω are defined recursively in ZF yielding functions p and t and hence

$$\psi_+(x_1, x_2, y) \equiv y = p(x_1, x_2),$$
$$\psi_\cdot(x_1, x_2, y) \equiv y = t(x_1, x_2).$$

The order is defined by simply translating its defining axiom

$$\psi_\leq(x_1, x_2) \equiv \exists z\,(z \in \omega \wedge p(z, x_1) = x_2).$$

Then it is straightforward to show that $\mathrm{ZF} \vdash \sigma^*$ for all $\sigma \in \{(Q1), \ldots, (Q8)\}$.

3.2 Robinson's minimal arithmetic Q

In this section we will study Robinson's minimal arithmetic Q, which is an important basic arithmetical theory, in more detail.

Definition 3.10. An arithmetical theory T is called sound if $\mathbb{N} \models T$.

Lemma 3.11. *Q is sound, i.e., $\mathbb{N} \models Q$.*

Proof. A quick glance suffices to convince oneself that every axiom of Q is true in $\mathbb{N}$. □

We start by establishing the provability of some simple statements in Q.

Definition 3.12. Let t be a variable-free arithmetical term. We define $\mathrm{val}(t) \in \mathbb{N}$ by induction on t as follows:

$$\mathrm{val}(0) = 0, \quad \mathrm{val}(s(t)) = \mathrm{val}(t) + 1, \quad \mathrm{val}(t + s) = \mathrm{val}(t) + \mathrm{val}(s), \quad \mathrm{val}(t \cdot s) = \mathrm{val}(t) \cdot \mathrm{val}(s).$$

Lemma 3.13.

1. *For all $m, n \in \mathbb{N}$: $Q \vdash \underline{m} + \underline{n} = \underline{m + n}$.*
2. *For all $m, n \in \mathbb{N}$: $Q \vdash \underline{m} \cdot \underline{n} = \underline{m \cdot n}$.*
3. *For all variable-free terms t: $Q \vdash t = \underline{\mathrm{val}(t)}$.*
4. *For all $m, n \in \mathbb{N}$ with $m \neq n$: $Q \vdash \underline{m} \neq \underline{n}$.*
5. *$Q \vdash x + y = 0 \to x = 0 \wedge y = 0$.*
6. *For all $n \in \mathbb{N}$: $Q \vdash s(x) \leq \underline{n + 1} \to x \leq \underline{n}$.*
7. *For all $n \in \mathbb{N}$: $Q \vdash x \leq \underline{n} \leftrightarrow x = 0 \vee x = \underline{1} \vee \cdots \vee x = \underline{n}$.*
8. *For all $m, n \in \mathbb{N}$ with $m \leq n$: $Q \vdash \underline{m} \leq \underline{n}$.*
9. *For all $m, n \in \mathbb{N}$ with $m > n$: $Q \vdash \neg \underline{m} \leq \underline{n}$.*
10. *For all $n \in \mathbb{N}$: $Q \vdash x + \underline{n + 1} = s(x) + \underline{n}$.*
11. *For all $n \in \mathbb{N}$: $Q \vdash x \leq \underline{n} \vee \underline{n + 1} \leq x$.*

Proof. For 1. we proceed by induction[1] on n. For $n = 0$ one application of (Q4) suffices. For the induction step, assume we already have $Q \vdash \underline{m}+\underline{n} = \underline{m+n}$ and work in Q: $\underline{m}+s(\underline{n}) =^{(Q5)} s(\underline{m}+\underline{n}) =^{(IH)} s(\underline{m+n}) = \underline{m+n+1}$.

For 2. we proceed by induction on n. If $n = 0$, one application of (Q6) suffices. For the induction step, assume we already have $Q \vdash \underline{m} \cdot \underline{n} = \underline{m \cdot n}$ and work in Q: $\underline{m} \cdot s(\underline{n}) =^{(Q7)} \underline{m} \cdot \underline{n} + \underline{m} =^{(IH)} \underline{m \cdot n} + \underline{m} =^{1.} \underline{m(n+1)}$.

3. follows immediately from 1. and 2. by induction on the structure of t.

For 4. let $m, n \in \mathbb{N}$ with $m \neq n$. Let $m > n$ and proceed by induction on n. If $n = 0$, then we are done by a single application of (Q1). In the induction step we have $m > n > 0$ and hence there are $m', n' \in \mathbb{N}$ s.t. $n = n'+1$, $m = m'+1$, and thus $m' > n'$. So, by induction hypothesis, $Q \vdash \underline{m'} \neq \underline{n'}$ and the contraposition of (Q2) yields $Q \vdash \underline{m} \neq \underline{n}$. If $n > m$ we have $Q \vdash \underline{n} \neq \underline{m}$ by the above and obtain $Q \vdash \underline{m} \neq \underline{n}$ from symmetry of equality in Q.

For 5. work in Q: if $y \neq 0$, then, by (Q3), there is a z s.t. $y = s(z)$. Thus $x+y = x+s(z) =^{(Q5)} s(x+z) \neq^{(Q1)} 0$. If $y = 0 \wedge x \neq 0$, then $x + y =^{(Q4)} x \neq 0$.

For 6. work in Q: if $s(x) \leq \underline{n+1}$, then, by (Q8), there is a z s.t. $z + s(x) = \underline{n+1}$, so, by (Q5), $s(z+x) = s(\underline{n})$, hence, by (Q2), $z + x = \underline{n}$ and thus, again by (Q8), $x \leq \underline{n}$.

For 7. we proceed by induction on n. For $n = 0$ work in Q: if $x \leq 0$ then, by (Q8), there is a z s.t. $z + x = 0$ and then, by 5., $x = z = 0$. If $x = 0$, then $x \leq 0$ by (Q8) and (Q4). For the induction step work in Q, assuming $x \leq \underline{n} \leftrightarrow x = 0 \vee \cdots x = \underline{n}$. For the left-to-right direction assume $y \leq \underline{n+1}$. If $y = 0$, we are done. If $y \neq 0$, then, by (Q3), there is an x s.t. $y = s(x)$, so $s(x) \leq \underline{n+1}$, so by 6., $x \leq \underline{n}$. Thus $x = 0 \vee \cdots \vee x = \underline{n}$ and therefore $y = \underline{1} \vee \cdots \vee y = \underline{n+1}$. For the right-to-left direction assume $y = 0 \vee \cdots \vee y = \underline{n+1}$ and make a case distinction on the value of y: for $y = \underline{i}$ we have $\underline{n-i+1} + y = \underline{n+1}$ by (Q5) and (Q4) so, by (Q8), $y \leq \underline{n+1}$.

For 8., let $m \leq n$. Then, by 7., $Q \vdash \underline{m} \leq \underline{n} \leftrightarrow \underline{m} = 0 \vee \cdots \vee \underline{m} = \underline{n}$ and, since $m \leq n$, the equation $\underline{m} = \underline{m}$ is among these cases.

For 9., let $m > n$. Then, by 7., $Q \vdash \neg \underline{m} \leq \underline{n} \leftrightarrow \underline{m} \neq 0 \wedge \cdots \wedge \underline{m} \neq \underline{n}$, and, since $m > n$, Q proves all conjuncts by 4..

For 10. we proceed by induction on n. If $n = 0$ work in Q: $x + s(0) = s(x+0) = s(x) = s(x)+0$. For the induction step work in Q: $x + \underline{n+2} = s(x + \underline{n+1}) =^{IH} s(s(x)+\underline{n}) = s(x) + \underline{n+1}$.

For 11., because of 7., it suffices to show that $Q \vdash x = 0 \vee \cdots \vee x = \underline{m-1} \vee \underline{m} \leq x$ by induction on m. If $m = 0$, then $Q \vdash 0 \leq x$ because $Q \vdash \exists z\, z + 0 = x$. If $m > 0$, we have $Q \vdash x = 0 \vee \cdots \vee x = \underline{m-2} \vee \underline{m-1} \leq x$ by induction hypothesis. Work in Q: if $x = 0$ we are done, …, if $x = \underline{m-2}$ we are done. If $\underline{m-1} \leq x$, then there is z s.t. $z + \underline{m-1} = x$. Make a case distinction on z by (Q3): if $z = 0$, then, by 1., $x = \underline{m-1}$ and we are done. If there is z' s.t. $z = s(z')$, then, by 10., $z' + \underline{m} = x$, i.e., $\underline{m} \leq x$ and we are done. □

However, there are also many simple true sentences which Q does not prove, for example the commutativity of addition. The standard method for showing non-provability is to construct a (counter-)model. We will therefore first clarify some basic facts about models of Q. To that aim let $\mathcal{M} \models Q$ and consider the mapping $f : \mathbb{N} \to \mathcal{M}, n \mapsto \underline{n}^{\mathcal{M}}$. We claim that f is an embedding, i.e., an injective homomorphism (w.r.t. L_A). For injectivity, let $m, n \in \mathbb{N}$ with $m \neq n$, then, by Lemma 3.13/4., $Q \vdash \underline{m} \neq \underline{n}$, so $\mathcal{M} \models \underline{m} \neq \underline{n}$, i.e., $\underline{m}^{\mathcal{M}} \neq \underline{n}^{\mathcal{M}}$, thus $f(m) \neq f(n)$. We now show that f is a homomorphism w.r.t. L_A. First we have $f(0) = \underline{0}^{\mathcal{M}} = 0^{\mathcal{M}}$ and $f(n+1) = \underline{n+1}^{\mathcal{M}} = s^{\mathcal{M}}(\underline{n}^{\mathcal{M}}) = s^{\mathcal{M}}(f(n))$, so f is a homomorphism w.r.t. zero and successor. For addition, observe that $f(m+n) = \underline{m+n}^{\mathcal{M}} =^{\text{Lem. 3.13/1.}} \underline{m}^{\mathcal{M}} +^{\mathcal{M}} \underline{n}^{\mathcal{M}} = f(m) +^{\mathcal{M}} f(n)$. We proceed analogously

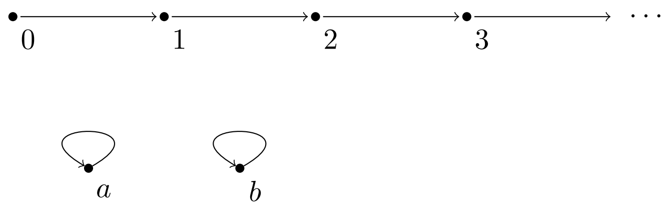
[1]Note that this is an induction on the *meta-level*. Q does not have an induction axiom.

for multiplication where $f(m \cdot n) = \underline{m \cdot n}^{\mathcal{M}} =^{\text{Lem. 3.13/2.}} \underline{m}^{\mathcal{M}} \cdot^{\mathcal{M}} \underline{n}^{\mathcal{M}} = f(m) \cdot^{\mathcal{M}} f(n)$. For the order, let $m \leq n$ and observe that, by Lemma 3.13/8., $Q \vdash \underline{m} \leq \underline{n}$ and therefore $f(m) \leq^{\mathcal{M}} f(n)$.

So every model $\mathcal{M}$ of Q contains a countably infinite subset, $\mathrm{rng}(f)$, which is isomorphic to $\mathbb{N}$. These elements of $\mathcal{M}$ are called *standard numbers*. For the sake of notational simplicity we will usually identify $\mathrm{rng}(f)$ and $\mathbb{N}$. But $\mathcal{M}$ may contain other elements in addition, these are called *nonstandard numbers*.

It is helpful to think of the domain of $\mathcal{M}$ as being partitioned into the connected components of the graph obtained by drawing a directed edge from a to b if $s^{\mathcal{M}}(a) = b$. First we observe that the component which contains the standard numbers contains only the standard numbers: suppose, for the sake of contradiction, that there is a nonstandard a and a standard n s.t. i) $s^{\mathcal{M}}(a) = n$ or ii) $s^{\mathcal{M}}(n) = a$. In case i) $n \neq 0^{\mathcal{M}}$ by (Q1) and then, by (Q2), $a = n - 1$ which cannot be both, standard and nonstandard. In case ii), since n is standard, so is $s^{\mathcal{M}}(n) = a$ which cannot be both, standard and nonstandard. Now, let C be a connected component of $\mathcal{M}$ which is different from $\mathbb{N}$. Then, since $0^{\mathcal{M}} \notin C$, every element of C has a predecessor by (Q3). By (Q2) the predecessor is unique. So every element of C has exactly one successor and exactly one predecessor. The only shapes that satisfy this condition are circles of any finite length or a line which is infinite in both directions.

