Introduction to Mathematical Logic

Course Notes

September 9, 2021

Manuel Bodirsky, Institut für Algebra, TU Dresden

Disclaimer: this is a draft and probably contains many typos and mistakes. Please report them to Manuel.Bodirsky@tu-dresden.de.

Contents

CHAPTER 1

Introduction

Understanding and finding proofs is a central part of mathematics. But what *is* a proof, actually? Can the concept of *a proof* be formalised? Can we program a computer so that the computer can verify proofs and even search for proofs? What are the usual axioms of mathematics that we are allowed to use in proofs?

To formalise the notion of a proof we need logic; more specifically, we will work with so-called *first-order logic*. First-order logic is in many respects the most important logic, both in mathematics and in computer science. It strikes a good balance between *expressiveness* on the one hand and *good mathematical properties* on the other hand. For example, first-order logic is expressive enough to formulate the axioms of axiomatic set theory, e.g., Zermelo–Fraenkel set theory (ZFC; Chapter 4) or Bernays-Gödel set theory. Practically all of mathematics can be formalised in this way. What is even more important is that first-order logic has a very rich model theory; here we have to refer to model theory courses. In computer science, first-order logic may be viewed as the most important database query language. Various restrictions and extensions of first-order logic might be relevant in applications, but first-order logic remains the central point of departure.

The first bigger goal of the course is proving *Gödel's completeness theorem* which shows that there exists a notion of formal proof such that a statement in first-order logic is true (valid) if and only if it has a formal proof. This statement has remarkable consequences; for example, it implies the *compactness theorem* for first-order logic, which has a great number of beautiful applications throughout mathematics, as we will see. The name *compactness* is borrowed from topology, and indeed there is a certain topology such that the compactness theorem for first-order logic translates into the statement that this space is compact.

A remarkable feature of the notion of a formal proof is that formal proofs can be checked by a computer program. Therefore, proofs can also be found by a program: a program may exhaustively list all potential proofs and then check their correctness. Obviously, this is not an efficient way of searching for proofs. What appears to be a computer science topic actually has quite remarkable consequences in mathematics, because we can use this fact to prove that there are first-order sentences about set theory that are *independent from ZFC*, i.e., there are models of ZFC where the sentence is true, and there are models of ZFC where the sentence is false. In fact, such independent statements must exist for any formal system that can express a sufficiently strong (in a certain formal sense) part of arithmetic; this is *Gödel's first incompleteness theorem*. This theorem can be strengthened as follows: we may even prove that if ZFC is consistent, then there is no proof in ZFC that ZFC is consistent; this is *Gödel's second incompleteness theorem*.

Outline. Before introducing first-order logic, we start with an elementary chapter about propositional logic. This is a small logic microcosmos, and certain general ideas and basic facts of logic can already be introduced there (Chapter 2).

We then introduce first-order logic, in several steps: we first introduce *(first-order) structures*. These are the objects that first-order logic is talking about. I assume that the reader is already familiar with some examples of structures, such as fields, groups, graphs, or partial orders. On the other hand, I do not assume that the reader is familiar with the formal definition of structures and present the formal definition in full detail, including basic operations on structures such as *homomorphisms*, *products*, *extensions*, *substructures*, *expansions*, and *reducts* of general structures. We then introduce the *syntax* of first-order logic (i.e., how do first-order formulas look like?) and finally the *semantics* of first-order logic (i.e., what is the *meaning* of a first-order formula?). Then central definition here is the definition of a *model* of a first-order sentence or a first-order theory (which is simply a set of first-order sentences). We also discuss plenty of examples of structures, sentences, theories, and their models (Chapter 3).

We then present the first-order theory of the Zermelo–Fraenkel axioms of set theory (ZF) which might be extended by the axiom of choice (ZFC). We illustrate how to work within ZFC and define ordinal and cardinals (Chapter 4).

The concept of a *formal proof*, introduced in Chapter 5, consists of two parts: one part consists of a set of easy-to-check *logical axioms*; it will be easy to see that they are *valid*, that is, every structure is a model of those axioms. The second part is a description of how to deduce new valid sentences from sentences that are already known to be valid. The key property of the notion of a formal proof is that *all* statements that are valid can be deduced with the system, and that the deduction step is simple so that it can be performed by a computer. Also in Chapter 5 we present an important consequence of the completeness theorem, namely the compactness theorem.

Chapters 6 and 7 explore the limits of the formal method. We first introduce μ-recursion and prove that these are precisely the functions that can be computed by a Turing machine. There are problems that cannot be solved by a Turing machine; the so-called *Halting problem* is one of those. It will be an easy consequence that Peano arithmetic is incomplete. The proof of Gödel's first incompleteness theorem (in a strengthening due to Rosser) and then Gödel's second incompleteness theorem requires further work.

We assume familiarity with basic (*'naive'*) set theory. The set $\{0, 1, 2, \dots\}$ of natural numbers is denoted by $\mathbb{N}$.

The text contains 96 exercises; the ones with a star are harder.

CHAPTER 2

Propositional Logic

Propositional logic can be seen as a part of first-order logic, but is much simpler and serves as a gentle introduction to the subject.

2.1. Propositional Formulas

A basic principle in logic is the separation of *syntax* and *semantics*. The syntax of a logic specifies how we may combine logical symbols to arrive at well-formed expressions. The *semantics* of a logic specifies how to assign a meaning to these expressions.

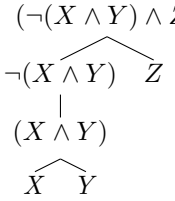
2.1.1. Syntax. The symbols of propositional logic are the symbols $\wedge$ (*conjunction*, pronounced *and*), $\neg$ (*negation*, pronounced *not*), $\top$ (*true*), the brackets '(' and ')', and variable symbols. Typically we will use X, Y, Z, or X_1, $X_2, \ldots$ as variable symbols, but the choice of the variable symbols does not matter (except that they should be distinct from the other logical symbols). The syntax of propositional logic is defined recursively. We define *propositional formulas* as follows.

(1) $\top$ is a propositional formula;
(2) all variables are propositional formulas;
(3) if ϕ is a propositional formula, then $\neg\phi$ is also a propositional formula.
(4) if ϕ_1 and ϕ_2 are propositional formulas, then $(\phi_1 \wedge \phi_2)$ is a propositional formula.

For example,

$$(\neg(X \wedge Y) \wedge Z) \tag{1}$$

is a propositional formula over the variables X, Y, Z. The outermost brackets may be omitted; that is, we may write $\neg(X \wedge Y) \wedge Z$ for the formula in (1). It is easy to see (see the exercises) that every propositional formula is of precisely *one* of the forms (1)-(4) in the recursive definition of propositional formulas. So we can associate to each propositional formula a unique *syntax tree* (which typically is depicted to grow from top to bottom); we do not go further into details, but rather illustrate the syntax tree of (1):



Exercises.

