Chapter 1

Undecidability and Incompleteness

In this final chapter, we explore how we can work with theories algorithmically. The highlight will be *Gödel's Incompleteness Theorems* from 1931, which show the limits of the formal approach and which halted the decades-long program of formalizing mathematics. We do not have enough space to provide formal definitions and complete proofs here, so we will sometimes operate on a somewhat intuitive level. We will focus on understanding the meaning of the statements and the ideas of the proofs.

We will understand the concept of an *algorithm* intuitively as well. If we wanted to formalize it, the most common (but by no means the only) choice would be the notion of a *Turing machine*.[1]

1.1 Recursive Axiomatization and Decidability

In the proof systems we have discussed (the tableau method, resolution, Hilbert calculus), we allowed the theory T in which we are proving to be infinite. However, we have not yet considered how it is given. If we want to verify that a given object (tableau, resolution tree, sequence of formulas) is a valid proof, we need some algorithmic access to all axioms of T.

One possibility would be to require an *enumerator* for T, i.e., an algorithm that writes out the axioms of T on the output, and each axiom is eventually written out.[2] Then it would be easy to confirm that a given proof is correct. However, if we received a proof that used an incorrect axiom, one that is not in T, we would never know: we would wait infinitely long to see if the enumerator outputs it at some point. Therefore, we require a stronger property that allows recognizing incorrect proofs as well: a *recursive axiomatization*.[3]

Definition 1.1.1 (Recursive Axiomatization)**.** A theory T is *recursively axiomatized* if there is an algorithm that, for every input formula φ, halts and answers whether $\varphi \in T$.

Remark 1.1.2. In fact, an enumerator for T would suffice if it is guaranteed to output the axioms in lexicographic order. That would be equivalent to a recursive axiomatization. (Think about why.)

[1] See the courses NTIN071 Automata and Grammars, NTIN090 Introduction to Complexity and Computability.

[2] A necessary condition is that T is countable. It suffices to assume that the language is countable.

[3] The word *recursive* here does not mean the commonly known recursion, but refers to the formalization of an algorithm using 'recursive functions' by Gödel. Recursive functions here mean the same as computable by some Turing machine; the theory of computability is sometimes called *recursion theory*.

We focus on the question whether we can 'algorithmically decide truth' in a given theory T (i.e., the validity of an input formula). If so, we say the theory is *decidable*. This is quite a strong property, so we also define *partial decidability*, which means that if the input formula is valid, the algorithm will tell us, but if it is not valid, we may never get an answer.

Definition 1.1.3 (Decidability)**.** A theory T is said to be

- *decidable* if there is an algorithm that, for every input formula φ, halts and answers whether $T \models \varphi$,
- *partially decidable* if there is an algorithm that, for every input formula:
  - if $T \models \varphi$, halts and answers "yes",
  - if $T \not\models \varphi$, either does not halt or halts and answers "no".

As usual, we can assume that φ in the above definition is a sentence. Let us prove a simple proposition:

Proposition 1.1.4. *Let T be recursively axiomatized. Then:*

(i) T is partially decidable,

(ii) if T is also complete, then it is decidable.

Proof. An algorithm showing partial decidability is the construction of a systematic tableau from T for the entry $\mathrm{F}\varphi$.[4] If φ is valid in T, the construction will end in finitely many steps and we can easily verify that the tableau is contradictory; otherwise, it may not halt.

If T is complete, we know that exactly one of the following holds: either $T \vdash \varphi$ or $T \vdash \neg\varphi$. We can therefore start constructing, in parallel, tableaux for $\mathrm{F}\varphi$ and for $\mathrm{T}\varphi$ (proof and refutation of φ from T): one of the constructions will halt after finitely many steps. □

1.1.1 Recursively Enumerable Completion

The requirement of completeness is too strong; we now show that it suffices to eb able to effectively describe all complete simple extensions.[5]

Definition 1.1.5 (Recursively Enumerable Completion)**.** We say that a theory T has *recursively enumerable completion* if (some) set of up to equivalence all complete simple extensions of the theory T is *recursively enumerable*, i.e., there is an algorithm that, for a given pair of natural numbers (i, j), outputs the i-th axiom of the j-th extension (in some fixed order[6]), or answers that such an axiom does not exist.[7]

Proposition 1.1.6. *If a theory T is recursively axiomatized and has a recursively enumerable completion, then T is decidable.*

[4] Here, an enumerator of the axioms of T suffices, or we can systematically generate all sentences (e.g., in lexicographic order) and test whether they are axioms.

[5] I.e., 'all models up to elementary equivalence'.

[6] Here, we need the language to be countable.