Every structure different from $\mathbb{N}$ which is a model of Q is called *nonstandard model*. We will now construct a concrete nonstandard model $\mathcal{M}$. The successor graph of $\mathcal{M}$ is:



Since 0 is not a successor, (Q1) is true. Since every element has at most one predecessor, (Q2) is true. Since every nonzero element has a predecessor, (Q3) is true. The table for addition is

$+$	0	1	2	$\cdots$	a	b
0					b	a
1					b	a
2					b	a
$\vdots$					$\vdots$	$\vdots$
a	a	a	a	$\cdots$	a	a
b	b	b	b	$\cdots$	b	b

Since $Q \vdash x + \underline{n} = s^n(x)$, the standard area of the last two rows is fixed by the successor graph. By (Q5) and the definition of the successor we have $e + a = e + s(a) = s(e + a)$ for all $e \in \mathcal{M}$ and, similarly, $e + b = s(e + b)$ for all $e \in \mathcal{M}$. Our choice for the last two columns satisfies this condition, so $\mathcal{M}$ satisfies (Q4) and (Q5). The table for multiplication is

$\cdot$	0	1	2	$\cdots$	a	b
0					b	a
1					b	a
2					b	a
$\vdots$					$\vdots$	$\vdots$
a	0	b	b	$\cdots$	a	a
b	0	a	a	$\cdots$	b	b

Since $Q \vdash \forall x\, x \cdot \underline{n} = (\cdots((0+x)+x)\cdots\cdots+x)$, the standard area of the last two rows is fixed by the table for addition. By (Q7) and the definition of successor we have $e \cdot a = e \cdot s(a) = e \cdot a + e$ for all $e \in \mathcal{M}$ and, similarly, $e \cdot b = e \cdot b + e$. Our choice for the last two columns satisfies this condition, so $\mathcal{M}$ satisfies (Q6) and (Q7). This model shows that $Q \nvdash \forall x \forall y\, x + y = y + x$, $Q \nvdash \forall x\, 0 + x = x$, $Q \nvdash \forall x \forall y\, x \cdot y = y \cdot x$, $Q \nvdash \forall x\, 0 \cdot x = 0$, $Q \nvdash \forall x \forall y\, (x \leq y \wedge y \leq x \to x = y)$, ...

Note that these results show that Q is not complete: Since it is sound it only proves sentences which are true in $\mathbb{N}$. For example $Q \nvdash \neg\forall x \forall y\, x + y = y + x$ On the other hand, as the above model shows, $Q \nvdash \forall x \forall y x + y = y + x$. This shows that the incompleteness of Q is not a deep result. The first incompleteness theorem will be formulated for any consistent theory *containing* Q.

Definition 3.14. An arithmetical theory T is called Σ_1-*complete* if $\mathbb{N} \models \sigma$ implies $T \vdash \sigma$ for every Σ_1 sentence σ.

Lemma 3.15. *Q is Σ_1-complete.*

Proof. Consider a Σ_1 sentence $\exists x\, \varphi(x)$, then $\varphi(x)$ is Δ_0. W.l.o.g. φ is in negation normal form, i.e., φ does not contain implication and negation occurs only immediately above atoms. If $\mathbb{N} \models \exists x\, \varphi(x)$, then there is an $n \in \mathbb{N}$ s.t. $\mathbb{N} \models \varphi(\underline{n})$ and it suffices to show that $Q \vdash \varphi(\underline{n})$.

We show that Q proves every true Δ_0-sentence σ by induction on σ. If σ is an atom, then σ is of the form i) $t = s$ or ii) $t \leq s$. In case i), if $\mathbb{N} \models t = s$, then $\mathrm{val}(t) = \mathrm{val}(s)$, so $\vdash \underline{\mathrm{val}(t)} = \underline{\mathrm{val}(s)}$ and therefore, by Lemma 3.13/3., $Q \vdash t = s$. In case ii), if $\mathbb{N} \models t \leq s$, then $\mathrm{val}(t) \leq \mathrm{val}(s)$, so, by Lemma 3.13/8., $Q \vdash \underline{\mathrm{val}(t)} \leq \underline{\mathrm{val}(s)}$ so, by Lemma 3.13/3., $Q \vdash t \leq s$.

If σ is a negated atom, it is of the form i) $t \neq s$ or ii) $\neg t \leq s$. In case i), if $\mathbb{N} \not\models t = s$, then $\mathrm{val}(t) \neq \mathrm{val}(s)$, so, by Lemma 3.13/4., $Q \vdash \underline{\mathrm{val}(t)} \neq \underline{\mathrm{val}(s)}$ and, by Lemma 3.13/3., $Q \vdash t \neq s$. In case ii), if $\mathbb{N} \not\models t \leq s$, then $\mathrm{val}(t) > \mathrm{val}(s)$, so, by Lemma 3.13/9., $Q \vdash \neg\, \underline{\mathrm{val}(t)} \leq \underline{\mathrm{val}(s)}$, so, by Lemma 3.13/3., $Q \vdash \neg\, t \leq s$.

If $\sigma \equiv \sigma_1 \wedge \sigma_2$, then $\mathbb{N} \models \sigma_1 \wedge \sigma_2$ implies $\mathbb{N} \models \sigma_1$ and $\mathbb{N} \models \sigma_2$, so, by induction hypothesis, $Q \vdash \sigma_1$ and $Q \vdash \sigma_2$ and hence $Q \vdash \sigma_1 \wedge \sigma_2$. If $\sigma \equiv \sigma_1 \vee \sigma_2$, we proceed analogously.

If $\sigma \equiv \exists x \leq t\, \psi(x)$ and $\mathbb{N} \models \sigma$, then there is $n \in \mathbb{N}$ s.t. $n \leq \mathrm{val}(t)$ and $\mathbb{N} \models \psi(\underline{n})$. Then, by induction hypothesis, $Q \vdash \psi(\underline{n})$ and by Lemma 3.13/8., $Q \vdash \underline{n} \leq \underline{\mathrm{val}(t)}$ so, by Lemma 3.13/3., $Q \vdash \underline{n} \leq t$ and therefore $Q \vdash \exists x \leq t\, \psi(x)$.

If $\sigma \equiv \forall x \leq t\, \psi(x)$ and $\mathbb{N} \models \sigma$, then $\mathbb{N} \models \psi(\underline{n})$ for all $n \leq \mathrm{val}(t)$. Then, by induction hypothesis, for all $n \leq \mathrm{val}(t)$, $Q \vdash \psi(\underline{n})$, so, by Lemma 3.13/7., $Q \vdash x \leq \underline{\mathrm{val}(t)} \to \psi(x)$ and, by Lemma 3.13/3., $Q \vdash x \leq t \to \psi(x)$ and therefore $Q \vdash \forall x \leq t\, \psi(x)$. $\square$

3.3 Coding proofs

We will now start to consider formal proofs. Which calculus we use is not essential for the results we discuss in this lecture (as long as it is sound and complete for first-order logic). It will however be practical to use a calculus whose *definition* is as simple as possible, even at the expense of rendering *actual proofs* in this calculus cumbersome.

Definition 3.16. The *logical axioms* are:

$$\varphi \to (\psi \to \varphi) \tag{L1}$$

$$(\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi)) \tag{L2}$$

$$(\neg\psi \to \neg\varphi) \to (\varphi \to \psi) \tag{L3}$$

$$\forall x\, \varphi \to \varphi[x \backslash t] \quad \text{if } t \text{ does not contain a variable that is bound in } \varphi \tag{L4}$$

The *rules* are:

$$\frac{\varphi \quad \varphi \to \psi}{\psi} \tag{MP}$$

$$\frac{\varphi \to \psi[x\backslash y]}{\varphi \to \forall x\, \psi} \quad \text{if } x \text{ is not free in } \varphi,\ y \text{ not bound in } \psi \tag{G}$$

Definition 3.17. Let T be a theory, let φ be a formula. A *T-proof of φ* is a sequence $\varphi_1, \dots, \varphi_n$ of formulas s.t. $\varphi_n \equiv \varphi$ and for all $i = 1, \dots, n$: 1. φ_i is a logical axiom, 2. φ_i is a T-axiom, 3. φ_i is obtained from φ_j and φ_k with $j, k < i$ by modus ponens, or 4. φ_i is obtained from some φ_j with $j < i$ by the generalisation rule.

Since we already have a Δ_0 encoding of finite sequences and an encoding of formulas, the code $\#\pi$ of the proof $\pi = \varphi_1, \dots, \varphi_n$ can be straightforwardly defined as the code of the sequence $\#\varphi_1, \dots, \#\varphi_n$ of natural numbers.

Definition 3.18. A theory T is called *axiomatisable* if the set of sentences T is decidable. A theory T is called *decidable* if the set of sentences $\{\sigma \mid T \vdash \sigma\}$ is decidable.

Note that in an axiomatisable theory, only the set of axioms is decidable. This does not entail that $\{\sigma \mid T \vdash \sigma\}$ is decidable. However, the following definition shows that $\{\sigma \mid T \vdash \sigma\}$ is r.e.

Definition 3.19. Let T be an axiomatisable theory, then there is a Σ_1 definition $\mathsf{Axiom}_T(x)$ of the set of T-axioms. Let $\mathsf{LAxiom}(x)$ be a Σ_1 definition of the set of logical axioms, let $\mathsf{MPRule}(x, y, z)$ be a Σ_1 definition of the modus ponens rule, and let GRule be a Σ_1 definition of the generalisation rule. Towards the definition of the provability predicate we define $\mathsf{P}_T(x, y) \equiv$

$$\begin{aligned}\exists u \Big(& \mathsf{Seq}(y, u+1) \wedge (y)_u = x \wedge \\ & \forall i \leq u \big(\mathsf{LAxiom}((y)_i) \vee \mathsf{Axiom}_T((y)_i) \vee \\ & \qquad \exists j < i \exists k < i\, \mathsf{MPRule}((y)_j, (y)_k, (y)_i) \vee \\ & \qquad \exists j < i\, \mathsf{GRule}((y)_j, (y)_i)\ \big)\Big).\end{aligned}$$

By Lemma 2.6 the formula $\exists y\, \mathsf{P}_T(x, y)$ is equivalent to a Σ_1 formula $\mathsf{Prov}_T(x) \equiv \exists y\, \mathsf{Proof}_T(x, y)$. $\mathsf{Prov}_T(x)$ is called *provability predicate* of the theory T.

Thus, for every axiomatisable theory T, we have obtained a Σ_1 definition $\mathsf{Prov}_T(x)$ of the set of formulas provable in T. This entails that $\{\sigma \mid T \vdash \sigma\}$ is r.e.

3.4 Representing computation in Q

In Chapter 2 we have seen how to represent computation in $\mathbb{N}$ by Σ_1 formulas. Here it will turn out that theories much weaker than $\mathrm{Th}(\mathbb{N})$ already suffice for that purpose.

Definition 3.20. Let T be an arithmetical theory and $R \subseteq \mathbb{N}^k$. An arithmetical formula $\varphi(x_1, \ldots, x_k)$ *defines R in T* if

$$(n_1, \ldots, n_k) \in R \text{ iff } T \vdash \varphi(\underline{n_1}, \ldots, \underline{n_k})$$

This notion generalises arithmetical definability in the sense that R is arithmetically definable iff R is definable in the theory $\mathrm{Th}(\mathbb{N})$.

Definition 3.21. An arithmetical theory T is called Σ_1-*sound* if, for every Σ_1 sentence σ, $T \vdash \sigma$ implies $\mathbb{N} \models \sigma$.

Note that, since there are false Σ_1 sentences, Σ_1 soundness implies consistency.

In this entire chapter we will only work with formulas in the language of arithmetic. However, the results we will prove extend straightforwardly to theories in other languages via theory interpretations. It will therefore be convenient to continue our slight abuse of notation concerning theory interpretations as follows. If T is a theory that contains an arithmetical theory and $*$ is the formula interpretation induced by this interpretation, we will say that an arithmetical formula $\varphi(x_1, \ldots, x_k)$ defines a relation $R \subseteq \mathbb{N}^k$ in T if $(n_1, \ldots, n_k) \in R$ iff $T \vdash \varphi(\underline{n_1}, \ldots, \underline{n_k})^*$. Similarly we will say that T is Σ_1-sound if $T \vdash \sigma^*$ implies $\mathbb{N} \models \sigma$ for all arithmetical Σ_1 sentences, regardless of the quantifier complexity of σ^*. Analogous conventions apply to other notions and will not be mentioned explicitly anymore.

Lemma 3.22. *Let T be a Σ_1-sound theory containing Q, then every r.e. relation is definable by a Σ_1 formula in T.*

Proof. If $R \subseteq \mathbb{N}^k$ is r.e., then there is a Σ_1 formula $\varphi(x_1, \ldots, x_k)$ s.t. $(n_1, \ldots, n_k) \in R$ iff $\mathbb{N} \models \varphi(\underline{n_1}, \ldots, \underline{n_k})$. Now, if $\mathbb{N} \models \varphi(\underline{n_1}, \ldots, \underline{n_k})$, then, by Σ_1 completeness of Q, $Q \vdash \varphi(\underline{n_1}, \ldots, \underline{n_k})$ and, since T contains Q, $T \vdash \varphi(\underline{n_1}, \ldots, \underline{n_k})$. In the other direction, if $T \vdash \varphi(\underline{n_1}, \ldots, \underline{n_k})$, then by Σ_1 soundness of T, $\mathbb{N} \models \varphi(\underline{n_1}, \ldots, \underline{n_k})$. $\square$

The definability of r.e. relations in Q has a number of consequences, the most immediate of which is the undecidability of Q.

Theorem 3.23. *Q is undecidable.*

Proof. Let $A \subseteq \mathbb{N}$ be r.e. but undecidable, then, by Lemma 3.22, there is a definition $\varphi(x)$ of A in Q, i.e., $n \in A$ iff $Q \vdash \varphi(\underline{n})$ for all $n \in \mathbb{N}$. If Q was decidable, then so would A be, contradiction. $\square$

Corollary 3.24. *Validity in first-order logic is undecidable.*

Proof. Suppose it was decidable, then, since $\vdash Q \to \varphi$ iff $Q \vdash \varphi$, also Q would be decidable, contradiction. $\square$

This corollary is a negative solution to Hilbert's *Entscheidungsproblem* (decision problem) that was posed in 1928: does there exist an algorithm which, given a first-order formula as input, determines whether that formula is valid? The historically first solutions to the decision problem where given by Turing and Church in 1936. In the form of the above corollary it follows straightforwardly from the representability of computation in Q.