(1) Show that every propositional formula is of precisely *one* of the forms (1)-(4) in the recursive definition of propositional formulas. Hint: first prove by induction on the length of formulas that no proper initial segment of a formula is a formula.

2.1.2. Semantics. Let ϕ be a propositional formula with the variables X_1, X_2, …, X_n. A *(Boolean) assignment for* $X_1, \dots, X_n$ is a function $s\colon \{X_1, \dots, X_n\} \to \{0, 1\}$.

REMARK 2.1.1. Here, 0 stands for *false* and 1 stands for *true*. We say that s *satisfies* ϕ if, informally, ϕ 'evaluates to true' if we replace X_i, for $i \in \{1, \dots, n\}$, by $s(X_i)$. Of course, all of this needs to be defined formally, and what I wrote in this paragraph is just to guide you when reading the formal definition below.

Formally, we define the satisfaction relation inductively over the structure of propositional formulas.

(1) s satisfies $\top$ (so $\top$ stands for *true*).
(2) s satisfies a variable X_i, for $i \in \{1, \dots, n\}$, if $s(X_i) = 1$.
(3) s satisfies $\neg\phi$ if s does *not* satisfy ϕ.
(4) s satisfies $(\phi_1 \wedge \phi_2)$ if s satisfies ϕ_1 and if s satisfies ϕ_2.

Let ϕ be a propositional formula with variables $X_1, \dots, X_n$. Then ϕ is called

- *satisfiable* if there exists an assignment $s\colon \{X_1, \dots, X_n\} \to \{0, 1\}$ that satisfies ϕ, and *unsatisfiable* otherwise.
- *a tautology* (or *tautological*) if every assignment $s\colon \{X_1, \dots, X_n\} \to \{0, 1\}$ satisfies ϕ.

Clearly, ϕ is unsatisfiable if and only if $\neg\phi$ is a tautology.

Propositional formulas can be used to describe Boolean operations. Recall that an *operation of arity* $n \in \mathbb{N}$ *over a set* A is a function from A^n to A. A *Boolean operation* is an operation over $\{0, 1\}$. Let ϕ be a propositional formula with variables $X_1, \dots, X_n$. Then ϕ describes the Boolean operation that maps $(a_1, \dots, a_n) \in \{0, 1\}^n$ to 1 if the map $X_i \mapsto a_i$ satisfies ϕ, and to 0 otherwise. Note that for this definition it is important that we consider a fixed tuple of variables (so that we know which argument of the Boolean operation corresponds to which variable).

We use $\bot$ as a shortcut for $\neg\top$ (which stands for *false*). If ϕ_1 and ϕ_2 are propositional formulas, then $\phi_1 \vee \phi_2$ (*disjunction*) is a shortcut for $\neg(\neg\phi_1 \wedge \neg\phi_2)$. The Boolean operation described by the propositional formulas $X_1 \wedge X_2$ and $X_1 \vee X_2$ are given by the following tables.

$\wedge$	0	1
0	0	0
1	0	1

$\vee$	0	1
0	0	1
1	1	1

There are other important shortcuts.

- $\phi_1 \Rightarrow \phi_2$ (*Implication*) is a shortcut for $\neg\phi_1 \vee \phi_2$. We may read $\phi_1 \Rightarrow \phi_2$ as *'if ϕ_1 (is true), then ϕ_2 (is true)'*.
- $\phi_1 \Leftrightarrow \phi_2$ (*equivalence*) is a shortcut for $\big((\phi_1 \Rightarrow \phi_2) \wedge (\phi_2 \Rightarrow \phi_1)\big)$. We may read $\phi_1 \Leftrightarrow \phi_2$ as *'ϕ_1 (is true) if and only if ϕ_2 (is true)'*.

The Boolean operations described by $X_1 \Rightarrow X_2$ and by $X_1 \Leftrightarrow X_2$ are shown on the right.

$\Rightarrow$	0	1
0	1	1
1	0	1

$\Leftrightarrow$	0	1
0	1	0
1	0	1

Let ϕ_1 and ϕ_2 be propositional formulas with variables $X_1, \dots, X_n$. Then ϕ_1 and ϕ_2 are called *equivalent* if they describe the same Boolean operation. Note that for all propositional formulas ϕ_1, ϕ_2, ϕ_3, the formulas $(\phi_1 \wedge (\phi_2 \wedge \phi_3))$ and $((\phi_1 \wedge \phi_2) \wedge \phi_3)$ are equivalent. Since we usually only care about the Boolean operation that is described by a propositional formula, we sometimes omit the brackets and simply write $\phi_1 \wedge \phi_2 \wedge \phi_3$ instead of the propositional formulas above.

If $I = \{i_1, \dots, i_n\}$ is a finite set and ϕ_i is a propositional formula for every $i \in I$, then we write $\bigwedge_{i\in I} \phi_i$ instead of $\phi_{i_1} \wedge \dots \wedge \phi_{i_n}$. It will be convenient to also use this expression if $n = 0$ (i.e., for $I = \emptyset$), in which case the expression equals $\top$ by definition. Similarly, we write $\bigvee_{i\in I} \phi_i$ instead of $\phi_{i_1} \vee \dots \vee \phi_{i_n}$; the empty disjunction equals $\bot$ by definition.

Exercises.

(2) Let ϕ_1 and ϕ_2 be propositional formulas with variables $X_1, \dots, X_n$. Show that ϕ_1 and ϕ_2 are equivalent if and only if $\phi_1 \Leftrightarrow \phi_2$ is a tautology.

(3) Show that the following propositional formula is a tautology.

$$(X \Rightarrow Y) \Leftrightarrow (\neg Y \Rightarrow \neg X) \qquad \text{(Contraposition)}$$

(4) There are three doors to different rooms; we know that there is a treasure in one of the rooms and a dragon in each of the other two.
- The first door is labelled *'A dragon is in this room'*.
- The second door is labelled *'A treasure is in this room'*.
- The third door is labelled *'A dragon is in the second room'*.

We also know that at most one of the three labels is true.

Task: Model this problem using a propositional formula. Explain the intended meaning for the propositional variables that you are using, and how you translate the given information into a formula. Which room contains the treasure?

2.1.3. Disjunctive and conjunctive normal form. Every propositional formula describes a Boolean operation; conversely, every Boolean operation can be described by a propositional formula.