[7] If there are fewer than j extensions or if the j-th extension has fewer than i axioms.

Proof. For a given sentence φ, either $T \vdash \varphi$, or there is a counterexample $\mathcal{A} \not\models \varphi$, thus also a complete simple extension T_i of the theory T such that $T_i \not\vdash \varphi$. From the completeness of T_i, it follows that $T_i \vdash \neg\varphi$. Our algorithm will construct in parallel the tableau proof of φ from T and (gradually) the tableau proofs of $\neg\varphi$ from all complete simple extensions $T_1, T_2, \ldots$ of the theory T.[8] We know that at least one of the tableaux that we are constructing in parallel is contradictory, and we can assume it is finite (as long as we do not extend contradictory branches of the tableau), so the algorithm will finish constructing it after finitely many steps. □

Exercise 1.1. Show that the following theories have recursively enumerable completion:

- The theory of pure equality (the empty theory in the language $L = \langle\rangle$ with equality),
- The theory of unary predicate (the empty theory in the language $L = \langle U \rangle$ with equality, where U is a unary relation symbol),
- The theory of dense linear order DeLO* (the complete simple extensions are described in Corollary **??**),

These are recursively axiomatized theories (as they are finite), so they are decidable.

Example 1.1.7. Finally, let us mention without proof a few more examples of decidable theories:

- The theory of Boolean algebras (Alfred Tarski 1940),
- The theory of algebraically closed fields (Tarski 1949),
- The theory of commutative groups (Wanda Szmielew 1955).

These theories are also not complete, but recursively axiomatized and have recursively enumerable completion.

1.1.2 Recursive Axiomatizability

In the previous chapter, specifically in Section **??**, we addressed the question of when a class of structures [or a theory] can be described using axioms [of a certain form]. Now, let us focus on the question of when this can be done *algorithmically*.

Definition 1.1.8 (Recursive Axiomatizability)**.** A class of L-structures $K \subseteq \mathrm{M}_L$ is *recursively axiomatizable* if there is a recursively axiomatized L-theory T such that $K = \mathrm{M}_L(T)$. A theory T' is *recursively axiomatizable* if the class of its models is recursively axiomatizable, i.e., if T' is equivalent to some recursively axiomatized theory.

Remark 1.1.9. Similarly, we could define *recursively enumerable axiomatizability*.

Let us prove the following simple proposition:

Proposition 1.1.10. *If $\mathcal{A}$ is a finite structure in a finite language with equality, then the theory* $\mathrm{Th}(\mathcal{A})$ *is recursively axiomatizable.*

[8] It does not matter that there are infinitely many of them; we can use so-called *dovetailing*: Perform the 1st step of constructing the 1st tableau, then the 2nd step of the 1st tableau and the 1st step of the 2nd tableau, the 3rd step of the 1st tableau, the 2nd step of the 2nd tableau, the 1st step of the 3rd tableau, etc.

Remark 1.1.11. It follows that $\mathrm{Th}(\mathcal{A})$ is decidable, which is not surprising: the validity of a sentence φ in a finite structure $\mathcal{A}$ can be easily verified.

Proof. Let us enumerate the elements of the domain as $A = \{a_1, \ldots, a_n\}$. The theory $\mathrm{Th}(\mathcal{A})$ can be axiomatized by a single sentence of the form "there exist exactly n elements $a_1, \ldots, a_n$ satisfying precisely those *'basic relationships'* on function values and relation tuples that hold in the structure $\mathcal{A}$".[9] □

Let us give some standard examples of structures that can be 'described algorithmically':

Example 1.1.12. For the following structures, $\mathrm{Th}(\mathcal{A})$ is recursively axiomatizable, and thus decidable:

- $\langle \mathbb{Z}, \leq \rangle$, this is the so-called theory of *discrete linear orders*,
- $\langle \mathbb{Q}, \leq \rangle$, this is the theory DeLO,
- $\langle \mathbb{N}, S, 0 \rangle$, the theory of *successor with zero*,
- $\langle \mathbb{N}, S, +, 0 \rangle$, *Presburger arithmetic*,
- $\langle \mathbb{R}, +, -, \cdot, 0, 1 \rangle$, the theory of *real closed fields*,[10]
- $\langle \mathbb{C}, +, -, \cdot, 0, 1 \rangle$, the theory of *algebraically closed fields of characteristic 0*.

Corollary 1.1.13. *For the structures listed in Example 1.1.12, it holds that* $\mathrm{Th}(\mathcal{A})$ *is decidable.*

Remark 1.1.14. As follows from Gödel's First Incompleteness Theorem (see below), the *standard model of arithmetic*, i.e., the structure $\underline{\mathbb{N}} = \langle \mathbb{N}, S, +, \cdot, 0, \leq \rangle$, does *not* have recursively axiomatizable theory.