Turning back to incompleteness, in Theorem 2.23 we have relied on the existence of an undecidable r.e. set to prove a purely semantic variant of the first incompleteness theorem, the fact that $\mathrm{Th}(\mathbb{N})$ is not r.e. Having made the above observation on the definability of r.e. sets in a Σ_1-sound theory T containing Q gets us a long way towards the (syntactic variant of the) first incompleteness theorem. We can now use an analogous proof for T instead of $\mathrm{Th}(\mathbb{N})$ thus obtaining an intermediate variant. It is syntactic in that is speaks about a theory T and not about $\mathbb{N}$ but it still contains the semantic notion of Σ_1 soundness.

Theorem 3.25. *Let T be an axiomatisable Σ_1-sound theory containing Q, then there is a Σ_1 sentence σ s.t. $T \nvdash \sigma$ and $T \nvdash \neg\sigma$.*

Proof. Let $A \subseteq \mathbb{N}$ be r.e. but undecidable, then, by Lemma 3.22, A is definable by a Σ_1 formula $\varphi(x)$ in T, i.e., $A = \{n \in \mathbb{N} \mid T \vdash \varphi(\underline{n})\}$. Let $B = \{n \in \mathbb{N} \mid T \vdash \neg\varphi(\underline{n})\}$. By consistency of T, $A \cap B = \emptyset$. Since T is axiomatisable, the set $\{\sigma \mid T \vdash \sigma\}$ is r.e. Therefore also B is r.e. (just check whether the value of the recursive enumeration function of $\{\sigma \mid T \vdash \sigma\}$ is of the form $\neg\varphi(\underline{n})$). Now suppose that $A \cup B = \mathbb{N}$, i.e., $B = \mathbb{N} \setminus A$, then, by Lemma 1.37, A would be decidable. Contradiction. So there is $n \in \mathbb{N} \setminus (A \cup B)$. The sentence $\sigma = \varphi(\underline{n})$ has the required properties. $\square$

Corollary 3.26. *Let T be an axiomatisable Σ_1-sound theory containing Q, then there is a true Π_1 sentence τ s.t. $T \nvdash \tau$.*

Proof. By Theorem 3.25 we obtain a Σ_1 sentence σ s.t. $T \nvdash \sigma$ and $T \nvdash \neg\sigma$. Let τ be $\neg\sigma$, then $T \nvdash \tau$ and $T \nvdash \neg\tau$. Suppose $\mathbb{N} \not\models \tau$, i.e., $\mathbb{N} \models \sigma$, then, by Σ_1-completeness of T, $T \vdash \sigma$, contradiction. $\square$

Note that the above theorem and corollary apply to all axiomatisable and Σ_1-sound theories containing Q. The incompleteness of Q itself is straightforward; it follows, e.g., from the model we have constructed in Section 3.2. Also note that the proof of Theorem 3.25 we have given here is not constructive in the sense that it does not exhibit a concrete sentence σ, we merely know that it is of the form $\varphi(\underline{n})$ for some $n \in \mathbb{N}$.

So far in this section we have defined r.e. sets in Q. We now turn our attention to partial recursive functions. The graph Γ_f of a partial function $f : \mathbb{N}^k \hookrightarrow \mathbb{N}$ satisfies the *uniqueness condition*, i.e., for all $\overline{x} \in \mathbb{N}^k$ and $y, y' \in \mathbb{N}$: if $(\overline{x}, y) \in \Gamma_f$ and $(\overline{x}, y') \in \Gamma_f$ then $y = y'$. If $f : \mathbb{N}^k \to \mathbb{N}$ is total then, in addition, Γ_f satisfies the *existence condition*, i.e. for all $\overline{x} \in \mathbb{N}^k$ there is a $y \in \mathbb{N}$ s.t. $(\overline{x}, y) \in \Gamma_f$.

Observe that a partial $f : \mathbb{N}^k \hookrightarrow \mathbb{N}$ is recursive iff Γ_f is r.e. So, Lemma 3.22 entails that, for every partial recursive function $f : \mathbb{N}^k \hookrightarrow \mathbb{N}$, there is a Σ_1 formula $\varphi(\overline{x}, y)$ s.t.

$$Q \vdash \varphi(\underline{n_1}, \dots, \underline{n_k}, \underline{m}) \text{ iff } m = f(n_1, \dots, n_k)$$

for all $(n_1, \dots, n_k) \in \mathrm{dom}(f)$. We will now show that this equivalence can be proved in Q. In order to achieve this stronger statement we use the following, slightly more complex, construction.

Lemma 3.27. *Let $f : \mathbb{N}^k \hookrightarrow \mathbb{N}$ be recursive, then there is a Σ_1 formula $\varphi(x_1, \ldots, x_k, y)$ s.t. for all $(n_1, \ldots, n_k) \in \mathrm{dom}(f)$:*

$$Q \vdash \varphi(\underline{n_1}, \ldots, \underline{n_k}, y) \leftrightarrow y = \underline{f(n_1, \ldots, n_k)}.$$

Proof. Since $f : \mathbb{N}^k \hookrightarrow \mathbb{N}$ is recursive, there is a Σ_1 formula $\psi(x_1, \ldots, x_k, y)$ that defines f. Let $\psi(x_1, \ldots, x_k, y) \equiv \exists z\, \psi_0(x_1, \ldots, x_k, y, z)$, then ψ_0 is a Δ_0-formula and define

$$\begin{aligned}\varphi(x_1, \ldots, x_k, y) &\equiv \exists z\, \varphi_0(x_1, \ldots, x_k, y, z)\\ &\equiv \exists z\, \big(\psi_0(x_1, \ldots, x_k, y, z) \land \forall u \leq y \forall v \leq y\, (u \neq y \to \neg\psi_0(x_1, \ldots, x_k, u, v))\\ &\qquad\qquad \land \forall u \leq z \forall v \leq z\, (u \neq y \to \neg\psi_0(x_1, \ldots, x_k, u, v))\big)\end{aligned}$$

Let $(n_1, \ldots, n_k) \in \mathrm{dom}(f)$, $m = f(n_1, \ldots, n_k)$, and $p \in \mathbb{N}$ s.t. $\mathbb{N} \models \psi_0(\underline{n_1}, \ldots, \underline{n_k}, \underline{m}, \underline{p})$.

For showing that $Q \vdash y = \underline{m} \to \varphi(\underline{n_1}, \ldots, \underline{n_k}, y)$ first observe that, by Σ_1 completeness of Q, $Q \vdash \psi_0(\underline{n_1}, \ldots, \underline{n_k}, \underline{m}, \underline{p})$. Since f is a function and Q is Σ_1-complete, $Q \vdash \neg\psi_0(\underline{n_1}, \ldots, \underline{n_k}, \underline{l}, \underline{q})$ for all $l \neq m$ and all $q \in \mathbb{N}$. Therefore $Q \vdash \varphi_0(\underline{n_1}, \ldots, \underline{n_k}, \underline{m}, \underline{p})$ and thus $Q \vdash y = \underline{m} \to \varphi(\underline{n_1}, \ldots, \underline{n_k}, y)$.

For the other direction, work in Q. Towards a contradiction suppose $y \neq \underline{m}$ and $\varphi(\underline{n_1}, \ldots, \underline{n_k}, y)$, i.e., $\varphi_0(\underline{n_1}, \ldots, \underline{n_k}, y, z)$. Since, by Lemma 3.13/11., we have $\forall x\, (x \leq \underline{\max\{m,p\}} \lor \underline{\max\{m,p\}} \leq x)$, we can make the case distinction in i) $\underline{\max\{m,p\}} \leq y$ or $\underline{\max\{m,p\}} \leq z$ or ii) $y, z \leq \underline{\max\{m,p\}}$. In case i) start from $\varphi_0(\underline{n_1}, \ldots, \underline{n_k}, y, z)$ and let $u = \underline{m}$ and $v = \underline{p}$ to obtain $\neg\psi_0(\underline{n_1}, \ldots, \underline{n_k}, \underline{m}, \underline{p})$ and hence a contradiction. In case ii) start from $\varphi_0(\underline{n_1}, \ldots, \underline{n_k}, \underline{m}, \underline{p})$ and obtain $\neg\psi_0(\underline{n_1}, \ldots, \underline{n_k}, y, z)$ and hence a contradiction. Therefore $Q \vdash \varphi_0(\underline{n_1}, \ldots, \underline{n_k}, y, z) \to y = \underline{m}$, i.e., $Q \vdash \varphi(\underline{n_1}, \ldots, \underline{n_k}, y) \to y = \underline{m}$. □

Note that the above lemma only shows that uniqueness is proven numeralwise. In general,

$$Q \nvdash \forall \overline{x} \forall y \forall y'\, (\varphi(\overline{x}, y) \land \varphi(\overline{x}, y') \to y = y')$$

for a Σ_1 definition $\varphi(\overline{x}, y)$ of a partial recursive function.

Example 3.28. Let $\varphi(x_1, x_2, y) \equiv x_1 = x_2 + y \lor x_2 = x_1 + y$. Then $\varphi(x_1, x_2, y)$ defines the function $(m_1, m_2) \mapsto |m_1 - m_2|$ in $\mathbb{N}$. However, Q does not prove the uniqueness property, because, letting $\mathcal{M}$ be the model considered in Section 3.2, we have $\mathcal{M} \models \varphi(a, a, a)$ because $\mathcal{M} \models a + a = a$ and $\mathcal{M} \models \varphi(a, a, b)$ because $\mathcal{M} \models a + b = a$.

An analogous observation can be made for totality. Letting $\varphi(\overline{x}, y)$ be a Σ_1 definition of a total recursive function f we have $Q \vdash \varphi(\underline{n_1}, \ldots, \underline{n_k}, \underline{f(n_1, \ldots, n_k)})$ for all $n_1, \ldots, n_k \in \mathbb{N}$. On the other hand, in general,

$$Q \nvdash \forall \overline{x} \exists y\, \varphi(\overline{x}, y).$$

It turns out that Q only proves the totality of very few functions. In fact, the question which total recursive functions are provably total in a theory is an important approach to classifying the strength of theories with deep connections to the second incompleteness theorem. Concerning Q we have for example:

Theorem 3.29. *Let $\mathsf{Exp}(x, y)$ be a Σ_1 definition of the total recursive function $x \mapsto 2^x$, then $Q \nvdash \forall x \exists y\, \mathsf{Exp}(x, y)$.*

Without Proof. □

3.5 The first incompleteness theorem

In Section 2.4 we have shown the fixed point lemma for $\mathbb{N}$. A crucial prerequisite for doing so was the definability of partial recursive functions by Σ_1 formulas in $\mathbb{N}$. Now that we have obtained the definability of partial recursive functions by Σ_1 formulas in Q we can, essentially by repeating the same proof, obtain a fixed point lemma for any theory containing Q.

Lemma 3.30 (Fixed point lemma)**.** *Let T be a theory that contains Q and let $\varphi(x)$ be a formula. Then there is a sentence σ s.t. $T \vdash \sigma \leftrightarrow \varphi(\ulcorner\sigma\urcorner)$.*

Proof. Given $\varphi(x)$ we want to define a formula $\psi(x)$ s.t.

$$T \vdash \psi(\ulcorner\chi(x)\urcorner) \leftrightarrow \varphi(\ulcorner\chi(\ulcorner\chi(x)\urcorner)\urcorner) \tag{*}$$

for all formulas $\chi(x)$. Then, applying $\psi(x)$ to itself we obtain

$$T \vdash \psi(\ulcorner\psi(x)\urcorner) \leftrightarrow \varphi(\ulcorner\psi(\ulcorner\psi(x)\urcorner)\urcorner)$$

and, by letting $\sigma \equiv \psi(\ulcorner\psi(x)\urcorner)$, we have

$$T \vdash \sigma \leftrightarrow \varphi(\ulcorner\sigma\urcorner).$$

It remains to define ψ and to show (*). To that aim we define the function $f : \mathbb{N} \to \mathbb{N}$ by

$$n \mapsto \begin{cases} \#\chi(\ulcorner\chi(x)\urcorner) & \text{if } n = \#\chi(x) \text{ for a formula } \chi(x) \\ 0 & \text{otherwise} \end{cases}$$

and note that f is primitive recursive. So, by Lemma 3.27, there is a Σ_1 formula $F(x, y)$ s.t. for all $n \in \mathbb{N}$: $Q \vdash F(\underline{n}, y) \leftrightarrow y = \underline{f(n)}$ and hence $T \vdash F(\underline{n}, y) \leftrightarrow y = \underline{f(n)}$. We define $\psi(x) \equiv \exists y\,(F(x, y) \wedge \varphi(y))$ and obtain

$$\begin{aligned} T \vdash \psi(\ulcorner\chi(x)\urcorner) &\leftrightarrow \exists y\,(F(\ulcorner\chi(x)\urcorner, y) \wedge \varphi(y)) \\ &\leftrightarrow \exists y\,(y = \underline{f(\#\chi(x))} \wedge \varphi(y)) \\ &\leftrightarrow \exists y\,(y = \ulcorner\chi(\ulcorner\chi(x)\urcorner)\urcorner \wedge \varphi(y)) \\ &\leftrightarrow \varphi(\ulcorner\chi(\ulcorner\chi(x)\urcorner)\urcorner). \end{aligned}$$

□

Definition 3.31. Let T be an axiomatisable theory containing Q. By the fixed point lemma there is a sentence G_T satisfying $T \vdash \mathsf{G}_T \leftrightarrow \neg\mathsf{Prov}_T(\ulcorner\mathsf{G}_T\urcorner)$. This sentence is called *Gödel sentence of T*.