PROPOSITION 2.1.2. *Every Boolean operation $f\colon \{0,1\}^n \to \{0,1\}$ can be described by a propositional formula with variables $X_1, \dots, X_n$.*

PROOF. For all $a_1, \dots, a_n \in \{0,1\}$ we have that $f(a_1, \dots, a_n) = 1$ if and only if the assignment $X_i \mapsto a_i$ satisfies

$$\bigvee_{\substack{b_1,\dots,b_n \in \{0,1\},\\ f(b_1,\dots,b_n)=1}} \left(\bigwedge_{i\in\{1,\dots,n\} \text{ with } b_i=1} X_i \wedge \bigwedge_{i\in\{1,\dots,n\} \text{ with } b_i=0} \neg X_i \right). \qquad \square$$

EXAMPLE 1. Let $f\colon \{0,1\}^3 \to \{0,1\}$ be the Boolean operation that is given by the table on the right.

a_1	0	0	0	0	1	1	1	1
a_2	0	0	1	1	0	0	1	1
a_3	0	1	0	1	0	1	0	1
$f(a_1, a_2, a_3)$	0	1	0	1	0	0	0	0

Then f is described by the propositional formula

$$(\neg X_1 \wedge \neg X_2 \wedge X_3) \vee (\neg X_1 \wedge X_2 \wedge X_3). \qquad \triangle$$

The proof of Proposition 2.1.2 shows something stronger: we see that every Boolean operation can be described by a disjunction of conjunctions of variables or negated variables. This specific form of a propositional formula is called *disjunctive normal form (DNF)*.

It can also be shown that every Boolean operation $f\colon \{0,1\}^n \to \{0,1\}$ can be described by a propositional formula in *conjunctive normal form (CNF)*, which is a conjunction of disjunctions of variables or negated variables. To see this, we apply

Proposition 2.1.2 to the Boolean operation which returns 0 if f returns 1, and returns 1 if f returns 0. We already know that there exists a propositional formula ϕ in DNF that describes this new Boolean operation. The clearly $\neg\phi$ describes f. Note that $\neg\phi$ is not yet in conjunctive normal form; but using the following equivalences and induction, one can easily find a formula which is in CNF and equivalent to $\neg\phi$.

$$\neg(\neg\phi) \Leftrightarrow \phi \tag{2}$$

$$\neg(\phi_1 \vee \phi_2) \Leftrightarrow (\neg\phi_1 \wedge \neg\phi_2) \tag{3}$$

$$\neg(\phi_1 \wedge \phi_2) \Leftrightarrow (\neg\phi_1 \vee \neg\phi_2) \tag{4}$$

The tautologies in (3) and in (4) are also called *De Morgan's laws*. The same idea can be used to transform CNF into DNF. There is an alternative approach for converting between CNF and DNF, based on using the distributive laws given below (we leave the details to the reader)

$$\phi_1 \wedge (\phi_2 \vee \phi_3) \Leftrightarrow ((\phi_1 \wedge \phi_2) \vee (\phi_1 \wedge \phi_3)) \qquad (\wedge\text{-Distributivity})$$

$$\phi_1 \vee (\phi_2 \wedge \phi_3) \Leftrightarrow ((\phi_1 \vee \phi_2) \wedge (\phi_1 \vee \phi_3)) \qquad (\vee\text{-Distributivity}).$$

The CNF is particularly useful; often, propositional formulas are assumed to be in CNF. So there is specialised terminology: the conjuncts of a propositional formula in CNF are also called *clauses*. The disjuncts of a clause are called *literals*; they are either variables or negated variables. In the first case, the literal is called *positive* and in the latter case *negative* (correspondingly, we also speak about positive and negative occurrences of variables in clauses). A clause is viewed as a set of literals (here we do not lose essential information, because changing the order or the number of occurrences of the literals leads to equivalent formulas).

Exercises.

(5) Find a propositional formula in CNF that describes the Boolean operation from Example 1.

(6) Show that *every* propositional formula in CNF that is equivalent to

$$(X_1 \wedge Y_1) \vee (X_2 \wedge Y_2) \vee \cdots \vee (X_n \wedge Y_n)$$

has at least 2^n clauses.

2.2. The Satisfiability Problem

The satisfiability problem is a computational problem, often denoted by SAT: for a given a proposition formula ϕ, the task is to determine whether ϕ is satisfiable. Many combinatorial problems can be coded into the satisfiability problem. Note that the problem of deciding whether a given propositional formula ϕ is a tautology can be reduced to the satisfiability problem: ϕ is a tautology if and only if $\neg\phi$ is unsatisfiable.

The satisfiability problem can be solved by a computer: we first list exhaustively all functions s from the variables $X_1, \ldots, X_n$ of ϕ to $\{0, 1\}$; for each of the functions we can verify easily whether s satisfies ϕ, by recursion on the syntax tree of ϕ from Section 2.1.1, evaluating the formula following the rules specified in Section 2.1.2. Unfortunately, the running time of the procedure is exponential, and the brute-force algorithm above is impractical already for inputs of moderate size. There is no algorithm known that can solve SAT in polynomial time in the size of the formula (i.e., the number of symbols of the formula).

Remark 2.2.1. The naive algorithm for SAT that we have presented above shows that the satisfiability problem belongs to the complexity class NP; see **[20]** for an introduction to computational complexity.

EXAMPLE 2. Is there a graph G with 43 vertices that contains neither a *clique with 5 vertices* nor an *independent set with 5 vertices*? A *clique* in a graph is a subset S of the vertices such every vertex in S is adjacent in the graph to any other vertex in S. An *independent set* in a graph is a subset S of the vertices such that every vertex in S is not adjacent in the graph to any other vertex in S. This problem can be formulated as an instance of SAT as follows. We use the variables $X_{1,2}, X_{1,3}, \ldots, X_{42,43}$ and consider the following propositional formula.

$$\bigwedge_{S\subseteq\{1,\ldots,43\},|S|=5} \left(\bigvee_{i,j\in S, i\neq j} X_{i,j} \wedge \bigvee_{i,j\in S, i\neq j} \neg X_{i,j} \right)$$

If the given propositional formula is satisfiable, then the answer to the question above is *yes*. This question is a famous open problem in Ramsey theory. The algorithm for the satisfiability problem mentioned above needs to examine 2^n cases, where n is the number of variables; for the above problem, $n = 43 \cdot 42/2 = 903$. IBM is currently working on a 100 GHz CPU, that is, the CPU has 100 billion clock cycles per second. Even if we could solve one case in one clock cycle, the computer would need $2^{903}/(31536000 * 10^{11}) > 10^{\lg 2 \cdot 903}/10^{19} > 10^{0,301 \cdot 903 - 19} > 10^{252}$ years. For comparison: the age of the universe is estimated to be $13.79 \cdot 10^9$ years. △

REMARK 2.2.2. The class of all problems that can be solved in polynomial time in the size of the input is denoted by P. So it is an open problem whether SAT is in P. It can be shown that the satisfiability problem belongs to the hardest problems in NP; it is *NP-complete*. If there is a polynomial-time algorithm for SAT, then P would be equal to NP; most researchers believe that P and NP are distinct. The P=NP question is one of the famous Millenium problems and believed to be one of the most difficult questions in mathematics.