1.2 Arithmetic

The properties of natural numbers play an important role not only in mathematics but also, for example, in cryptography. Recall that the language of arithmetic is the language $L = \langle S, +, \cdot, 0, \leq \rangle$ with equality. As mentioned in Remark 1.1.14, the so-called *standard model of arithmetic* $\underline{\mathbb{N}} = \langle \mathbb{N}, S, +, \cdot, 0, \leq \rangle$ does not have a recursively axiomatizable theory. Therefore, we use recursively axiomatized theories that attempt to describe the properties of $\underline{\mathbb{N}}$ partially; these theories are called *arithmetics*.

[9] For example, if $f^{\mathcal{A}}(a_4, a_2) = a_{17}$, we add the atomic formula $f(x_{a_4}, x_{a_2}) = x_{a_{17}}$ to the conjunction (where x_{a_i} are variables corresponding to the individual elements). And if $(a_3, a_3, a_1) \notin R^{\mathcal{A}}$, we add $\neg R(x_{a_3}, x_{a_3}, x_{a_1})$.

[10] This significant result by A. Tarski (1949) also means that it is possible to algorithmically decide which properties hold in Euclidean geometry.

1.2.1 Robinson and Peano Arithmetic

We will mention only the two most important examples of arithmetics: *Robinson arithmetic* and *Peano arithmetic*.

Definition 1.2.1 (Robinson Arithmetic)**.** *Robinson arithmetic* is the theory Q in the language of arithmetic consisting of the following (finite) set of axioms:

$$\begin{array}{ll} \neg S(x) = 0 & x \cdot 0 = 0 \\ S(x) = S(y) \to x = y & x \cdot S(y) = x \cdot y + x \\ x + 0 = x & \neg x = 0 \to (\exists y)(x = S(y)) \\ x + S(y) = S(x + y) & x \leq y \leftrightarrow (\exists z)(z + x = y) \end{array}$$

Robinson arithmetic is very weak; it cannot prove, for example, the commutativity or associativity of addition or multiplication, or the transitivity of order.

On the other hand, it can prove all *existential statements about numerals* that are valid in $\underline{\mathbb{N}}$. By this, we mean formulas that, in prenex form, have only existential quantifiers, and into which we have substituted *numerals* $\underline{n} = S(\dots S(0)\dots)$ for the free variables.

Example 1.2.2. For instance, for the formula $\varphi(x, y)$ of the form $(\exists z)(x + z = y)$, we have that $Q \vdash \varphi(\underline{1}, \underline{2})$, where $\underline{1} = S(0)$ and $\underline{2} = S(S(0))$.

Thus, the following proposition holds; we will omit the proof.

Proposition 1.2.3. *If $\varphi(x_1, \dots, x_n)$ is an existential formula and $a_1, \dots, a_n \in \mathbb{N}$, then:*

$$Q \vdash \varphi(x_1/\underline{a_1}, \dots, x_n/\underline{a_n}) \text{ if and only if } \underline{\mathbb{N}} \models \varphi[e(x_1/a_1, \dots, x_n/a_n)]$$

A useful extension of Robinson arithmetic is the so-called Peano arithmetic, which allows for *proof by induction*:

Definition 1.2.4 (Peano Arithmetic)**.** *Peano arithmetic PA* is an extension of Robinson arithmetic Q with the *axiom schema of induction*, i.e., for each L-formula $\varphi(x, \overline{y})$, the following axiom is added:

$$(\varphi(0, \overline{y}) \wedge (\forall x)(\varphi(x, \overline{y}) \to \varphi(S(x), \overline{y}))) \to (\forall x)\varphi(x, \overline{y})$$

Peano arithmetic is a much better approximation of the theory $\mathrm{Th}(\underline{\mathbb{N}})$; it can prove all 'basic' properties valid in $\underline{\mathbb{N}}$ (such as the commutativity and associativity of addition). However, there are still sentences in the language of arithmetic that are valid in $\underline{\mathbb{N}}$ but independent in Peano arithmetic.[11]

Remark 1.2.5. If we moved up to *second-order* logic, we could axiomatize the structure $\underline{\mathbb{N}}$ (up to isomorphism) by extending Peano's arithmetic with the following second-order formula, the so-called *axiom of induction*:

$$(\forall X)((X(0) \wedge (\forall x)(X(x) \to X(S(x)))) \to (\forall x)X(x))$$

Recall that X represents (any) unary relation, i.e., subset of the universe. By applying the axiom of induction to the set of all successors of 0, we obtain that every element (of the given model) is a successor of zero. Thus, we can construct an isomorphism with $\underline{\mathbb{N}}$.