The Gödel sentence G_T is a fixed point of the negation of the provability predicate of T. It can be understood as expressing "I am not T-provable" in Q. The Gödel sentence G_T is a sentence in the language of T which is equivalent to the interpretation of a Π_1-sentence, namely $\neg\mathsf{Prov}_T(\ulcorner\mathsf{G}_T\urcorner)$, in T. Whether this is still of the form $\forall x\,\varphi(x)$ for $\varphi(x)$ a translation of a Δ_0 formula depends on the interpretation. We can now give the first constructive variant of the first incompleteness theorem.

Theorem 3.32. *Let T be an axiomatisable theory containing Q. If T is consistent, then $T \nvdash \mathsf{G}_T$. If T is Σ_1-sound, then $T \nvdash \neg\mathsf{G}_T$.*

Proof. If $T \vdash \mathsf{G}_T$, then $T \vdash \neg\mathsf{Prov}_T(\ulcorner \mathsf{G}_T \urcorner)$. On the other hand, then also $\mathbb{N} \models \mathsf{Prov}_T(\ulcorner \mathsf{G}_T \urcorner)$, so, by Σ_1-completeness of Q, $T \vdash \mathsf{Prov}_T(\ulcorner \mathsf{G}_T \urcorner)$ and thus T is inconsistent.

If $T \vdash \neg\mathsf{G}_T$, then $T \vdash \mathsf{Prov}_T(\ulcorner \mathsf{G}_T \urcorner)$ and, by Σ_1-soundness of T, $\mathbb{N} \models \mathsf{Prov}_T(\ulcorner \mathsf{G}_T \urcorner)$, i.e., $T \vdash \mathsf{G}_T$, and thus T is inconsistent. □

From the fixed point lemma it is also straightforward to show a syntactic generalisation of Tarski's theorem on the arithmetical undefinability of arithmetical truth.

Theorem 3.33. *Let T be a consistent theory containing Q. Then there is no formula* $\mathrm{Tr}(x)$ *s.t. for every sentence τ: $T \vdash \tau \leftrightarrow \mathrm{Tr}(\ulcorner \tau \urcorner)$.*

Proof. Suppose there is such a formula $\mathrm{Tr}(x)$, then, by the fixed point lemma, there is a sentence σ s.t. $T \vdash \sigma \leftrightarrow \neg\mathrm{Tr}(\ulcorner \sigma \urcorner)$. Then $T \vdash \mathrm{Tr}(\ulcorner \sigma \urcorner) \leftrightarrow \neg\mathrm{Tr}(\ulcorner \sigma \urcorner)$ and hence T would be inconsistent. □

In other words, T cannot define its own truth.

We will now prove the first incompleteness theorem in its full strength. By replacing the assumption of Σ_1 soundness by that of consistency it will become fully syntactic while remaining constructive in the sense of explicitly proving a sentence which is neither provable nor refutable. It has been shown by Rosser in 1936 as an improvement of Gödel's original result from 1931. Let $\mathrm{neg} : \mathbb{N} \to \mathbb{N}$ be defined by

$$\mathrm{neg}(n) = \begin{cases} \#\neg\varphi & \text{if } n = \#\varphi \text{ for some formula } \varphi \\ 0 & \text{otherwise} \end{cases}$$

Then neg is primitive recursive. Let $\mathsf{Neg}(x, y)$ be a Σ_1 definition of neg and, for an axiomatisable theory T, consider the formula

$$\rho(x) \equiv \exists y\, (\mathsf{Neg}(x, y) \wedge \mathsf{Prov}_T(y)).$$

Then $\rho(x)$ expresses the existence of a T-refutation of the formula with code x and is equivalent to a Σ_1 formula $\exists y\, \mathsf{Ref}_T(x, y)$ where $\mathsf{Ref}_T(x, y)$ is Δ_0.

Definition 3.34. Let T be an axiomatisable theory, then the *Rosser provability predicate* is

$$\mathsf{Prov}^{\mathsf{R}}_T(x) \equiv \exists y\, (\mathsf{Proof}_T(x, y) \wedge \forall z < y\, \neg\mathsf{Ref}_T(x, z)).$$

Since T is axiomatisable, its provability predicate is $\exists y\, \mathsf{Proof}_T(x, y)$ where $\mathsf{Proof}_T(x, y)$ is a Δ_0 formula. Therefore $\mathsf{Prov}^{\mathsf{R}}_T(x)$ is a Σ_1 formula. $\mathsf{Prov}^{\mathsf{R}}_T(x)$ expresses that there is a proof of the formula with code x below which there is no refutation of the formula with code x. Note that, for a consistent theory T, we have $\mathbb{N} \models \mathsf{Prov}_T(x) \leftrightarrow \mathsf{Prov}^{\mathsf{R}}_T(x)$. Moreover, note that $\neg\mathsf{Prov}^{\mathsf{R}}_T(x)$ is logically equivalent to

$$\forall y\, (\neg\mathsf{Proof}_T(x, y) \vee \exists z < y\, \mathsf{Ref}_T(x, z))$$

which can be read as the implication: "for every proof of x there is a smaller refutation of x".

Definition 3.35. Let T be an axiomatisable theory containing Q. By the fixed point lemma there is a sentence R_T satisfying $T \vdash \mathsf{R}_T \leftrightarrow \neg\mathsf{Prov}^{\mathsf{R}}_T(\ulcorner \mathsf{R}_T \urcorner)$. This sentence is called *Rosser sentence of T*.

Theorem 3.36 (First incompleteness theorem)**.** *Let T be a consistent and axiomatisable theory containing Q, then $T \nvdash \mathsf{R}_T$ and $T \nvdash \neg\mathsf{R}_T$.*

Proof. If $T \vdash \mathsf{R}_T$, then $T \vdash \neg\mathsf{Prov}^{\mathsf{R}}_T(\ulcorner \mathsf{R}_T \urcorner)$. On the other hand, there is $p \in \mathbb{N}$ s.t. $\mathbb{N} \models \mathsf{Proof}_T(\ulcorner \mathsf{R}_T \urcorner, \underline{p})$. Moreover, since T is consistent, $T \nvdash \neg\mathsf{R}_T$, so, for all $r \in \mathbb{N}$, $\mathbb{N} \models \neg\mathsf{Ref}_T(\ulcorner \mathsf{R}_T \urcorner, \underline{r})$. By Σ_1-completeness of Q we have $T \vdash \mathsf{Proof}_T(\ulcorner \mathsf{R}_T \urcorner, \underline{p})$ and, for all $r \in \mathbb{N}$, $T \vdash \neg\mathsf{Ref}_T(\ulcorner \mathsf{R}_T \urcorner, \underline{r})$. Hence $T \vdash \forall z \leq \underline{p}\, \neg\mathsf{Ref}_T(\ulcorner \mathsf{R}_T \urcorner, z)$ and so $T \vdash \mathsf{Prov}^{\mathsf{R}}_T(\ulcorner \mathsf{R}_T \urcorner)$, contradiction.

If $T \vdash \neg\mathsf{R}_T$, then $T \vdash \mathsf{Prov}^{\mathsf{R}}_T(\ulcorner \mathsf{R}_T \urcorner)$. On the other hand, there is $r \in \mathbb{N}$ s.t. $\mathbb{N} \models \mathsf{Ref}_T(\ulcorner \mathsf{R}_T \urcorner, \underline{r})$. Moreover, since T is consistent, $T \nvdash \mathsf{R}_T$, so, for all $p \in \mathbb{N}$, $\mathbb{N} \models \neg\mathsf{Proof}_T(\ulcorner \mathsf{R}_T \urcorner, \underline{p})$. By Σ_1-completeness of Q we have $T \vdash \mathsf{Ref}_T(\ulcorner \mathsf{R}_T \urcorner, \underline{r})$ and, for all $p \in \mathbb{N}$, $T \vdash \neg\mathsf{Proof}_T(\ulcorner \mathsf{R}_T \urcorner, \underline{p})$. So we obtain $T \vdash y \leq \underline{r} \to \neg\mathsf{Proof}_T(\ulcorner \mathsf{R}_T \urcorner, y)$ and $T \vdash \underline{r+1} \leq y \to \exists z < y\, \mathsf{Ref}_T(\ulcorner \mathsf{R}_T \urcorner, z)$ and, since $Q \vdash y \leq \underline{r} \vee \underline{r+1} \leq y$, we have

$$T \vdash \forall y \left(\neg\mathsf{Proof}_T(\ulcorner \mathsf{R}_T \urcorner, y) \vee \exists z < y\, \mathsf{Ref}_T(\ulcorner \mathsf{R}_T \urcorner, z)\right)$$

i.e., $T \vdash \neg\mathsf{Prov}^{\mathsf{R}}_T(\ulcorner \mathsf{R}_T \urcorner)$, contradiction. □

A theory is called *essentially undecidable* if all its consistent extensions are undecidable. The following corollary shows that Q is essentially undecidable, even in the slightly stronger sense of interpretability instead of extension.

Corollary 3.37. *Let T be a consistent theory containing Q, then T is undecidable.*

Proof. First observe that, if S is a consistent theory and σ a sentence, then at least one of $S \cup \{\sigma\}$ and $S \cup \{\neg\sigma\}$ is consistent for suppose $S, \sigma \vdash \bot$ and $S, \neg\sigma \vdash \bot$, then $S \vdash \neg\sigma$ and $S \vdash \neg\neg\sigma$, hence S would be inconsistent.

Let L be the language of T and let $\sigma_0, \sigma_1, \ldots$ be a recursive enumeration of the set of L sentences. Let $T_0 = T$, define

$$T_{n+1} = T_n \cup \{\alpha_n\} \quad \text{where} \quad \alpha_n = \begin{cases} \sigma_n & \text{if } T_n \cup \{\sigma_n\} \text{ is consistent} \\ \neg\sigma_n & \text{otherwise} \end{cases}$$

for $n \geq 0$ and $T^* = \bigcup_{n \geq 0} T_n$. By induction on n all T_n are consistent. But then also T^* is consistent, for suppose $T^* \vdash \bot$, then there would be an n s.t. $T_n \vdash \bot$. Moreover, T^* is complete because, for every n, either σ_n or $\neg\sigma_n$ is an axiom of T^*.

Now, for the sake of contradiction, suppose that T is decidable. First note that $T_n \cup \{\sigma_n\}$ is consistent iff $T \nvdash \bigwedge_{i=0}^{n-1} \alpha_i \wedge \sigma_n \to \bot$. So, if T is decidable, then the function $n \mapsto \alpha_n$ is recursive. Therefore $A = \{\alpha_i \mid i \in \mathbb{N}\}$ is decidable: given a sentence σ find an n with $\sigma \equiv \sigma_n$. If $\alpha_n \equiv \sigma_n$ then $\sigma \in A$. If $\alpha_n \equiv \neg\sigma_n$ then $\sigma \notin A$. Therefore, the axiomatisation $\{\sigma \mid T \vdash \sigma\} \cup A$ of T^* is decidable. So T^* would be a consistent, complete, and axiomatisable theory containing Q; this contradicts the first incompleteness theorem. □

3.6 Open induction

A key for proving the second incompleteness theorem is to formalise proofs of several statements whose truth we have relied on for the first incompleteness theorem in a weak arithmetical base theory: in particular we will have to prove equivalences of formulas to Σ_n- and Π_n-formulas and deal with the encoding of sequences. To that aim we now start to consider arithmetical theories with induction axioms.

Definition 3.38. Let $\varphi(x, \overline{z})$ be an arithmetical formula, then the *induction axiom for φ w.r.t. x* is the sentence

$$I_x\varphi \equiv \forall \overline{z} \left(\varphi(0, \overline{z}) \to \forall y \left(\varphi(y, \overline{z}) \to \varphi(s(y), \overline{z})\right) \to \forall x\, \varphi(x, \overline{z})\right).$$

We define the theories

$$\begin{aligned} \mathrm{IOpen} &= Q \cup \{I_x\varphi \mid \varphi \text{ is a quantifier-free arithmetical formula}\}, \\ \mathrm{I}\Sigma_k &= Q \cup \{I_x\varphi \mid \varphi \text{ is a } \Sigma_k\text{-formula}\} \text{ for } k \geq 0, \text{ and} \\ \mathrm{PA} &= Q \cup \{I_x\varphi \mid \varphi \text{ is an arithmetical formula}\}. \end{aligned}$$

Now, for any $k \geq 0$, $\mathrm{I}\Sigma_k$ is an extension of Q, $\mathrm{I}\Sigma_{k+1}$ is an extension of $\mathrm{I}\Sigma_k$ and PA is an extension of $\mathrm{I}\Sigma_k$. All these extensions take place in the same language L_A. A natural and convenient (but not the minimal) choice for a base theory for the second incompleteness theorem is $\mathrm{I}\Sigma_1$. Before formalising various results in $\mathrm{I}\Sigma_1$ we first study the weaker theory IOpen in this section.