Exercises.

(7) Show that there exists a polynomial-time algorithm that decides the satisfiability problem for propositional formulas given in DNF.
(8) We have seen that every propositional formula is equivalent to a formula in DNF, and that the satisfiability problem for propositional formulas can be solved in polynomial time. Why doesn't this provide a solution to the Millenium problem about a polynomial-time algorithm for SAT?

2.2.1. Propositional Resolution. Propositional resolution provides a conceptually simple algorithm to solve the satisfiability problem; however, similarly as the brute-force algorithm that we have seen in the previous section, it is not an efficient algorithm, neither in theory nor in practice.

To formulate propositional resolution, we assume that the propositional formula ϕ is given in conjunctive normal form, as a set of clauses. We now apply the following inference rule:

$$\frac{\{L_1, \ldots, L_n, X\}, \quad \{\neg X, M_1, \ldots, M_k\}}{\{L_1, \ldots, L_n, M_1, \ldots, M_k\}} \tag{5}$$

Such rules are to be read as follows: suppose ϕ contains the two clauses that are written above the line. Then we *add* the clause below the line to ϕ. Note that in order to apply the rule (5), one of the clauses must contain a variable X, and the other must contain the same variable, but negated, $\neg X$. Also note that the formula obtained from ϕ by adding the new clause is equivalent to ϕ: if s is a satisfying assignment for ϕ, and $s(X) = 1$, then s must satisfy the literal M_i for some $i \leq k$ because s satisfies the right clause above the line. If $s(X) = 0$, then s must satisfy

the literal L_i for some $i \leq n$ because s satisfies the left clause above the line. In both cases it follows that s satisfies the clause below the line.

Note that if we apply the resolution rule exhaustively to derive new clauses, we eventually arrive at a set of clauses where no new clauses can be derived, because for a fixed finite set of variables there are only finitely many clauses that can be formed. Such sets of clauses will be called *closed under resolution*. If ϕ contains an empty clause, then clearly ϕ is unsatisfiable (this property of resolution is sometimes referred to as *soundness* of resolution). The following lemma states a converse to this fact (this property is sometimes referred to as *completeness* of resolution).

THEOREM 2.2.3. *Let ϕ be a propositional formula in CNF which is closed under resolution, and does not contain the empty clause. Then ϕ is satisfiable.*

To prove the theorem we need some preparations. If ϕ is a propositional formula in CNF and L a literal of ϕ, then we write ϕ_L for the formula obtained from ϕ by dropping all clauses that contain L and removing all occurrences of the literal $\neg L$ in all the remaining clauses (here and in the following we identify $\neg(\neg L)$ with L).

LEMMA 2.2.4. *Let ϕ be a formula in CNF that is closed under resolution and let L be a literal of ϕ. Then ϕ_L is closed under resolution, too.*

PROOF. Let C_1 and C_2 be two clauses in ϕ_L that resolve to C. We have to show that C is a clause of ϕ_L. Note that neither C_1 nor C_2 contains the literal $\neg L$, so neither does C. If C_1 and C_2 were already clauses of ϕ, then so is C, and hence C is a clause of ϕ_L. Otherwise, C_1 was obtained from the clause $C_1 \cup \{\neg L\}$ of ϕ or C_2 was obtained from the clause $C_2 \cup \{\neg L\}$ of ϕ. Since ϕ is closed under resolution it contains $C \cup \{\neg L\}$, and hence ϕ_L contains the clause C. □

PROOF OF THEOREM 2.2.3. We prove the statement by induction on the number of variables that appear in ϕ. Suppose that ϕ contains a clause that consists of a single negative literal $\{\neg X\}$. Then $\phi_{\neg X}$ cannot contain the empty clause: otherwise, ϕ must contain the clause $\{X\}$, and we could have derived the empty clause from the clauses $\{X\}$ and $\{\neg X\}$ by resolution, contradicting the assumptions that ϕ is closed under resolution and does not contain the empty clause. By Lemma 2.2.4, the formula $\phi_{\neg X}$ is closed under resolution, but has less variables, so by the inductive assumption it has a satisfying assignment s. Then the extension of s given by $X \mapsto 0$ is a satisfying assignment to the original formula.

So we assume in the following that ϕ does not contain clauses of the form $\{\neg X\}$. If all variables appear negatively in ϕ, then the function that maps all variables to 0 is a satisfying assignment and we are done. So ϕ contains a clause with a positive literal, say X. Then ϕ_X does not contain the empty clause because of the assumption that ϕ does not contain clauses of the form $\neg X$. Moreover, ϕ_X is closed under resolution by Lemma 2.2.4, and has fewer variables, so it has a satisfying assignment s by the inductive assumption. Then the extension of s defined by $X \mapsto 1$ is a satisfying assignment to the original formula. □

Exercises.

(9) A propositional formula in CNF is called *bijunctive* if all clauses have at most two literals. Show that resolution can be used to obtain a polynomial-time algorithm for the satisfiability problem for bijunctive formulas.

2.2.2. The DPLL Algorithm. In this section we present a third algorithm for SAT, the Davis-Putnam-Logemann-Loveland (DPLL) algorithm, which is the basis for almost all modern SAT solvers. Such SAT solvers have reached remarkably good

performance on randomly generated instances of SAT and on large instances of SAT that arise in industry (but they do not perform sufficiently well on difficult hand-crafted instances as the one in Example 2). In essence, the DPLL algorithm is closely related to propositional resolution. Again we assume that the propositional formula ϕ is given in conjunctive normal form. We use an important inference step, called *unit propagation* (which you might recognise from the completeness proof of the resolution procedure).

Unit Propagation. If ϕ contains a clause of the form $\{X\}$, i.e., a clause that just contains one positive literal, then X must take value 1 in every solution to ϕ. Hence, all clauses that contain the literal X can be removed, and all literals of the form $\neg X$ can be removed from the remaining clauses. In this way we have eliminated the variable X. Similarly, unit propagation eliminates X if ϕ contains a clause of the form $\{\neg X\}$.

Pure Literal Elimination. If a variable X only occurs positively in ϕ, then we may remove all clauses that contain X, and solve the resulting instance recursively. If the resulting instance has a satisfying assignment, we obtain a satisfying assignment to ϕ extending α by $X \mapsto 1$. Similarly, we can simplify the instance if X only occurs negatively in ϕ (in these cases we say that X is *pure* in ϕ).