[11] As we will show in Gödel's First Incompleteness Theorem.

1.3 Undecidability of Predicate Logic

In this section, we will show that it is not possible to (algorithmically) decide the logical validity of first-order formulas. (In other words, the empty theory over a given language is undecidable.)

Theorem 1.3.1 (On undecidability of predicate logic)**.** *There is no algorithm that, given an input formula φ, decides whether it is logically valid.*[12]

Since we do not yet have the necessary formalism regarding algorithms, such as the notion of a Turing machine, we will choose another *undecidable problem* as a starting point. The most well-known is the so-called *Halting problem*, i.e., the question of whether a given program halts on a given input.[13] We will make our task simpler by choosing another undecidable problem, the so-called *Hilbert's Tenth Problem.*[14]

1.3.1 Hilbert's Tenth Problem

Let us have a polynomial $p(x_1, \ldots, x_n)$ with integer coefficients. Hilbert's Tenth Problem asks for an algorithm that decides whether such an input polynomial has an integer root, or in other words whether the *Diophantine equation* $p(x_1, \ldots, x_n) = 0$ has an (integer) solution:

> "Find an algorithm that, after finitely many steps, determines whether a given Diophantine equation with any number of variables and integer coefficients has an integer solution."

If Hilbert had lived to see the resolution of his tenth problem in 1970, he would have been surprised that no such algorithm exists.

Theorem 1.3.2 (Matiyasevich, Davis, Putnam, Robinson)**.** *The problem of existence of an integer solution to a given Diophantine equation with integer coefficients is (algorithmically) undecidable.*

We will not provide the proof here, for lack of space. Actually, to prove undecidability, we will use the following corollary, which talks about polynomials with natural coefficients and solutions in natural numbers.

Corollary 1.3.3. *There is no algorithm that, given a pair of polynomials $p(x_1, \ldots, x_n)$ and $q(x_1, \ldots, x_n)$ with* natural *coefficients, decides whether they have a natural solution, i.e., whether it holds that*

$$\underline{\mathbb{N}} \models (\exists x_1) \ldots (\exists x_n)\ p(x_1, \ldots, x_n) = q(x_1, \ldots, x_n)$$

Proof of the corollary. The proof is straightforward, utilizing the fact that every integer can be expressed as the difference of a pair of natural numbers, and conversely, every natural number can be expressed as the sum of four squares (of integers).[15] Thus, any Diophantine equation can be transformed into an equation from the corollary, and vice versa. □

[12] That is, whether the formula φ is a tautology, i.e., whether $\models \varphi$. Here we are talking about first-order formulas, in any language.

[13] You will prove its undecidability in the courses NTIN071 Automata and Grammars and again in NTIN090 Foundations of Complexity and Computability.

[14] Hilbert posed it in 1900 and published in 1902 along with 22 other problems; his problems significantly influenced 20th and 21st-century mathematics. Some remain unsolved, such as the Riemann hypothesis, see Wikipedia.

[15] The so-called Lagrange's Four-Square Theorem.

1.3.2 Proof of Undecidability

Recall that Robinson's arithmetic Q has only finitely many axioms, $\underline{\mathbb{N}}$ is its model, and it can prove all *existential statements about numerals* valid in $\underline{\mathbb{N}}$. We are now ready to prove the theorem on undecidability of predicate logic.

Proof of the theorem on undecidability of predicate logic. Consider a formula φ of the form

$$(\exists x_1) \dots (\exists x_n)\ p(x_1, \dots, x_n) = q(x_1, \dots, x_n)$$

where p and q are polynomials with natural coefficients. According to Proposition 1.2.3, we have:

$$\underline{\mathbb{N}} \models \varphi \text{ if and only if } Q \vdash \varphi$$

Let ψ_Q denote the conjunction (of the general closures) of all axioms of Q. Clearly, $Q \vdash \varphi$ if and only if $\psi_Q \vdash \varphi$, which holds if and only if $\vdash \psi_Q \to \varphi$. According to the Completeness Theorem, this is equivalent to $\models \psi_Q \to \varphi$. We thus obtain the following equivalence:

$$\underline{\mathbb{N}} \models \varphi \text{ if and only if } \models \psi_Q \to \varphi$$

This means that if there were an algorithm deciding logical validity, we could decide the existence of a natural solution to the equation $p(x_1, \dots, x_n) = q(x_1, \dots, x_n)$, i.e., Hilbert's Tenth Problem would be decidable.[16] This would be a contradiction. □

1.4 Gödel's Theorems

Finally, we will present the famous Gödel's incompleteness theorems, the understanding of which should be a natural part of every computer scientist's education. We will attempt to explain the principles of the proofs, but we will omit all technical details.