Lemma 3.39. IOpen *proves the following formulas*

1. $x + y = y + x$
2. $x + (y + z) = (x + y) + z$
3. $x \cdot y = y \cdot x$
4. $x \cdot (y \cdot z) = (x \cdot y) \cdot z$
5. $x \cdot (y + z) = x \cdot y + x \cdot z$

Proof. For 1 we first show $0+x = x$ (*) in IOpen by induction[2] on x: $0+0 = 0$ follows from (Q4). If $0 + x = x$, then $0 + s(x) =^{(Q5)} s(0 + x) =^{\mathrm{IH}} s(x)$. Next we show $s(x) + y = s(x + y)$ (**) in IOpen by induction on y: for the induction base, we have $s(x)+0 =^{(Q4)} s(x) =^{(Q4)} s(x+0)$. For the induction step, we have $s(x)+s(y) =^{(Q5)} s(s(x)+y) =^{\mathrm{IH}} s(s(x+y)) =^{(Q5)} s(x+s(y))$. Now we show $x + y = y + x$ in IOpen by induction on y. If $y = 0$ we have $x + 0 =^{(Q4)} x =^{(*)} 0 + x$. For the induction step we have $x + s(y) =^{(Q5)} s(x + y) =^{\mathrm{IH}} s(y + x) =^{(**)} s(y) + x$.
2-5: without proof. □

For showing 2-5 in the above lemma one proceeds similarly to the proof of 1: it suffices to formalise natural symbolic proofs up to a level of detail that allows to see that all inductions are quantifier-free and, beyond that, only axioms of Q and lemmas which are provable in IOpen have been used. From this point on, we will generously leave out details of formalised proofs and only mention crucial turning points. Filling out these details can be done as in the above proof of Lemma 3.39/1.

We will now show closure properties of arithmetically definable relations along the lines of Lemma 2.6 in IOpen. This will strengthen the results of Lemma 2.6 by replacing (arithmetical) equivalence by provable equivalence in IOpen. Note that, for any sound theory T, provable equivalence in T implies (arithmetical) equivalence and is implied by logical equivalence. We start by developing our encoding of pairs in IOpen. As usual we write $z = \langle x, y\rangle$ as abbreviation for the L_A-formula $\underline{2} \cdot z = (x + y) \cdot (x + y + \underline{1}) + \underline{2} \cdot y$.

Lemma 3.40. IOpen *proves the following formulas:*

[2] Note that (*) is quantifier-free.

1. $\forall x \forall y \exists! z\, z = \langle x, y \rangle$

2. $\forall z \exists! (x, y)\, z = \langle x, y \rangle$, *i.e.*,
   $\forall z \exists x \exists y (z = \langle x, y \rangle \land \forall x' \forall y' (z = \langle x', y' \rangle \to x' = x \land y' = y))$

3. $\forall x \forall y \forall z\, (z = \langle x, y \rangle \to x \leq z \land y \leq z)$

Without proof. □

Lemma 3.41. *For all $n \geq 0$, the Σ_{n+1} formulas are closed under existential quantification in* IOpen *and the Π_{n+1} formulas are closed under universal quantification in* IOpen.

Proof. We follow the proof of Lemma 2.6/3 and 4: we proceed by induction on n. Let $\exists z\, \varphi(\overline{x}, y, z)$ be Σ_{n+1}, i.e., $\varphi(\overline{x}, y, z)$ is Π_n. Define

$$\psi(\overline{x}) \equiv \exists u \forall y \forall z\, (u = \langle y, z \rangle \to \varphi(\overline{x}, y, z)) \text{ and}$$
$$\psi_{\mathrm{b}}(\overline{x}) \equiv \exists u \forall y \leq u \forall z \leq u\, (u = \langle y, z \rangle \to \varphi(\overline{x}, y, z)).$$

Now we claim that IOpen $\vdash \exists y \exists z\, \varphi(\overline{x}, y, z) \leftrightarrow \psi(\overline{x})$. Work in IOpen: for the left-to-right direction assume $\varphi(\overline{x}, y, z)$, then, by Lemma 3.40, there is a u s.t. $u = \langle y, z \rangle$ and, moreover, $\forall y' \forall z'\, (u = \langle y', z' \rangle \to y' = y \land z' = z)$. Since $\varphi(\overline{x}, y, z)$ and $u = \langle y, z \rangle$, we obtain $\forall y' \forall z'\, (u = \langle y', z' \rangle \to \varphi(\overline{x}, y', z'))$. For the right-to-left direction let u be s.t. $\forall y \forall z\, (u = \langle y, z \rangle \to \varphi(\overline{x}, y, z))$, then, by Lemma 3.40, there are y, z s.t. $u = \langle y, z \rangle$ and thus $\varphi(\overline{x}, y, z)$. Furthermore we also have IOpen $\vdash \exists y \exists z\, \varphi_{\mathrm{b}}(\overline{x}, y, z) \leftrightarrow \psi(\overline{x})$ because, by Lemma 3.40, IOpen $\vdash \psi(\overline{x}) \leftrightarrow \psi_{\mathrm{b}}(\overline{x})$. The rest of this proof is exactly as in that of Lemma 2.6/3 and 4. □

Lemma 3.42. *For all $n \geq 0$: the Σ_n-, Π_n-, and Δ_n-definable relations are closed under union and intersection in* IOpen.

Proof. The statement is trivial for $n = 0$, so let $n > 0$. If $\exists y\, \varphi(\overline{x}, y)$ and $\exists z\, \psi(\overline{x}, z)$ are Σ_n formulas, then $\exists y\, \varphi(\overline{x}, y) \land \exists z\, \psi(\overline{x}, z)$ is logically equivalent to $\exists y \exists z\, (\varphi(\overline{x}, y) \land \psi(\overline{x}, z))$ which is equivalent to a Σ_n formula in IOpen by Lemma 3.41. Similarly, $\exists y\, \varphi(\overline{x}, y) \lor \exists z\, \psi(\overline{x}, z)$ is logically equivalent to $\exists y\, (\varphi(\overline{x}, y) \lor \psi(\overline{x}, y))$ which is a Σ_n formula too. The cases for Π_n are analogous. The cases for Δ_n follow from those of Σ_n and Π_n. □

So while IOpen, by virtue of its induction axioms, considerably extends Q, it still has quite narrow limits. Several closure properties of arithmetically definable relations require stronger induction axioms, see Section 3.7. In terms of concrete mathematical statements, IOpen does, for example, not prove the irrationality of $\sqrt{2}$, which is expressed by the arithmetical sentence $\neg \exists x \exists y\, (y \neq 0 \land 2y^2 = x^2)$.

Remark 3.43. It is still open whether the following problem, which is related to the MRDP theorem, is decidable: given a Diophantine equation $p(\overline{x}) = q(\overline{x})$ for $p(\overline{x}), q(\overline{x}) \in \mathbb{N}[\overline{x}]$, does there exist an $\mathcal{M} \models$ IOpen with $\mathcal{M} \models \exists \overline{x}\, p(\overline{x}) = q(\overline{x})$. The corresponding problem for Q has been shown to be decidable. For theories slightly stronger that IOpen it is known to be undecidable.

3.7 Σ_1 induction

In this section we study $\mathrm{I}\Sigma_1$, minimal arithmetic Q together with induction for Σ_1 formulas. We start by establishing the provable closure of Σ_1-definable sets under bounded quantifiers.

Definition 3.44. Let $\varphi(x, y, \overline{z})$ be an arithmetical formula. Then we define the *collection axiom for* φ as

$$B\varphi \equiv \forall\overline{z}\forall u \left(\forall x \leq u \exists y\, \varphi(x, y, \overline{z}) \to \exists v \forall x \leq u \exists y \leq v\, \varphi(x, y, \overline{z})\right)$$

In preparation of the next proof, note that, if $\varphi(x)$ is equivalent in $\mathrm{I}\Sigma_1$ to a Σ_1 formula, then $\mathrm{I}\Sigma_1 \vdash I_x\varphi(x)$.

Lemma 3.45. *Let* $\varphi(x, y, \overline{z})$ *be a* Δ_0 *formula, then* $\mathrm{I}\Sigma_1 \vdash B\varphi$.

Proof. We work in $\mathrm{I}\Sigma_1$: given $\overline{z}$ and u assume that $\forall x \leq u \exists y\, \varphi(x, y, \overline{z})$. We show

$$u' \leq u \to \exists v \forall x \leq u' \exists y \leq v\, \varphi(x, y, \overline{z}) \qquad (*)$$

by induction[3] on u'. If $u' = 0$, let y_0 be s.t. $\varphi(0, y_0, \overline{z})$ and set $v = y_0$. Then $\forall x \leq 0 \exists y \leq v\, \varphi(x, y, \overline{z})$. For the induction step make a case distinction: if $s(u') > u$ then we are done. So let $s(u') \leq u$ and assume (*) for u', so there is a v s.t. $\forall x \leq u' \exists y \leq v\, \varphi(x, y, \overline{z})$. Let y_0 be s.t. $\varphi(u' + 1, y_0, \overline{z})$ and set $w = \max\{v, y_0\}$, then $\forall x \leq s(u') \exists y \leq w\, \varphi(x, y, \overline{z})$. So we have (*) for all u', in particular for $u' = u$ which is what we wanted to show. □

Lemma 3.46. $\mathrm{I}\Sigma_1$ *proves collection for* Σ_1 *formulas.*

Proof. Let $\varphi(x, y_1, \overline{z})$ be a Σ_1 formula, let $\varphi(x, y_1, \overline{z}) \equiv \exists y_2\, \psi(x, y_1, y_2, \overline{z})$, then $\psi(x, y_1, y_2, \overline{z})$ is a Σ_0 formula. Work in $\mathrm{I}\Sigma_1$: given $\overline{z}$ and u, assume $\forall x \leq u \exists y_1 \exists y_2\, \psi(x, y_1, y_2, \overline{z})$. Then, as in the proof of Lemma 3.41, we have

$$\forall x \leq u \exists y \forall y_1 \leq y \forall y_2 \leq y\, (y = \langle y_1, y_2\rangle \wedge \psi(x, y_1, y_2, \overline{z}))$$

and thus, by Σ_0 collection, we have

$$\exists v \forall x \leq u \exists y \leq v \forall y_1 \leq y \forall y_2 \leq y\, (y = \langle y_1, y_2\rangle \wedge \psi(x, y_1, y_2, \overline{z})),$$

so, by Lemma 3.40,

$$\exists v \forall x \leq u \exists y_1 \leq v \exists y_2 \leq v\, \psi(x, y_1, y_2, \overline{z}).$$

□

Lemma 3.47. Σ_1 *formulas are closed under bounded quantification in* $\mathrm{I}\Sigma_1$.

Proof. Let $\exists y\, \varphi(x, y, \overline{z})$ be a Σ_1 formula, then $\exists x \leq t \exists y\, \varphi(x, y, \overline{z})$ is logically equivalent to $\exists y\, \exists x \leq t\, \varphi(x, y, \overline{z})$. Moreover, by Σ_0 collection, $\forall x \leq t \exists y\, \varphi(x, y, \overline{z})$ is equivalent in $\mathrm{I}\Sigma_1$ to $\exists v \forall x \leq t \exists y \leq v\, \varphi(x, y, \overline{z})$ which is a Σ_1 formula. □

Now we can come back to the provability predicate. In Section 3.3 we have defined the formula $\mathsf{P}_T(x, y) \equiv$

$$\begin{aligned}\exists u \Big(&\mathsf{Seq}(y, u) \wedge x = (y)_{u-1} \wedge \\ &\forall i < u \big(\ \mathsf{LAxiom}((y)_i) \vee \mathsf{Axiom}_T((y)_i) \vee \\ &\qquad \exists j < i \exists k < i\, \mathsf{MPRule}((y)_j, (y)_k, (y)_i) \vee \\ &\qquad \exists j < i\, \mathsf{GRule}((y)_j, (y)_i)\ \big)\Big).\end{aligned}$$

[3] Note that (*) is equivalent to a Σ_1 formula in $\mathrm{I}\Sigma_1$.

and have observed that $\exists y\, \mathsf{P}_T(x,y)$ is (arithmetically) equivalent to a Σ_1 formula $\mathsf{Prov}_T(x) \equiv \exists y\, \mathsf{Proof}_T(x,y)$. By Lemmas 3.41, 3.42, and 3.47 we even have $\mathrm{I}\Sigma_1 \vdash \exists y\, \mathsf{P}_T(x,y) \leftrightarrow \exists y\, \mathsf{Proof}_T(x,y)$. This equivalence is a crucial prerequisite for proving properties of $\mathsf{Prov}_T(x)$ in $\mathrm{I}\Sigma_1$.