The overall algorithm can be found in Figure 2.1.

DPLL(ϕ)
// Input: A propositional formula ϕ in CNF.
// Output: *yes* if ϕ is satisfiable, *no* otherwise.
While there are unit clauses or pure literals in ϕ:
    apply unit propagation and pure literal elimination.
If ϕ contains an empty clause then return *no*.
If ϕ contains no clauses then return *yes*.
Pick a literal L from some clause.
Return *yes* if DPLL($\phi \cup \{\{\neg L\}\}$) returns *yes* or DPLL($\phi \cup \{\{L\}\}$) returns *yes*, otherwise return *no*.

FIGURE 2.1. An algorithm for the satisfiability problem.

Exercises.

(10) A propositional formula in CNF is called *Horn* if each clause of the formula contains at most one positive literal. Show that repeatedly applying Unit Propagation is a sound and complete procedure to test satisfiability of propositional Horn formulas. Show that the running time of this algorithm is polynomial.

CHAPTER 3

First-order Logic

In this section we introduce *first-order logic*. First-order logic can be seen as an extension of propositional logic by the ability to talk about functions and relations of so-called *structures* (sometimes also called *first-order structures*) and in particular the ability to *quantify* over the elements of the structure. There are many other logics: as the name suggests, there is for instance *second-order logic*, where it is also allowed to quantify over subsets of elements of the domain, or even over relations.

We first introduce structures (Section 3.1). Then we introduce the syntax and semantics of first-order logic. The syntax of first-order logic has two 'layers': we first have to introduce terms (Section 3.2.1), and then formulas (Section 3.2.1).

3.1. First-order Structures

3.1.1. Signatures. A *signature* τ is a set of relation and function symbols, each equipped with an *arity* $k \in \mathbb{N}$. Examples of important signatures:

- $\tau_{\text{Graph}} = \{E\}$
- $\tau_{\text{AGroup}} = \{+, -, 0\}$
- $\tau_{\text{Ring}} = \tau_{\text{AG}} \cup \{1, \cdot\}$
- $\tau_{\text{Group}} = \{\circ, ^{-1}, e\}$
- $\tau_{\text{LO}} = \{<\}$
- $\tau_{\text{OGroup}} = \tau_{\text{Group}} \cup \tau_{\text{LO}}$
- $\tau_{\text{Arithm}} = \{0, s, +, \cdot, <\}$
- $\tau_{\text{Set}} = \{\in\}$

Here, $+$, $\circ$, and $\cdot$ are binary function symbols, $-$, $^{-1}$, and s are unary function symbols, e, 0, and 1 are 0-ary function symbols, and E, $\in$, and $<$ are binary relation symbols.

3.1.2. Structures. A *τ-structure* $\underline{A}$ is a non-empty[1] set A (called the *domain*, or the *base set*, or the *universe* of $\underline{A}$) together with

- a relation $R^{\underline{A}} \subseteq A^k$ for each k-ary relation symbol $R \in \tau$. Here we allow the case $k = 0$, in which case $R^{\underline{A}}$ is either empty or of the form $\{()\}$, i.e, the set consisting of the empty tuple;
- a function $f^{\underline{A}} \colon A^k \to A$ for each k-ary function symbol $f \in \tau$; here we also allow the case $k = 0$ to model constants from A.

Unless stated otherwise, $A, B, C, \ldots$ denotes the domain of the structure $\underline{A}$, $\underline{B}$, $\underline{C}$, $\ldots$, respectively. We sometimes write $(A; R_1^{\underline{A}}, R_2^{\underline{A}}, \ldots, f_1^{\underline{A}}, f_2^{\underline{A}}, \ldots)$ for the structure $\underline{A}$ with relations $R_1^{\underline{A}}, R_2^{\underline{A}}, \ldots$ and functions $f_1^{\underline{A}}, f_2^{\underline{A}}, \ldots$ Well-known examples of structures are $(\mathbb{Q}; +, -, 0, 1, \cdot, <)$, $(\mathbb{C}; +, -, 0, 1, \cdot, <)$, $(\mathbb{Z}; 0, s, +, \cdot, <)$, etc. In cases where the reference to the structure $\underline{A}$ is clear we are sometimes sloppy with the notation and write f instead of $f^{\underline{A}}$. We say that a structure is infinite if its domain is infinite.

[1] In some texts, structures with empty sets are allowed. Occasionally, this makes a difference, but it is always straightforward to translate between results in one setting to results in the other setting.

EXAMPLE 3. A *(simple, undirected) graph* is a pair (V, E) consisting of a set of *vertices* V and a set of *edges* $E \subseteq \binom{V}{2}$, that is, E is a set of 2-element subsets of V. Graphs can be modelled as relational structures $\underline{G}$ using a signature that contains a single binary relation symbol R, putting $G := V$ and adding (u, v) to $R^{\underline{G}}$ if $\{u, v\} \in E$. If we insist that a structure with this signature satisfies $(x, y) \in R^{\underline{G}} \Rightarrow (y, x) \in R^{\underline{G}}$ and $(x, x) \notin R^{\underline{G}}$, then we can associate to such a structure an undirected graph and obtain a bijective correspondence between undirected graphs with vertices V and structures $\underline{G}$ with domain V as described above. △

EXAMPLE 4. A *group* is a structure $\underline{G}$ with the signature $\tau_{\mathrm{Group}} = \{\circ, ^{-1}, e\}$ such that for all $x, y, z \in G$ we have that

- $x \circ (y \circ z) = (x \circ y) \circ z$,
- $e \circ x = x \circ e = x$,
- $x \circ x^{-1} = x^{-1} \circ x = e$.

A group is Abelian if it additionally satisfies $x \circ y = y \circ x$ for all $x, y \in G$. For Abelian groups, one often uses the signature $\tau_{\mathrm{AGroup}} = \{+, -, 0\}$ instead of τ_{Group}. △

3.1.3. Substructures. A τ-structure $\underline{A}$ is a *substructure* of a τ-structure $\underline{B}$ iff

- $A \subseteq B$,
- for each $R \in \tau$, and for all tuples $\bar{a}$ from A, $\bar{a} \in R^{\underline{A}}$ iff $\bar{a} \in R^{\underline{B}}$, and
- for each $f \in \tau$ we have that $f^{\underline{A}}(\bar{a}) = f^{\underline{B}}(\bar{a})$.