1.4.1 The First Incompleteness Theorem

First, we state Gödel's *First Incompleteness Theorem*, and explain the meaning of its assumptions.

Theorem 1.4.1 (The First Incompleteness Theorem)**.** *For every consistent recursively axiomatized extension T of Robinson's arithmetic, there exists a sentence that is true in $\underline{\mathbb{N}}$ but not provable in T.*

Such a sentence is called a *Gödel sentence*. Very informally speaking, Gödel's First Incompleteness Theorem states that the properties of the arithmetic of natural numbers cannot be 'reasonably', effectively described (in first-order logic), every such description is necessarily 'incomplete'. It is important to realize that we are talking about *truth* in the standard model of arithmetic, i.e., in the structure $\underline{\mathbb{N}}$, while *provability* is in the theory T. (According to the Completeness Theorem, every sentence *true in* T is also provable in T.)

Consistency is a necessary assumption because in an inconsistent theory every sentence is provable. Recall that recursive axiomatizability can be understood as 'effective specification'

[16] We show that there is a *reduction* of the 'hard' problem (Hilbert's Tenth) to our problem, thus our problem is also 'hard'.

of axioms (using an algorithm), without this property such a theory would be useless. The requirement that the theory be an extension of Robinson's arithmetic should be understood as the assumption that it has at least 'basic arithmetic strength', that it can 'talk' about natural numbers in a certain way. There are various versions of this assumption, with theories other than Robinson's arithmetic, and it is not necessary, for example, that it be a direct extension, it is enough if Robinson's arithmetic is 'definable' in the theory in a certain sense. But a theory in which 'natural numbers cannot be encoded' (and here it is important that we can talk not only about addition but also about multiplication) is 'too weak'.

It is good to realize that the following statement about 'incompleteness' also holds in particular:

Corollary 1.4.2. *If a theory T satisfies the assumptions of the First Incompleteness Theorem and moreover $\underline{\mathbb{N}}$ is a model of the theory T, then T is not complete.*

Proof. Assume for contradiction that T is complete. Consider the sentence φ, which is true in $\underline{\mathbb{N}}$ but not provable in T. By completeness, we know that $T \vdash \neg\varphi$, but then the Soundness Theorem says that $T \models \neg\varphi$, thus φ is false in $\underline{\mathbb{N}}$, which is a contradiction. □

Not only is the statement of the First Incompleteness Theorem interesting, but also its proof: Gödel came up with a completely new, revolutionary proof technique for its time. The sentence constructed in the proof formalizes the statement *"I am not provable in T"*, and the proof is based on the following two principles, which we will describe somewhat informally below:

- *arithmetization of syntax*, i.e., encoding sentences and their provability into natural numbers,
- *self-reference*, i.e., the ability of a sentence to 'talk about itself' (about its code).

Arithmetization of Provability

Finite syntactic objects, such as terms, formulas, finite tableaux, and thus also tableau proofs, can be 'reasonably' encoded into natural numbers.[17] The specific way this can be done, the so-called *Gödel numbering*, we will skip as a technical detail. It is enough for us that we can 'algorithmically' encode and decode objects (or 'simulate manipulation with objects' on their codes).

Denote the code of a formula φ as $\ulcorner\varphi\urcorner$, similarly for other syntactic objects. We will denote the numeral corresponding to the code of φ, i.e., the $\ulcorner\varphi\urcorner$-th numeral, as $\underline{\varphi}$. For a given theory T, we define the following binary relation on the set of all natural numbers:

$(n, m) \in \mathrm{Proof}_T$ if and only if $n = \ulcorner\varphi\urcorner$ and $m = \ulcorner\tau\urcorner$, where τ is a tableau proof of the sentence φ from T

If we have effective access to the axioms, we can also effectively check whether τ is really a proof of φ (where τ and φ we obtain by decoding m and n), so it holds:

Observation 1.4.3. *If T is recursively axiomatized, the relation $\mathrm{Proof}_T \subseteq \mathbb{N}^2$ is* recursive.

[17] Imagine any reasonable way to write the given object into a file. The file in binary code is a sequence of 0s and 1s. Prefix it with a one so as not to start with zero, and you have a binary representation of a natural number.

A crucial but very technical part of the proof of the First Incompleteness Theorem is the following proposition, which we will leave without proof.