$\mathrm{I}\Sigma_1$ also allows to work with finite sequences in a comfortable way. For our encoding of finite sequences consisting of the Δ_0 formulas $\mathsf{Seq}(w,v)$ and $(w)_u = x$ we obtain:

Lemma 3.48. $\mathrm{I}\Sigma_1$ *proves the sequence axioms*

$$\mathsf{Seq}(w,v) \to \forall u < v \exists! x\, (w)_u = x \tag{S1}$$
$$\exists w\, \mathsf{Seq}(w,0) \tag{S2}$$
$$\mathsf{Seq}(w,v) \to \forall x \exists w' \,(\mathsf{Seq}(w', s(v)) \wedge \forall u < v \forall y\, ((w')_u = y \leftrightarrow (w)_u = y) \wedge (w')_v = x) \tag{S3}$$

as well as:

1. $\mathsf{Seq}(x,u) \wedge \mathsf{Seq}(y,v) \to \exists z\, (\mathsf{Seq}(z, u+v) \wedge \forall i < u\, (z)_i = (x)_i \wedge \forall i < v\, (z)_{u+i} = (y)_i)$
2. $\mathsf{Seq}(w, s(u)) \to \exists w' (\mathsf{Seq}(w', u) \wedge \forall i < s(u) \exists x\, ((w')_i = x \wedge (w)_i = x))$

Without Proof. □

3.8 The derivability conditions

Definition 3.49. Let T be an axiomatisable theory. The *derivability conditions for* Prov_T are:

$$\text{If } T \vdash \sigma \text{ then } \mathrm{I}\Sigma_1 \vdash \mathsf{Prov}_T(\ulcorner \sigma \urcorner) \tag{D1}$$
$$\mathrm{I}\Sigma_1 \vdash \mathsf{Prov}_T(\ulcorner \sigma \urcorner) \to \mathsf{Prov}_T(\ulcorner \mathsf{Prov}_T(\ulcorner \sigma \urcorner) \urcorner) \tag{D2}$$
$$\mathrm{I}\Sigma_1 \vdash \mathsf{Prov}_T(\ulcorner \sigma \urcorner) \wedge \mathsf{Prov}_T(\ulcorner \sigma \to \tau \urcorner) \to \mathsf{Prov}_T(\ulcorner \tau \urcorner) \tag{D3}$$

for all sentences σ and τ.

The provability conditions (D1), (D2), (D3) are the key properties required for a provability predicate for the second incompleteness theorem to hold. We will first establish them for axiomatisable theories T which contain Q and then prove the second incompleteness theorem from them.

Lemma 3.50 (D1)**.** *Let T be an axiomatisable theory and σ a sentence. If $T \vdash \sigma$, then* $\mathrm{I}\Sigma_1 \vdash \mathsf{Prov}_T(\ulcorner \sigma \urcorner)$.

Proof. Since T is axiomatisable, $\mathsf{Prov}_T(\ulcorner \sigma \urcorner)$ is a Σ_1 sentence. $T \vdash \sigma$ is equivalent to $\mathbb{N} \models \mathsf{Prov}_T(\ulcorner \sigma \urcorner)$, and so, by Σ_1 completeness of Q, $Q \vdash \mathsf{Prov}_T(\ulcorner \sigma \urcorner)$ and hence $\mathrm{I}\Sigma_1 \vdash \mathsf{Prov}_T(\ulcorner \sigma \urcorner)$. □

Lemma 3.51 (D3)**.** *Let T be an axiomatisable theory, let σ and τ be sentences, then* $\mathrm{I}\Sigma_1 \vdash \mathsf{Prov}_T(\ulcorner \sigma \urcorner) \wedge \mathsf{Prov}_T(\ulcorner \sigma \to \tau \urcorner) \to \mathsf{Prov}_T(\ulcorner \tau \urcorner)$

Proof. Work in $\mathrm{I}\Sigma_1$: assume that there are finite sequences p and q s.t. p is a T-proof of σ and q is a T-proof of $\sigma \to \tau$. By Lemma 3.48/1 there is a finite sequence $r = p; q$. Moreover, by (S3), there is a finite sequence $r' = p; q; \ulcorner \tau \urcorner$. Since p and q are proofs and τ is obtained from modus ponens from the last element of p and the last element of q, r' is a T-proof of τ. □

Proving the derivability condition (D2) requires more work. First, note that, since $\mathsf{Prov}_T(\ulcorner\sigma\urcorner)$ is a Σ_1 sentence it suffices to show $\mathrm{I}\Sigma_1 \vdash \tau \to \mathsf{Prov}_T(\ulcorner\tau\urcorner)$ for every Σ_1 sentence τ. This is the *formalisation of* Σ_1 *completeness of* T *in* Σ_1. Hence we will establish (D2) only for a theory T which contains Q and thus is actually Σ_1-complete. We will proceed by formalising the proof of Lemma 3.15.

In order to do that we will need to speak about codes of formulas with free variables and about codes arising from these by substitution. To that aim, consider a Σ_1 formula $\mathsf{NumC}(x, y)$ which defines the function $n \mapsto \#\underline{n}$. By Lemma 3.27 we can assume $Q \vdash \mathsf{NumC}(\underline{n}, y) \leftrightarrow y = \ulcorner\underline{n}\urcorner$. We have already seen a Σ_1 formula $\mathsf{Subst}(x, y, z, u)$ s.t. for every formula φ, every variable x, and every term t: $Q \vdash \mathsf{Subst}(\ulcorner\varphi\urcorner, \ulcorner x\urcorner, \ulcorner t\urcorner, u) \leftrightarrow u = \ulcorner\varphi[x\backslash t]\urcorner$. This can be generalised to the simultaneous substitution of several variables thus obtaining, for all $k \geq 1$, a Σ_1 formula $\mathsf{Subst}_k(x, y_1, \dots, y_k, z_1, \dots, z_k, u)$ s.t. for every formula φ, pairwise different variables $x_1, \dots, x_k$ and all terms $t_1, \dots, t_k$:

$$Q \vdash \mathsf{Subst}_k(\ulcorner\varphi\urcorner, \ulcorner x_1\urcorner, \dots, \ulcorner x_k\urcorner, \ulcorner t_1\urcorner, \dots, \ulcorner t_k\urcorner, u) \leftrightarrow u = \ulcorner\varphi[x_1\backslash t_1, \dots, x_k\backslash t_k]\urcorner$$

Definition 3.52. For arithmetical formulas $\varphi(x_1, \dots, x_k)$ and $\psi(y)$, we define $\psi(\ulcorner\varphi(\dot{x}_1, \dots, \dot{x}_k)\urcorner)$ as abbreviation for

$$\begin{aligned}\exists y' \exists x'_1 \cdots \exists x'_k \big(&\mathsf{NumC}(x_1, x'_1) \wedge \cdots \wedge \mathsf{NumC}(x_k, x'_k) \wedge \\ &\mathsf{Subst}_k(\ulcorner\varphi(x_1, \dots, x_k)\urcorner, \ulcorner x_1\urcorner, \dots, \ulcorner x_k\urcorner, x'_1, \dots, x'_k, y') \wedge \psi(y')\big)\end{aligned}$$

Note that $\psi(\ulcorner\varphi(\dot{x}_1, \dots, \dot{x}_k)\urcorner)$ is an arithmetical formula with free variables $x_1, \dots, x_k$. Also note that, if $\psi(y)$ is equivalent to a Σ_1 formula in $\mathrm{I}\Sigma_1$, then so it $\psi(\ulcorner\varphi(\dot{x}_1, \dots, \dot{x}_k)\urcorner)$. The formula $\psi(\ulcorner\varphi(\dot{x}_1, \dots, \dot{x}_k)\urcorner)$ allows to substitute terms from the object level (where ψ lives) into the object object level (where φ lives). In particular

$$Q \vdash \psi(\ulcorner\varphi(\dot{x}_1, \dots, \dot{x}_k)\urcorner)[x_1\backslash\underline{n_1}, \dots, x_k\backslash\underline{n_k}] \leftrightarrow \psi(\ulcorner\varphi(\underline{n_1}, \dots, \underline{n_k})\urcorner)$$

for all $n_1, \dots, n_k \in \mathbb{N}$ and

$$\mathrm{I}\Sigma_1 \vdash \psi(\ulcorner\varphi(\dot{x}_1, \dots, \dot{x}_k)\urcorner)[x_1\backslash t_1(\overline{x}), \dots, x_k\backslash t_k(\overline{x})] \leftrightarrow \psi(\ulcorner\varphi(t_1(\dot{x}_1, \dots, \dot{x}_k), \dots, t_k(\dot{x}_1, \dots, \dot{x}_k))\urcorner)$$

for all L_A terms $t_1(x_1, \dots, x_k), \dots, t_k(x_1, \dots, x_k)$.

For formalised Σ_1 completeness of Q we start with Lemma 3.13/1 where we have shown that, for all $m, n \in \mathbb{N}$, $Q \vdash \underline{m} + \underline{n} = \underline{m+n}$. This is formalised as follows:

Lemma 3.53. $\mathrm{I}\Sigma_1 \vdash \forall m \forall n\, \mathsf{Prov}_Q(\ulcorner\dot{m} + \dot{n} = \dot{z}\urcorner)[z\backslash m+n]$

Proof. Note that $\mathsf{Prov}_Q(\ulcorner\dot{m} + \dot{n} = \dot{z}\urcorner)[z\backslash m+n]$ is

$$\begin{aligned}\exists y', m', n', z' \big(&\mathsf{NumC}(m, m') \wedge \mathsf{NumC}(n, n') \wedge \mathsf{NumC}(m+n, z') \wedge \\ &\mathsf{Subst}_3(\ulcorner m+n = z\urcorner, \ulcorner m\urcorner, \ulcorner n\urcorner, \ulcorner z\urcorner, m', n', z', y) \wedge \\ &\mathsf{Prov}_Q(y)\big).\end{aligned}$$

Work in $\mathrm{I}\Sigma_1$: Show $\mathsf{Prov}_Q(\ulcorner\dot{m} + \dot{n} = \dot{z}\urcorner)[z\backslash m+n]$ by induction on n. If $n = 0$ then $m + n = m$ and work in Q: by (Q4) we have $\underline{m} + 0 = \underline{m}$. Now, back in $\mathrm{I}\Sigma_1$, for the induction step we have $\mathsf{Prov}_Q(\ulcorner\dot{m} + \dot{n} = \dot{z}\urcorner)[z\backslash m+n]$ as induction hypothesis (IH) and we have to show that $\mathsf{Prov}_Q(\ulcorner\dot{m} + \dot{n} = \dot{z}\urcorner)[z\backslash m+n][n\backslash s(n)]$, i.e., $\mathsf{Prov}_Q(\ulcorner\dot{m} + s(\dot{n})) = \dot{z}\urcorner)[z\backslash m+n+1]$. Work in Q: We have $\underline{m} + s(\underline{n}) =^{(Q5)} s(\underline{m} + \underline{n}) =^{(\mathrm{IH})} s(\underline{m+n}) = \underline{m+n+1}$.

∎

We proceed similarly with the other points of Lemma 3.13 which are necessary for Lemma 3.15, for example Lemma 3.13/4 is formalised as $\mathrm{I}\Sigma_1 \vdash \forall m, n\,(m \neq n \to \mathsf{Prov}_Q(\ulcorner \dot{m} \neq \dot{n} \urcorner))$. We then proceed to show:

Lemma 3.54. *Let σ be a Σ_1 sentence, then $\mathrm{I}\Sigma_1 \vdash \sigma \to \mathsf{Prov}_Q(\ulcorner \sigma \urcorner)$.*

Proof sketch. First one shows

$$\mathrm{I}\Sigma_1 \vdash \varphi(x_1, \ldots, x_k) \to \mathsf{Prov}_Q(\ulcorner \varphi(\dot{x_1}, \ldots, \dot{x_k}) \urcorner) \tag{*}$$

for every Δ_0 formula $\varphi(x_1, \ldots, x_k)$ by induction on the structure of $\varphi(x_1, \ldots, x_k)$ as in the proof of Lemma 3.15.

Then one can carry out the following argument in $\mathrm{I}\Sigma_1$ for any Σ_1 sentence $\sigma \equiv \exists x\, \varphi(x)$: Assume σ, then there is an x s.t. $\varphi(x)$. So, by (*), there is a Q-proof p_x of $\varphi(\underline{x})$ and hence $q_x = p_x; \ulcorner \varphi(\underline{x}) \to \sigma \urcorner; \ulcorner \sigma \urcorner$ is a Q-proof of σ because $\varphi(\underline{x}) \to \sigma$ is an axiom and σ follows from modus ponens. ☐

Lemma 3.55 (D2)**.** *Let T be an axiomatisable theory containing Q and let σ be a formula, then $\mathrm{I}\Sigma_1 \vdash \mathsf{Prov}_T(\ulcorner \sigma \urcorner) \to \mathsf{Prov}_T(\ulcorner \mathsf{Prov}_T(\ulcorner \sigma \urcorner) \urcorner)$*

Proof. $\mathsf{Prov}_T(\ulcorner \sigma \urcorner)$ is a Σ_1 sentence, so, by Lemma 3.54, $\mathrm{I}\Sigma_1 \vdash \mathsf{Prov}_T(\ulcorner \sigma \urcorner) \to \mathsf{Prov}_Q(\ulcorner \mathsf{Prov}_T(\ulcorner \sigma \urcorner) \urcorner)$. Since T contains Q, $\mathrm{I}\Sigma_1 \vdash \mathsf{Prov}_Q(x) \to \mathsf{Prov}_T(x)$. Thus we obtain $\mathrm{I}\Sigma_1 \vdash \mathsf{Prov}_T(\ulcorner \sigma \urcorner) \to \mathsf{Prov}_T(\ulcorner \mathsf{Prov}_T(\ulcorner \sigma \urcorner) \urcorner)$. ☐

3.9 The second incompleteness theorem

Definition 3.56. For an axiomatisable theory T containing Q define $\mathsf{Con}_T \equiv \neg \mathsf{Prov}_T(\ulcorner 0 = \underline{1} \urcorner)$.

Lemma 3.57. *Let T be an axiomatisable theory containing Q and let σ be a sentence. Then*

1. $\mathrm{I}\Sigma_1 \vdash \neg \mathsf{Prov}_T(\ulcorner \sigma \urcorner) \to \mathsf{Con}_T$ *and*
2. $\mathrm{I}\Sigma_1 \vdash \mathsf{Prov}_T(\ulcorner \sigma \urcorner) \wedge \mathsf{Prov}_T(\ulcorner \neg \sigma \urcorner) \to \neg \mathsf{Con}_T$.