In this case, we also say that $\underline{B}$ is an *extension* of $\underline{A}$. Substructures $\underline{A}$ of $\underline{B}$ and extensions $\underline{B}$ of $\underline{A}$ are called *proper* if the domains of $\underline{A}$ and $\underline{B}$ are distinct.

Note that for every subset S of the domain of $\underline{B}$ there is a unique smallest substructure of $\underline{B}$ whose domain contains S, which is called the *substructure of $\underline{B}$ generated by S*, and which is denoted by $\underline{B}[S]$.

EXAMPLE 5. When we view a graph as an $\{E\}$-structure $\underline{G}$, then a subgraph is not necessarily a substructure of $\underline{G}$. In graph theory, the substructures of $\underline{G}$ are called *induced subgraphs*: the difference is that in an induced subgraph (V', E') of (V, E) the edge set must be of the form $E' := E \cap \binom{V'}{2}$ instead of an arbitrary subset of it. △

EXAMPLE 6. Due to the choice of our signature τ_{Group}, the subgroups of $\underline{G}$ are precisely the substructures of $\underline{G}$ as defined above. △

3.1.4. Expansions and reducts. Let σ, τ be signatures with $\sigma \subseteq \tau$. If $\underline{A}$ is a σ-structure and $\underline{B}$ is a τ-structure, both with the same domain, such that $R^{\underline{A}} = R^{\underline{B}}$ for all relations $R \in \sigma$ and $f^{\underline{A}} = f^{\underline{B}}$ for all function symbols $f \in \sigma$, then $\underline{A}$ is called a *reduct* of $\underline{B}$, and $\underline{B}$ is called an *expansion* of $\underline{A}$.

EXAMPLE 7. A *linearly ordered group* is a τ_{OGroup}-structure $\underline{G}$ whose τ_{Group}-reduct is a group and such that $<^{\underline{G}}$ is a linear order which is *translation-invariant*, i.e., $a \leq b$ implies $c \circ a \leq c \circ b$ for all $a, b, c \in G$, where $x \leq y$ stands for $x <^{\underline{G}} y$ or $x = y$. △

DEFINITION 3.1.1. Let $\underline{A}$ be a τ-structure and $B \subseteq A$. We write $\underline{A}_B$ for the expansion of $\underline{A}$ with the signature $\tau \cup \{c_b \mid b \in B\}$, where c_b is a new constant symbol for every element $b \in B$, such that $c_b^{\underline{A}_B} := b$.

3.1.5. Homomorphisms. In the following, let $\underline{A}$ and $\underline{B}$ be τ-structures. A *homomorphism* h from $\underline{A}$ to $\underline{B}$ is a mapping from A to B that *preserves* each function and each relation for the symbols in τ; that is,

- if $(a_1, \ldots, a_k)$ is in $R^{\underline{A}}$, then $(h(a_1), \ldots, h(a_k))$ must be in $R^{\underline{B}}$;
- $f^{\underline{B}}(h(a_1), \ldots, h(a_k)) = h(f^{\underline{A}}(a_1, \ldots, a_k))$.

A homomorphism from $\underline{A}$ to $\underline{B}$ is called a *strong homomorphism* if it also preserves the complements of the relations from $\underline{A}$. Injective strong homomorphisms are called *embeddings*. Surjective embeddings are called *isomorphisms*.

EXAMPLE 8. Group-, ring-, and field homomorphisms. △

EXAMPLE 9. The *graph colorability problem* is an important problem in discrete mathematics, with many applications in theoretical computer science (it can be used to model e.g. frequency assignment problems). As a computational problem, the graph n-colorability problem has the following form.

Given: a finite graph $G = (V, E)$.
Question: can we colour the vertices of G with n colours such that adjacent vertices get different colours?

The n-colorability problem can be formulated as a graph homomorphism problem: is there a homomorphism from G to

$$K_n := (\{1, \dots, n\}; E^{K_n}) \text{ where } E^{K_n} := \{(u, v) \in V^2 \mid u \neq v\}.$$

We also refer to these homomorphisms as *proper colourings* of G, and say that G is *n-colourable* if such a colouring exists. The *chromatic number* $\chi(G)$ of G is the minimal natural number $n \in \mathbb{N}$ such that G is n-colourable. For example, the chromatic number of K_n is n. △

EXAMPLE 10. We present a concrete instance of a mathematical colouring problem. Let (V, E) the unit distance graph on $\mathbb{R}^2$, i.e., the graph has the vertex set $V := \mathbb{R}^2$ (we imagine the nodes as the points of the Euclidean plane) and edge set

$$E := \left\{(x, y) \in V^2 \;\middle|\; |x - y| = 1\right\}.$$

In other words, two points are linked by an edge if they have distance one. What is the chromatic number of this graph? △

We will see later (in Corollary 5.5.4, as a consequence of the compactness theorem) that a graph G is k-colourable if and only if all finite subgraphs of G are k-colourable. The problem to determine the chromatic number χ of the graph in Example 10 is known as the Hadwiger-Nelson problem. It is known that $\chi \leq 7$. We have seen that $4 \leq \chi$. In April 11, 2018, Aubrey de Grey announced a proof that $5 \leq \chi$. The precise value of $\chi \in \{5, 6, 7\}$ is not known.

Exercises.

(11) Find a signature τ for vector spaces and describe how a vector space may be viewed as a τ-structure. Your definition should have the property that homomorphisms between the structures you consider correspond precisely to linear maps.
(12) Let $\underline{A}$ and $\underline{B}$ be τ-structures and suppose that $G \subseteq A$ generates $\underline{A}$, i.e., $\underline{A} = \underline{A}[G]$. Then every homomorphism $h\colon \underline{A} \to \underline{B}$ is determined by its values on G.
(13) Let $\underline{A}$ and $\underline{B}$ be τ-structures, and suppose that τ has no relation symbols. Show that every bijective homomorphism from $\underline{A}$ to $\underline{B}$ is an isomorphism.
(14) Find an example of a τ-structure $\underline{A}$ and a bijective homomorphism from $\underline{A}$ to $\underline{A}$ which is not an isomorphism.

3.2. Formulas, Sentences, Theories

To define the syntax of first-order logic, we first introduce terms, then (first-order) formulas and (first-order) sentences, and finally (first-order) theories.

3.2.1. Terms. Let τ be a signature. In this section we will see how to use the function symbols in τ to build *terms*.

DEFINITION 3.2.1. A *(τ-)term* is defined inductively:

- constants from τ are τ-terms;
- variables $x_0, x_1, \dots$ are τ-terms;
- if $t_1, \dots, t_k$ are τ-terms, and $f \in \tau$ has arity k, then $f(t_1, \dots, t_k)$ is a τ-term.