Proposition 1.4.4. *If T is also an extension of Robinson's arithmetic Q, then there exists a formula $Prf_T(x, y)$ in the language of arithmetic that* represents *the relation* Proof_T*, i.e., for all $n, m \in \mathbb{N}$ it holds:*

- *If $(n, m) \in \text{Proof}_T$, then $Q \vdash Prf_T(\underline{n}, \underline{m})$,*
- *otherwise $Q \vdash \neg Prf_T(\underline{n}, \underline{m})$.*

The formula $Prf_T(x, y)$ thus expresses *"y is a proof of x in T"*.[18] Then we can express that *"x is provable in T"* by the formula $(\exists y) Prf_T(x, y)$. Note that the following observation holds, as the witness provides the code of some tableau proof, and $\underline{\mathbb{N}}$ satisfies the axioms of Q:

Observation 1.4.5. *$T \vdash \varphi$ if and only if $\underline{\mathbb{N}} \models (\exists y) Prf_T(\underline{\varphi}, y)$.*

We will also need the following corollary, which we will state without proof:

Corollary 1.4.6 (On the provability predicate)**.** *If $T \vdash \varphi$, then $T \vdash (\exists y) Prf_T(\underline{\varphi}, y)$.*

We can thus express that a given sentence is, or is not, provable. But how can a sentence say 'about itself' that it is not provable? For this, we use the *principle of self-reference*.

Self-reference

To illustrate the principle of self-reference, for clarity, let us imagine a sentence in Czech instead of a logical sentence, and instead of the property of "being provable" a statement about the number of letters. Consider the following sentence:

> `This sentence has 22 characters.`

In natural language, we easily express self-reference with the pronoun "This", from the context we know that it refers to the sentence itself. In formal systems, however, we typically do not have self-reference directly available. We usually have *direct reference* available, as in the following example:

> `The following sentence has 29 characters.  "The following sentence has 29 characters."`

Here, however, there is no self-reference. We help ourselves with a trick we will call 'doubling':

> `The following sentence, written once and again in quotes, has 149 characters.  "The following sentence, written once and again in quotes has 149 characters."`

Using direct reference and doubling, we can obtain self-reference.

Remark 1.4.7. The same principle can be used to construct a program in C whose output is its own code (34 is the ASCII code for quotes):

```
main(){char *c="main(){char *c=%c%s%c; printf(c,34,c,34);}"; printf(c,34,c,34);}
```

[18] More precisely, the tableau whose code is y is a proof of the sentence whose code is x.

1.4.2 Proof and Consequences

In this subsection, we will prove Gödel's First Incompleteness Theorem and discuss its consequences. We will need the following theorem, which describes how we technically use the principle of self-reference. It can be viewed as a form of 'diagonalization argument',[19] hence this theorem is sometimes also called the *diagonal lemma*.

Theorem 1.4.8 (Fixed-point theorem)**.** *If T is an extension of Robinson's arithmetic, then for every formula $\varphi(x)$ (in the language of theory T) there exists a sentence ψ such that:*

$$T \vdash \psi \leftrightarrow \varphi(\underline{\psi})$$

The sentence ψ is thus *self-referential*, it says about itself: "I have the property φ".[20] We will explain only the idea of the proof. Note how direct reference and doubling are used in the proof.

Proof. Consider a *doubling function*, a function $d\colon \mathbb{N} \to \mathbb{N}$ such that for every formula $\chi(x)$ it holds:

$$d(\ulcorner \chi(x) \urcorner) = \ulcorner \chi(\underline{\chi(x)}) \urcorner$$

The function d thus takes as input a natural number n, which it decodes as a formula in one variable, substitutes the corresponding numeral $\underline{n}$ into this formula,[21] and re-encodes the resulting sentence.

Using the assumption that T is an extension of Q, it can be shown that this function is *representable* in T. For simplicity, assume it is representable by a term,[22] and denote it also as d. This means that for every formula $\chi(x)$ it holds:

$$T \vdash d(\underline{\chi(x)}) = \underline{\chi(\underline{\chi(x)})}$$

Thus Robinson's arithmetic, and therefore our theory T, proves *about numerals*, that d really 'doubles'.

The desired self-referential sentence ψ is the sentence:[23]

$$\varphi(d(\underline{\varphi(d(x))}))$$

We want to prove that $T \vdash \psi \leftrightarrow \varphi(\underline{\psi})$, i.e., $T \vdash \varphi(d(\underline{\varphi(d(x))})) \leftrightarrow \varphi(\underline{\varphi(d(\underline{\varphi(d(x))}))})$. To do this, we need to verify that:

$$T \vdash d(\underline{\varphi(d(x))}) = \underline{\varphi(d(\underline{\varphi(d(x))}))}$$

But this we know from the representability of d, where we substitute the formula $\varphi(d(x))$ for $\chi(x)$. □

[19] Diagonalization refers to an argument reminiscent of *Cantor's diagonal argument*, known from the proof of the uncountability of $\mathbb{R}$. A similar argument, using self-reference, can be found, for example, in the *Barber paradox*, or in the proof of the undecidability of the *Halting problem*.