Proof. Let τ, ν be any sentences, then $\tau \to \neg \tau \to \nu$ is a tautology, so $T \vdash \tau \to \neg \tau \to \nu$. Therefore, by (D1), $\mathrm{I}\Sigma_1 \vdash \mathsf{Prov}_T(\ulcorner \tau \to \neg \tau \to \nu \urcorner)$. By applying (D3) twice we obtain

$$\mathrm{I}\Sigma_1 \vdash \mathsf{Prov}_T(\ulcorner \tau \urcorner) \to \mathsf{Prov}_T(\ulcorner \neg \tau \urcorner) \to \mathsf{Prov}_T(\ulcorner \nu \urcorner).$$

This immediately entails 2 by letting $\tau \equiv \sigma$ and $\nu \equiv 0 = \underline{1}$. For 1 let $\tau \equiv 0 = \underline{1}$ and $\nu \equiv \sigma$. Observe that, since $Q \vdash 0 \neq \underline{1}$, we have $\mathrm{I}\Sigma_1 \vdash \mathsf{Prov}_T(\ulcorner 0 \neq \underline{1} \urcorner)$ by (D1). Therefore, $\mathrm{I}\Sigma_1 \vdash \mathsf{Prov}_T(\ulcorner 0 = \underline{1} \urcorner) \to \mathsf{Prov}_T(\ulcorner \sigma \urcorner)$ and we obtain 1 by contraposition. ☐

Theorem 3.58 (Second incompleteness theorem)**.** *Let T be a consistent and axiomatisable theory containing $\mathrm{I}\Sigma_1$, then $T \nvdash \mathsf{Con}_T$.*

Proof. By the first incompleteness theorem for G_T we know $T \nvdash \mathsf{G}_T$. Therefore it suffices to show that $T \vdash \mathsf{G}_T \leftrightarrow \mathsf{Con}_T$. We have $T \vdash \mathsf{G}_T \to \neg \mathsf{Prov}_T(\ulcorner \mathsf{G}_T \urcorner)$ and so, by Lemma 3.57/1., $T \vdash \mathsf{G}_T \to \mathsf{Con}_T$. Conversely, we will show $T \vdash \neg \mathsf{G}_T \to \neg \mathsf{Con}_T$. To that aim, it suffices to show that $T \vdash \mathsf{Prov}_T(\ulcorner \mathsf{G}_T \urcorner) \to \neg \mathsf{Con}_T$. First, by (D2), we have

$$T \vdash \mathsf{Prov}_T(\ulcorner \mathsf{G}_T \urcorner) \to \mathsf{Prov}_T(\ulcorner \mathsf{Prov}_T(\ulcorner \mathsf{G}_T \urcorner) \urcorner).$$

Moreover, since $T \vdash \mathsf{Prov}_T(\ulcorner \mathsf{G}_T \urcorner) \to \neg \mathsf{G}_T$, by (D1), we have

$$T \vdash \mathsf{Prov}_T(\ulcorner \mathsf{Prov}_T(\ulcorner \mathsf{G}_T \urcorner) \to \neg \mathsf{G}_T \urcorner)$$

so, by (D3),

$$T \vdash \mathsf{Prov}_T(\ulcorner \mathsf{G}_T \urcorner) \to \mathsf{Prov}_T(\ulcorner \neg \mathsf{G}_T \urcorner).$$

But now, by Lemma 3.57/2., $T \vdash \mathsf{Prov}_T(\ulcorner \mathsf{G}_T \urcorner) \to \neg \mathsf{Con}_T$. $\square$

We start the discussion of this result with a few technical remarks. First, observe that Con_T is a Π_1 sentence. It can be written as $\forall x\, \neg \mathsf{Proof}_T(\ulcorner 0 = 1 \urcorner, x)$. Each of its instances $\neg \mathsf{Proof}_T(\ulcorner 0 = 1 \urcorner, \underline{n})$ is a Δ_0 sentence and, for a consistent theory T, provable already in Q. The difficulty lies proving the universally quantified sentence.

Let T be a consistent, axiomatisable theory containing $\mathrm{I}\Sigma_1$, then $T \nvdash \mathsf{Con}_T$, and therefore $T \cup \{\neg \mathsf{Con}_T\}$ is consistent. By the completeness theorem, there is a model $\mathcal{M} \models T \cup \{\neg \mathsf{Con}_T\}$. Then $\mathcal{M} \models T$ and there is a $p \in \mathcal{M}$ s.t. $\mathcal{M} \models \mathsf{Proof}_T(\ulcorner 0 = 1 \urcorner, p)$. Now, p cannot be a standard number, for if it were, then $\mathbb{N} \models \mathsf{Proof}_T(\ulcorner 0 = 1 \urcorner, p)$ and thus T would be inconsistent. So, even though T is consistent, $\mathcal{M}$ thinks it is not, while, at the same time, being a model of T and, in this sense, a witness for its consistency.

Note that, if T is inconsistent, then T proves everything, including its own consistency. Therefore the assumption of consistency of T is necessary in the above theorem. The assumption of axiomatisability is a very mild one, in particular it applies to all theories that have been used for formalising mathematics, like PA, ZFC, etc. Moreover, if we want to formalise usual mathematical reasoning in a logical theory T, then surely T contains $\mathrm{I}\Sigma_1$ since the axioms of Q are very basic properties of the natural numbers and induction (not just for Σ_1 formulas) is an indispensable reasoning principle in mathematics. Therefore the second incompleteness theorem applies to any sensible logical theory T that is intended as formalisation of usual mathematical reasoning and thus shows that $T \nvdash \mathsf{Con}_T$.

Coming back to the discussion of the historical context of this result, remember that Hilbert's programme called for a proof of consistency of a logical theory T formalising usual mathematical reasoning based on "finitary mathematics", i.e., in a theory S formalising the elementary properties of strings of symbols. Now, if T does not prove Con_T, then the much weaker theory S does not prove Con_T either. Thus the second incompleteness theorem has put an end to Hilbert's programme. Nevertheless, consistency proofs can be carried out in interesting and useful ways (but necessarily in a theory stronger than the one whose consistency is proven). It would go beyond the scope of this course to treat consistency proofs in detail, we just give a short overview of prominent approaches:

1. *Ordinal analysis*: for proving the consistency of a theory T we define a simple, usually primitive recursive, proof transformation. The iteration of this proof transformation, "cut-elimination", translates an arbitrary T proof into one of which it is possible to establish with elementary means that it does not prove $\perp$. The part that transcends the theory T is the statement that the iteration terminates. This statement follows from the assumption of the well-foundedness of a certain ordinal which depends on, and, in a certain sense, characterises the strength of the theory T.

2. *Functional interpretations*: the consistency of a first-order theory T is reduced to that of a quantifier-free proof system which, instead of quantifiers, contains primitive recursion of higher types. This reduction is achieved by a proof transformation. Such functional interpretations are also useful tools when applied to actual mathematical proofs for obtaining computational information from them.

3. *Reverse mathematics*: Instead of proving theorems from axioms, as usual in (formal) mathematics, one can also prove axioms from (sufficiently strong) theorems in a (weak) base theory. For example, this allows to establish that the Bolzano Weierstraß theorem (every bounded sequence of real numbers has an accumulation point) is equivalent to ACA_0, a conservative extension of PA.

4. *Relative consistency proofs*: the consistency of a theory T is shown under the assumption of the consistency of some other theory S. A famous example is $S = \mathrm{ZF}$ and $T = \mathrm{ZFC} + \mathrm{CH}$.

The deeper understanding of the foundations of mathematics obtained since the inception of Hilbert's programme, including many results from the above-mentioned approaches, has led to the dissipation of doubts about the consistency of mathematical reasoning. So, even though Hilbert's programme could not be carried out as conceptualised originally, it is fair to say that its strategic aim has largely been met.

Chapter 4

Further Topics

4.1 Provability logic

In this section we will have a quick look on *provability logic*, a modal logic that interprets the box modality as "provable". The motivation stems from the observation that, in a suitable syntax for Prov_T, the proof of the second incompleteness theorem from the derivability conditions is purely propositional.

Definition 4.1. Formulas in *modal logic* are built from propositional variables $p_1, p_2, \ldots$, truth values $\top, \bot$, propositional connective $\wedge, \vee, \neg, \rightarrow$, and the modal operator $\Box$ ("box") which, given any formula φ yields a formula $\Box\varphi$.

Often one considers a second modal operator $\Diamond$ ("diamond") which in dual to $\Box$, i.e., $\Diamond\varphi$ is defined as abbreviation of $\neg\Box\neg\varphi$. The modal operators $\Box$ and $\Diamond$ have a variety of different interpretations in the literature, depending on the intended applications of particular modal logics. For example, in epistemic logic $\Box\varphi$ is interpreted as "I know φ", in temporal logic $\Box\varphi$ is interpreted as "φ holds at all future points in time", etc. and the $\Diamond$ operator has the corresponding dual meaning. In the context of provability logic we will think of $\Box\varphi$ as the statement "φ is provable" (in some fixed theory T). Consequently $\Diamond\varphi$ expresses that φ is consistent with T. More precisely:

Definition 4.2. Let T be a consistent and axiomatisable theory containing $\mathrm{I}\Sigma_1$. An *arithmetical interpretation for T* is an assignment of modal formulas A to arithmetical sentences A^* satisfying:

1. If p is atomic then p^* is an L_A sentence.
2. $\top^* \equiv 0 = 0$ and $\bot^* \equiv 0 = \underline{1}$.
3. $\cdot^*$ commutes with $\wedge$, $\vee$, $\rightarrow$, and $\neg$
4. $(\Box A)^* = \mathsf{Prov}_T(\ulcorner A^* \urcorner)$

Definition 4.3. The modal logic **K4** consists of all formulas derivable from propositional tautologies together with

(D1) the rule $\dfrac{\varphi}{\Box\varphi}$ (necessitation),

(D2) the axiom scheme $\Box\varphi \rightarrow \Box\Box\varphi$, (D3) the axiom scheme $\Box\varphi \to \Box(\varphi \to \psi) \to \Box\psi$, and

the rule $\dfrac{\varphi \quad \varphi \to \psi}{\psi}$ (modus ponens)

Note that, under an arithmetical interpretation, the definition of **K4** is propsitional logic with the derivability conditions. In particular, whenever $\mathbf{K4} \vdash \varphi$, then $T \vdash \varphi^*$ since T satisfies the derivability conditions. This property is also called *arithmetical soundness* of **K4**. Since it encapsulates the derivability conditions, one can formulate our proof of the second incompleteness theorem in this logic. To that aim suppose we have a Gödel sentence, i.e., a propositional variable G s.t. $\vdash G \leftrightarrow \neg\Box G$. The consistency is the formula $\neg\Box\bot$ which we abbreviate as C. Then the second incompleteness theorem can be formulated as follows:

Theorem 4.4. $\mathbf{K4} \nvdash C$

Proof Sketch. First, Lemma 3.57 is formalised as

1. $\mathbf{K4} \vdash \neg\Box\sigma \to C$
2. $\mathbf{K4} \vdash \Box\sigma \wedge \Box\neg\sigma \to \neg C$

for σ being any propositional variable. This lemma can be proved using the derivability conditions, i.e., the definiton of **K4**

For Theorem 3.58 we assume that $\mathbf{K4} \nvdash G$ so that it suffices to show that $\mathbf{K4} \vdash G \leftrightarrow C$. For the left-to-right direction we have $\mathbf{K4} \vdash G \leftrightarrow \neg\Box G$ so, by D1, $\mathbf{K4} \vdash G \to C$. For the right-to-left direction we have to show $\mathbf{K4} \vdash C \to G$, i.e., $\mathbf{K4} \vdash C \to \neg\Box G$, i.e., $\mathbf{K4} \vdash \Box G \to \neg C$. By (D2) we have $\mathbf{K4} \vdash \Box G \to \Box\Box G$. Moreover, since $\mathbf{K4} \vdash \Box G \to \neg G$, by (D1) we have

$$\mathbf{K4} \vdash \Box(\Box G \to \neg G).$$

So, by (D3) we have

$$\mathbf{K4} \vdash \Box G \to \Box\neg G.$$

So, by 2. we have

$$\mathbf{K4} \vdash \Box G \to \neg C.$$

□

The logic **K4** thus captures a significant part of reasoning with provability. However, it pays out to go still one more step further. After consideration of the Gödel sentence G_T which expresses "I am not provable" it is natural to ask about the status of a sentence which expresses "I am provable". We are now in a position to clarify its status.

Definition 4.5. Let T be an axiomatisable theory containing Q. By the fixed point lemma there is a sentence H_T satisfying $T \vdash \mathsf{H}_T \leftrightarrow \mathsf{Prov}_T(\ulcorner \mathsf{H}_T \urcorner)$. This sentence is called *Henkin sentence of T*.