Note that item (1) is a special case of item (3). If t is term, then we write $t(x_1, \dots, x_n)$ if all variables that appear in t come from $\{x_1, \dots, x_n\}$; we do not require that each variable x_i appears in t, but every variables that appears in t must be of the form x_i for some $i \leq n$.

EXAMPLE 11. Well-known examples of terms are polynomials over a ring $\underline{R}$: they are terms over a signature that contains a binary symbols $+, \cdot$ for addition and multiplication, together with a constant symbol for each element of R. However, note that for $+$ and $\cdot$ we usually use in-fix notation, i.e., we write $t_1 + t_2$ instead of $+(t_1, t_2)$. △

EXAMPLE 12. Propositional formulas can be viewed as $\{\wedge, \neg, \top\}$-terms. △

3.2.2. Semantics of terms. In this section we describe how τ-terms over a given τ-structure describe functions (in the same way as polynomials describe polynomial functions over a given ring).

Let $\underline{A}$ be a τ-structure and let $x_1, \dots, x_n$ be distinct variables. Every τ-term $t(x_1, \dots, x_n)$ describes an operation $t^{\underline{A}}\colon A^n \to A$ as follows:

(1) if t equals $c \in \tau$ then $t^{\underline{A}}$ is the operation $(a_1, \dots, a_n) \mapsto c^{\underline{A}}$;
(2) if t equals x_i then $t^{\underline{A}}$ is the operation $(a_1, \dots, a_n) \mapsto a_i$;
(3) if t equals $f(t_1, \dots, t_k)$ for a k-ary $f \in \tau$ and τ-terms $t_1, \dots, t_k$, then $t^{\underline{A}}$ is the operation

$$(a_1, \dots, a_n) \mapsto f^{\underline{A}}\big(t_1^{\underline{A}}(a_1, \dots, a_n), \dots, t_k^{\underline{A}}(a_1, \dots, a_n)\big).$$

The function described by t is also called the *term function* of t (with respect to $\underline{A}$).

Exercises.

(15) Let $\underline{B}$ be a τ-structure and $G \subseteq B$. Let $\underline{A}$ be the substructure of $\underline{B}$ generated by G. Show that for every element $a \in A$ there exists a τ-term $t(x_1, \dots, x_n)$ and elements $g_1, \dots, g_n \in G$ such that $t^{\underline{B}}(g_1, \dots, g_n) = a$.

3.2.3. Formulas and sentences. Let τ be a signature. The relation symbols in the signature τ did not play any role when defining τ-terms, but they become important when defining *τ-formulas*. Moreover, the equality symbol $=$ is 'hard-wired' into first-order logic; we can use it to create formulas by equating terms. Finally, we can combine formulas using Boolean connectives, and quantify over variables.

DEFINITION 3.2.2. An *atomic τ-formula* is an expression of the form

- $\top$;
- $t_1 = t_2$ where t_1 and t_2 are τ-terms;
- $R(t_1, \dots, t_k)$ where $t_1, \dots, t_k$ are τ-terms and $R \in \tau$ is a k-ary relation symbol.

Then *τ-formulas* are defined inductively as follows:

- atomic τ-formulas are τ-formulas;
- if ϕ is a τ-formula, then $\neg\phi$ is a τ-formula *(negation)*;
- if ϕ and ψ are τ-formulas, then $\phi \wedge \psi$ is a τ-formula *(conjunction)*;

- if ϕ is a τ-formula, and x is a variable, then $\exists x.\, \phi$ is a τ-formula *(existential quantification)*.

Note that we may reconstruct from every τ-formula, viewed as a string of symbols, the way how it was built inductively. The purpose of the dot in the formula $\exists x.\phi$ is to increase readability, e.g. in the formula $\exists \mathrm{na}.\, \mathrm{Nu}(\mathrm{na})$, which would otherwise become $\exists \mathrm{na}\mathrm{Nu}(\mathrm{na})$. In formulas of the form $\exists x.\, (\phi \wedge \psi)$ we omit the dot since this does not harm readability.

Atomic formulas and negations of atomic formulas are sometimes called *literals*. Similarly as for terms, we write $\phi(x_1, \ldots, x_n)$ if all non-quantified variables in the formula ϕ come from $\{x_1, \ldots, x_n\}$. A *(first-order) sentence* is a formula without free variables, i.e., all variables that appear in the formula are *bound*, i.e., quantified by some quantifier.

Shortcuts: We use the shortcuts $\bot$, $\vee$, $\Rightarrow$, $\Leftrightarrow$ that we already know from propositional logic. Additionally, we define the following shortcuts:

- *Universal quantification:* $\forall x.\, \phi(x)$ is an abbreviation for $\neg\exists x.\, \neg\phi(x)$
- *Inequality:* $x \neq y$ is an abbreviation for $\neg(x = y)$

If A is a unary relation symbol, then we may write $\exists x \in A.\, \phi$ instead of $\exists x\, (A(x) \wedge \phi)$ and $\forall x \in A.\, \phi$ instead of $\forall x\, (A(x) \Rightarrow \phi)$.

Example 13. Let $\tau = \{R\}$ where R is a binary relation symbol. Then the following formula is an example of a first-order sentence

$$\forall x_1, x_2, x_3 \big((R(x_1, x_2) \wedge R(x_2, x_3)) \Rightarrow R(x_1, x_3) \big). \qquad \triangle$$

3.2.4. Semantics of formulas. So far, we have just introduced the *syntax* of first-order logic, i.e., the shape of formulas and sentences, without discussing what these expressions actually mean. In this section we discuss their *semantics*; the idea is that formulas can be used to define new relations and functions in a given structure, and that sentences can be used to describe properties that a structure might have.

Let $\underline{A}$ is a τ-structure. Then every τ-formula $\phi(x_1, \ldots, x_n)$ describes a relation $\phi^{\underline{A}} \subseteq A^n$ as follows:

- if ϕ equals $\top$ then $\phi^{\underline{A}} = A^n$;
- if ϕ equals $t_1 = t_2$ then $\phi^{\underline{A}}$ is the relation
$$\big\{(a_1, \ldots, a_n) \mid t_1^{\underline{A}}(a_1, \ldots, a_n) = t_2^{\underline{A}}(a_1, \ldots, a_n)\big\};$$
- if ϕ equals $R(t_1, \ldots, t_k)$ then
$$\phi^{\underline{A}} := \big\{(a_1, \ldots, a_n) \mid (t_1^{\underline{A}}(\bar{a}), \ldots, t_k^{\underline{A}}(\bar{a})) \in R^{\underline{A}}\big\};$$
- if ϕ equals $\phi_1 \wedge \phi_2$ then $\phi^{\underline{A}} := \phi_1^{\underline{A}} \cap \phi_2^{\underline{A}}$;
- if ϕ equals $\neg\psi$ then $\phi^{\underline{A}} := A^n \setminus \psi^{\underline{A}}$;
- if ϕ equals $\exists x.\, \psi(x, x_1, \ldots, x_n)$ then

$$\phi^{\underline{A}} := \{(a_1, \ldots, a_n) \mid \text{there exists } a \in A \text{ such that } (a, a_1, \ldots, a_n) \in \psi^{\underline{A}}\}.$$