[20] More precisely, it says this about the numeral corresponding to its code.

[21] Here *numeral* corresponds to 'quotes' from the previous informal description of self-reference, and $d(\ulcorner \chi \urcorner)$ means "χ written once and again in quotes."

[22] Though in reality, it is represented by (a complex) formula.

[23] The following sentence, written once and again in quotes, has the property φ. "The following sentence, written once and again in quotes has the property φ."

Before proceeding to the proof of Gödel's theorem, we will show as a warm-up one consequence of the Fixed-point Theorem: By a *definition of truth* in an arithmetic theory T, we mean a formula $\tau(x)$ such that for every sentence ψ it holds:

$$T \vdash \psi \leftrightarrow \tau(\underline{\psi})$$

If a definition of truth existed, it would mean that instead of proving a sentence, it would suffice to calculate its code, substitute the corresponding numeral into τ, and evaluate.

Theorem 1.4.9 (Undefinability of truth)**.** *In no consistent extension of Robinson's arithmetic does a definition of truth exist.*

The proof uses the *Liar paradox*, expressing the sentence "I am not true in T".

Proof. Assume for contradiction that there exists a definition of truth $\tau(x)$. We use the Fixed-point Theorem, where we take $\neg\tau(x)$ as the formula $\varphi(x)$. We get the existence of a sentence ψ such that:

$$T \vdash \psi \leftrightarrow \neg\tau(\underline{\psi})$$

Since $\tau(x)$ is a definition of truth, it also holds that $T \vdash \psi \leftrightarrow \tau(\underline{\psi})$, hence $T \vdash \tau(\underline{\psi}) \leftrightarrow \neg\tau(\underline{\psi})$. This would mean that T is inconsistent. □

The proof of Gödel's theorem uses the same trick, but for the sentence "I am not provable in T".

Proof of the First Incompleteness Theorem. Let T be a consistent recursively axiomatized extension of Robinson's arithmetic. We want to find a Gödel sentence ψ_T that is true in $\underline{\mathbb{N}}$ but not provable in T.

Such a sentence is obtained from the Fixed-point Theorem as a sentence expressing "I am not provable in T". Let $\varphi(x)$ be the formula $\neg(\exists y)Prf_T(x, y)$ ("x is not provable in T"). According to the Fixed-point Theorem, there exists a sentence ψ_T satisfying:

$$T \vdash \psi_T \leftrightarrow \neg(\exists y)Prf_T(\underline{\psi_T}, y)$$

The sentence ψ_T is thus in T equivalent to the sentence that expresses that ψ_T is not provable in T. It can be shown that the same equivalence holds in $\underline{\mathbb{N}}$ (as that is how we constructed Prf_T and ψ_T):

$$\underline{\mathbb{N}} \models \psi_T \text{ if and only if } \underline{\mathbb{N}} \models \neg(\exists y)Prf_T(\underline{\psi_T}, y)$$

From Observation 1.4.5, we get that

$$\underline{\mathbb{N}} \models \psi_T \text{ if and only if } T \nvdash \psi_T$$

that is, ψ_T is true in $\underline{\mathbb{N}}$ if and only if it is not provable in T. It is sufficient to show that ψ_T is not provable in T. Assume for contradiction that $T \vdash \psi_T$. From self-reference, we know that $T \vdash \neg(\exists y)Prf_T(\underline{\psi_T}, y)$. From Corollary 1.4.6 on the provability predicate, we get that $T \vdash (\exists y)Prf_T(\underline{\psi_T}, y)$, which would mean that T is inconsistent. □

Finally, we will show two consequences and one strengthening. The following immediate consequence we mentioned earlier:

Corollary 1.4.10. *If T is a recursively axiomatized extension of Robinson's arithmetic and $\underline{\mathbb{N}}$ is a model of the theory T, then T is not complete.*

Proof. Since T has a model, it is not inconsistent. It thus satisfies the assumptions of the First Incompleteness Theorem, so the Gödel sentence ψ_T is not provable in it. If it were complete, it would have to prove $\neg\psi_T$. But that would mean that $\underline{\mathbb{N}} \models \neg\psi_T$, while we know that ψ_T is true in $\underline{\mathbb{N}}$. □