Lemma 4.6. *Let T be an axiomatisable theory containing Q, σ be a sentence and $T' = T \cup \{\sigma\}$, then*

$$\mathrm{I}\Sigma_1 \vdash \neg\mathsf{Con}_{T'} \to (\neg\mathsf{Con}_T \vee \mathsf{Prov}_T(\ulcorner \neg\sigma \urcorner))$$

Proof. Work in $\mathrm{I}\Sigma_1$: if $\neg\mathsf{Con}_{T'}$, then there is a T' proof p' of $0 = \underline{1}$. If p' is a T proof, we are done. If not, obtain a T proof p of $\sigma \to 0 = \underline{1}$ from p' and, by appending a Q proof of $0 \neq 1$ and some propositional reasoning, obtain a T-proof of $\neg\sigma$. □

Theorem 4.7 (Löb's Theorem)**.** *Let T be a consistent and axiomatisable theory containing* $\mathrm{I}\Sigma_1$ *and let τ be a sentence. Then $T \vdash \mathsf{Prov}_T(\ulcorner\tau\urcorner) \to \tau$ implies $T \vdash \tau$.*

Note that $T \vdash \tau$ implies $T \vdash \mathsf{Prov}_T(\ulcorner\tau\urcorner) \to \tau$ trivially.

Proof. Let $T' = T \cup \{\neg\tau\}$ and assume that $T \vdash \mathsf{Prov}_T(\ulcorner\tau\urcorner) \to \tau$. Then $T \vdash \neg\tau \to \neg\mathsf{Prov}_T(\ulcorner\tau\urcorner)$, thus $T' \vdash \neg\mathsf{Prov}_T(\ulcorner\tau\urcorner)$ and, by Lemma 3.57/1, $T' \vdash \mathsf{Con}_T$. Moreover, by Lemma 4.6, $T' \vdash \mathsf{Con}_T \wedge \neg\mathsf{Prov}_T(\ulcorner\tau\urcorner) \to \mathsf{Con}_{T'}$ and hence $T' \vdash \mathsf{Con}_{T'}$. Now, T' is axiomatisable and contains $\mathrm{I}\Sigma_1$, so, by the second incompleteness theorem, T' is inconsistent, i.e., $T \vdash \tau$. □

Corollary 4.8. *Let T be a consistent and axiomatisable theory containing* $\mathrm{I}\Sigma_1$*, then $T \vdash \mathsf{H}_T$.*

Proof. By definition, $T \vdash \mathsf{Prov}_T(\ulcorner\mathsf{H}_T\urcorner) \to \mathsf{H}_T$, so, by Löb's Theorem, $T \vdash \mathsf{H}_T$. □

Definition 4.9. The logic **GL** ("Gödel-Löb") is obtained from **K4** by adding the rule

$$\frac{\Box\varphi \to \varphi}{\varphi}$$

Note that this rule is just Löb's theorem. Just as **K4** also **GL** is arithemtically sound. Moreover, Gödel-Löb logic has a remarkable completeness property w.r.t. arithmetical interpreations.

Theorem 4.10 (Arithmetical Soundness)**.** *Let T be a consistent and axiomatisable theory that contains* $\mathrm{I}\Sigma_1$*. Let φ be a modal formula. If* **GL** $\vdash \varphi$*, then, for all arithmetical interpretations $\cdot^*$ for T, $T \vdash \varphi^*$.*

Proof. This follows straightforwardly from the fact that T satisfies the derivability conditions and Löb's theorem. □

Theorem 4.11 (Arithmetical Completeness)**.** *Let T be a consistent and axiomatisable theory that contains* $\mathrm{I}\Sigma_1$*. Let φ be a modal formula. If, for all arithmetical interpretations $\cdot^*$ of T, $T \vdash \varphi^*$, then* **GL** $\vdash \varphi$*.*

Without Proof. □

Another remarkble property of **GL** is that it is decidable. Therefore, many questions about which statements about provability are provable are surprisingly easy to settle.

4.2 Presburger arithmetic

In this section we will have a look at arithmetic without multiplication to see that this changes the situation drastically. In Corollary 3.37 to the first incompleteness theorem, we have seen that every theory that contains Q is undecidable. In the absence of multiplication, this is no longer true.

Definition 4.12. We define the structure $\mathbb{N}_+ = (\mathbb{N}, 0, s, +, \leq)$.

The theory $\mathrm{Th}(\mathbb{N}_+)$, being the theory of a model, is consistent and complete. In this chapter we will show that it is also decidable. This is in stark contrast to $\mathrm{Th}(\mathbb{N})$ which is not even arithmetically definable, cf. Theorem 2.26. The proof technique for showing this decidability result is quantifier elimination.

A theory T is said to have quantifier-elimination, if, for every formula φ, there is a quantifier-free formula ψ s.t. $T \vdash \varphi \leftrightarrow \psi$. Quantifier elimination is an important technique for proving decidability results that has been applied successfully in many cases. It is typically used as follows: if the mapping from φ to ψ is computable and T-provability of quantifier-free formulas is decidable, then T is decidable.

For showing the decidability of $\mathrm{Th}(\mathbb{N}_+)$ it will be helpful to work in a larger structure.

Definition 4.13. For $a, b \in \mathbb{Z}$ and $m \geq 2$ write $a \equiv_m b$ if a is congruent to b modulo m. Define the language $L_\equiv = \{0/0, s/1, +/2, -/1, </2, \equiv_2/2, \equiv_3/2, \dots\}$ and the $L_\equiv$-structure $\mathbb{Z}_\equiv = (\mathbb{Z}, 0, s, +, -, <, \equiv_2, \equiv_3, \dots)$.

Lemma 4.14. *There is an algorithm that transforms every $L_\equiv$-formula φ into a quantifier-free formula ψ s.t. $\mathrm{FV}(\psi) \subseteq \mathrm{FV}(\varphi)$ and $\mathbb{Z}_\equiv \models \varphi \leftrightarrow \psi$.*

Proof. By replacing $\forall x$ by $\neg\exists x \neg$ we can assume that φ does not contain $\forall$. We proceed by induction on the structure of φ. The case of atomic formulas, as well as the induction steps concerning $\wedge$, $\vee$, and $\neg$ are trivial. It thus remains to treat the existential quantifier: by induction hypothesis φ is equivalent to a formula $\exists x\, \psi$ where ψ is quantifier-free. Using logical equivalences, ψ can be assumed to be in negation normal form. We obtain a negation-free formula ψ' which is $\mathbb{Z}_\equiv$-equivalent to ψ by applying the following equivalences:

$$\begin{aligned}
&\mathbb{Z}_\equiv \models \neg(t = u) \leftrightarrow t < u \vee u < t \\
&\mathbb{Z}_\equiv \models \neg(t < u) \leftrightarrow t = u \vee u < t \\
&\mathbb{Z}_\equiv \models \neg(t \equiv_m u) \leftrightarrow t \equiv_m s(u) \vee \dots \vee t \equiv_m s^{m-1}(u)
\end{aligned}$$

Using logical equivalences, we have

$$\mathbb{Z}_\equiv \models \exists x\, \psi' \quad \leftrightarrow \quad \exists x \bigvee_{i=1}^{n} \bigwedge_{j=1}^{k_i} A_{i,j} \quad \leftrightarrow \quad \bigvee_{i=1}^{n} \exists x \bigwedge_{j=1}^{k_i} A_{i,j}$$

where the $A_{i,j}$ are atoms. So it suffices to eliminate the quantifier from a formula χ_1 of the form $\exists x\, (B_1 \wedge \dots \wedge B_k)$.

In $\mathbb{Z}_\equiv$ every equation is equivalent to one of the form $nx = t$, every $<$-atom to one of the form $nx < t$ or $nx > t$ and every modulo-equation to one of the form $nx \equiv_m t$ where $n \in \mathbb{N}$, nx is an abbreviation for $x + \dots + x$ (n times), and t is a term that does not contain x. Thus we obtain a formula χ_2, equivalent to χ_1 in $\mathbb{Z}_\equiv$, where all atoms are of this form. Moreover, we can assume that every atom in χ_2 contains x, for if one, say B_1, does not, use $\mathbb{Z}_\equiv \models \exists x\, (B_1 \wedge \dots \wedge B_k) \leftrightarrow B_1 \wedge \exists x\, (B_2 \wedge \dots \wedge B_k)$. So,

$$\chi_2 \ \equiv \exists x \,\big(\bigwedge_{i=1}^{j} n_i x = t_i \bigwedge_{i=j+1}^{k} n_i x > t_i \bigwedge_{i=k+1}^{l} n_i x < t_i \bigwedge_{i=l+1}^{m} n_i x \equiv_{m_i} t_i \big)$$

where $t_1, \dots, t_m$ are terms not containing x. Let p be the least common multiple of $n_1, \dots, n_m$ and, for $i = 1, \dots, n$ define the term $u_i = \frac{p}{n_i} t_i$. Then

$$\chi_3 \ \equiv \exists x \,\big(\bigwedge_{i=1}^{j} px = u_i \bigwedge_{i=j+1}^{k} px > u_i \bigwedge_{i=k+1}^{l} px < u_i \bigwedge_{i=l+1}^{m} px \equiv_{m_i} u_i \big)$$

is equivalent to χ_2 in $\mathbb{Z}_\equiv$. So, in χ_3, x only occurs with coefficient p. Therefore, χ_3 is equivalent to

$$\chi_4 \;\equiv \exists y\,\big(\bigwedge_{i=1}^{j} y = u_i \;\bigwedge_{i=j+1}^{k} \; y > u_i \;\bigwedge_{i=k+1}^{l} \; y < u_i \;\bigwedge_{i=l+1}^{m} \; y \equiv_{m_i} u_i \wedge y \equiv_p 0\big)$$

and we can set $m_{m+1} = p$ and $u_{m+1} = 0$. Now, if $j \geq 1$, then χ_4 is equivalent to

$$\bigwedge_{i=2}^{j} u_1 = u_i \;\bigwedge_{i=j+1}^{k} \; u_1 > u_i \;\bigwedge_{i=k+1}^{l} \; u_1 < u_i \;\bigwedge_{i=l+1}^{m+1} \; u_1 \equiv_{m_i} u_i$$

and we are done. So we assume $j = 0$ and thus that χ_4 is of the form

$$\chi_4 \;\equiv \exists y\,\big(\bigwedge_{i=1}^{k} y > u_i \;\bigwedge_{i=k+1}^{l} \; y < u_i \;\bigwedge_{i=l+1}^{m+1} \; y \equiv_{m_i} u_i\big)$$

Now let q be the least common multiple of $m_{l+1}, \ldots, m_{m+1}$. Then $a + q \equiv_{m_i} a$ for all $a \in \mathbb{Z}$, so the pattern of residues modulo $m_{l+1}, \ldots, m_{m+1}$ has period q. If there are no upper and no lower bounds, i.e., $l = 0$, then χ_4 is equivalent to

$$\bigvee_{d=1}^{q} \bigwedge_{i=1}^{m+1} \underline{d} \equiv_{m_i} u_i$$

If there is at least one lower bound, i.e., $k \geq 1$, we start the case distinction at u_1 instead of 0, and thus χ_4 is equivalent to

$$\bigvee_{d=1}^{q} \big(\bigwedge_{i=2}^{k} u_1 + \underline{d} > u_i \;\bigwedge_{i=k+1}^{l} \; u_1 + \underline{d} < u_i \;\bigwedge_{i=l+1}^{m+1} \; u_1 + \underline{d} \equiv_{m_i} u_i\big)$$

If also $k = 0$ but there is at least one upper bound, i.e., $l \geq 1$, we start the case distinction at u_1 and use $-\underline{d}$ instead of $+\underline{d}$ to obtain

$$\bigvee_{d=1}^{q} \big(\bigwedge_{i=2}^{l} u_1 + -\underline{d} < u_i \;\bigwedge_{i=l+1}^{m+1} \; u_1 + -\underline{d} \equiv_{m_i} u_i\big)$$

which is equivalent to χ_4.

□

Theorem 4.15. Th($\mathbb{Z}_\equiv$) *is decidable.*

Proof. In light of the above quantifier-elimination lemma it suffices to observe that the truth of variable-free atoms in Th($\mathbb{Z}_\equiv$) is decidable. This is entailed by the decidability of the relations $<$, $=$, $\equiv_m$ on $\mathbb{Z} \times \mathbb{Z}$. □

Corollary 4.16. Th($\mathbb{N}_+$) *is decidable.*

Proof. We interpret $\mathrm{Th}(\mathbb{N}_+)$ in $\mathrm{Th}(\mathbb{Z}_\equiv)$ by using $\mathsf{N}(x) \equiv x = 0 \vee x > 0$ as definition of $\mathbb{N}$ and $\mathsf{LEQ}(x, y) \equiv x = y \vee x < y$ as definition of $\leq$. The symbols 0, s, and $+$ have trivial interpretations. Then we can decide $\mathrm{Th}(\mathbb{N}_+)$ as follows: given a $\{0, s, +, \leq\}$ sentence σ we compute its interpretation σ^* in $\mathrm{Th}(\mathbb{Z}_\equiv)$ and apply the decision procedure from Theorem 4.15 to σ^*. Since $*$ is an interpretation, $\mathbb{N}_+ \models \sigma$ implies $\mathbb{Z}_\equiv \models \sigma^*$. For the converse implication assume $\mathbb{N}_+ \not\models \sigma$, then $\mathbb{N}_+ \models \neg\sigma$, so $\mathbb{Z}_\equiv \models (\neg\sigma)^*$ and, by definition of $*$, $\mathbb{Z}_\equiv \models \neg(\sigma^*)$, i.e., $\mathbb{Z} \not\models \sigma^*$. $\square$

Quantifier-elimination has been used to show decidability results for many theories, e.g., algebraically closed fields, real closed fields, Abelian groups, etc.