From this, it follows that the standard model of natural numbers cannot be recursively axiomatized:

Corollary 1.4.11. *The theory* $\mathrm{Th}(\underline{\mathbb{N}})$ *is not recursively axiomatizable.*

Proof. The theory $\mathrm{Th}(\underline{\mathbb{N}})$ is an extension of Robinson's arithmetic and is true in the model $\underline{\mathbb{N}}$. If it were recursively axiomatizable, any of its recursive axiomatizations could not be complete, according to the previous corollary. But $\mathrm{Th}(\underline{\mathbb{N}})$ is complete. □

One of the strengthenings of Gödel's First Theorem is the following statement, which we will state without proof. It shows that the assumption $\underline{\mathbb{N}} \models T$ in the first corollary above is, in fact, redundant.

Theorem 1.4.12 (Rosser's Trick, 1936)**.** *In every consistent recursively axiomatized extension of Robinson's arithmetic, there exists an independent sentence. Thus, such a theory is not complete.*

1.4.3 The Second Incompleteness Theorem

Gödel's Second Incompleteness Theorem states, informally speaking, that an effectively given, sufficiently rich theory cannot prove its own consistency. Consistency ("conservatism") is expressed by the following sentence, which we denote as Con_T:

$$\neg(\exists y)Prf_T(\underline{0 = S(0)}, y)$$

Note that it holds $\underline{\mathbb{N}} \models Con_T$, if and only if $T \nvdash 0 = S(0)$. In other words, the sentence Con_T indeed expresses *"The theory T is consistent".*

Theorem 1.4.13 (The Second Incompleteness Theorem)**.** *For every consistent recursively axiomatized extension T of Peano's arithmetic, the sentence Con_T is not provable in T.*

Note that the sentence Con_T is true in $\underline{\mathbb{N}}$ (as T is indeed consistent). It is also worth noting that full Peano arithmetic is not required, a weaker assumption is sufficient. Now we will show the main idea of the proof of the Second Theorem:

Proof of the Second Incompleteness Theorem. Consider the Gödel sentence ψ_T expressing "I am not provable in T". In the proof of the First Incompleteness Theorem (specifically in the first part), we showed that:

> "If T is consistent, then ψ_T is not provable in T."

From this, it follows that $T \nvdash \psi_T$, as T is consistent. On the other hand, it can be formulated as "$Con_T \to \psi_T$ holds" and if T is an extension of Peano's arithmetic, this statement can be formalized within the theory T, so it can be shown that:

$$T \vdash Con_T \to \psi_T$$

If it held that $T \vdash Con_T$, we would also get $T \vdash \psi_T$, which would be a contradiction. □

Finally, we will show three consequences of the Second Theorem.

Corollary 1.4.14. *There exists a model of PA, in which the sentence* $(\exists y) Prf_{PA}(\underline{0 = S(0)}, y)$ *holds.*

Proof. The sentence Con_{PA} is not provable, hence also not true in PA. But it holds in $\underline{\mathbb{N}}$ (as PA is consistent), which means that Con_{PA} is independent in PA. Therefore, its negation must hold in some model, which is equivalent to $(\exists y) Prf_{PA}(\underline{0 = S(0)}, y)$.

□

Note that this must be a non-standard model of PA, with the witness being a *non-standard* element (i.e., one that is not the value of any numeral).

Corollary 1.4.15. *There exists a consistent recursively axiomatized extension T of Peano's arithmetic, which 'proves its inconsistency', i.e., such that $T \vdash \neg Con_T$.*

Proof. Consider the theory $T = PA \cup \{\neg Con_{PA}\}$. This theory is consistent, as $PA \nvdash Con_{PA}$. It trivially holds that $T \vdash \neg Con_{PA}$ (i.e., T 'proves the inconsistency' of the theory PA). Since $PA \subseteq T$, it also holds that $T \vdash \neg Con_T$. □

Note that $\underline{\mathbb{N}}$ cannot be a model of the theory T.

Finally, let us look at the theory ZFC, i.e., Zermelo-Fraenkel set theory with the axiom of choice, which forms the foundation of formalized mathematics. This theory is not formally an extension of PA, but Peano arithmetic can be 'interpreted' in it (in a certain sense). This means that even this theory cannot prove its own consistency.

Corollary 1.4.16. *If the set theory ZFC is consistent, then the sentence Con_{ZFC} is not provable in ZFC.*

Thus, if someone were to prove within the theory ZFC that ZFC is consistent, it would mean that ZFC is inconsistent. This serves as a fitting conclusion to our lecture.

Bibliography