A relation $R \subseteq A^k$ is called *(first-order) definable (in $\underline{A}$)* if there exists a τ-formula $\phi(x_1, \ldots, x_k)$ such that $R = \phi^{\underline{A}}$; we also say that ϕ *defines* R over $\underline{A}$. We say that two τ-formulas $\phi(x_1, \ldots, x_n)$ and $\psi(x_1, \ldots, x_n)$ are *equivalent* if for every τ-structure $\underline{A}$ we have $\phi^{\underline{A}} = \psi^{\underline{A}}$. For $\phi(x_1, \ldots, x_n)$ we write

$$\underline{A} \models \phi(a_1, \ldots, a_n)$$

instead of $(a_1, \ldots, a_n) \in \phi^{\underline{A}}$. In particular, if ϕ is a sentence, i.e., if $n = 0$, we write $\underline{A} \models \phi$, and say that $\underline{A}$ *satisfies* ϕ, if $() \in \phi^{\underline{A}}$ (that is, if $\phi^{\underline{A}} \neq \emptyset$). Otherwise, we write $\underline{A} \not\models \phi$. Clearly, $\underline{A} \not\models \phi$ if and only if $\underline{A} \models \neg\phi$.

EXAMPLE 14. The following statements about well-known structures follow straightforwardly from the definitions.

- $(\mathbb{Z}; <) \models 0 < 1$
- $(\mathbb{Q}; <) \models \forall x, y \left(x < y \Rightarrow \exists z \left(x < z \wedge z < y\right)\right)$ (density)
- $(\mathbb{Z}; <) \not\models \forall x, y \left(x < y \Rightarrow \exists z \left(x < z \wedge z < y\right)\right)$ △

DEFINITION 3.2.3. Let $\underline{A}$ be a τ-structure. An operation $f\colon A^k \to A$ is called *(first-order) definable* in $\underline{A}$ if the relation $\{(a_0, a_1, \ldots, a_n) \mid a_0 = f(a_1, \ldots, a_n)\}$ is first-order definable in $\underline{A}$; this relation is also called the *graph of* f.

Exercises.

(16) Let $\phi(x_1, \ldots, x_n)$ and $\psi(x_1, \ldots, x_n)$ be τ-formulas. Show that ϕ and ψ are equivalent if and only if $\underline{A} \models \forall \bar{x} \left(\phi(\bar{x}) \Leftrightarrow \psi(\bar{x})\right)$ for all τ-structures $\underline{A}$.

(17) Let $\underline{G}$ be a group. Show that the constant operation $e^{\underline{G}}$ and the unary operation and $(^{-1})^{\underline{G}}$ are definable in $(G; \circ^{\underline{G}})$.

(18) A formula is in *prenex normal form* if it is of the form $Q_1 x_1 \ldots Q_n x_n . \phi$ where Q_i is either $\exists$ or $\forall$ and ϕ is without quantifiers. Show that every formula $\phi(y_1, \ldots, y_n)$ is equivalent to a formula $\psi(y_1, \ldots, y_n)$ in prenex normal form.

3.2.5. First-order theories. Let τ be a signature. A *(τ-)theory* is a set of first-order τ-sentences. Let $\underline{A}$ be a τ-structure and T a τ-theory. Then $\underline{A}$ is a *model of* T, in symbols $\underline{A} \models T$, if $\underline{A} \models \phi$ for all $\phi \in T$. We write $T \models \phi$, and say that T implies ϕ, if $\underline{A} \models \phi$ for every model $\underline{A}$ of T. If neither $T \models \phi$ nor $T \models \neg\phi$, then we say that ϕ is *independent from* T. If S and T are τ-theories then we write $S \models T$ if every model of S is also a model of T.

A τ-theory T is called

- *satisfiable* if T has a model,
- *complete* if T is satisfiable and for every τ-sentence either $T \models \phi$ or $T \models \neg\phi$, and
- *incomplete* if T is not complete.

The *(first-order) theory of* $\underline{A}$ is defined as the set of all first-order τ-sentences that are satisfied by $\underline{A}$. Note that $\mathrm{Th}(\underline{A})$ is always a complete theory.

EXAMPLE 15. The theory T_{AGroup} of abelian groups is over the signature $\tau_{\mathrm{AGroup}} = \{+, -, 0\}$ and contains the following axioms:

$\forall x, y, z.\ (x + y) + z = x + (y + z)$	(associativity)
$\forall x.\ 0 + x = x + 0 = x$	(neutral element)
$\forall x.\ x + (-x) = 0$	(inverse elements)
$\forall x, y.\ x + y = y + x$	(abelian) △

EXAMPLE 16. The theory T_{CRing} of commutative rings is over the signature τ_{Ring}, contains T_{AGroup} and the following additional axioms:

$\forall x, y, z.\ (xy)z = x(yz)$	(associativity)
$\forall x.\ 1 \cdot x = x$	(multiplicative unit)
$\forall x, y.\ xy = yx$	(commutativity)
$\forall x, y, z.\ x(y + z) = xy + xz$	(distributivity) △

EXAMPLE 17. The theory T_{Field} of fields is over the signature τ_{Ring}, contains T_{CRing} and the following additional axioms:

$$\begin{aligned} &\neg(0 = 1) \\ &\forall x\ (\neg(x = 0) \Rightarrow \exists y.\, xy = 1) \end{aligned}$$ △

Exercises.

(19) Show that the theory of abelian groups is incomplete.

(20) Show that $(\mathbb{Q}; +, -, 0, 1, \cdot)$ and $(\mathbb{R}; +, -, 0, 1, \cdot)$ do not have the same first-order theory.

(21) Write down a first-order theory T such that there is a bijection between the models of T with domain B and the graphs with vertex set B.

(22) Show that the sentence

$$\begin{aligned} & \forall x \exists y.\, P(x, y) \\ \wedge\ & \forall x, y, z \big((P(x, y) \wedge P(y, z)) \Rightarrow P(x, z) \big) \\ \wedge\ & \forall x.\, \neg P(x, x) \end{aligned}$$

has no finite models (with at least one element), but infinite models.

(23) Write down the axioms of algebraically closed fields in first-order logic.

(24) Let $\underline{A}$ and $\underline{B}$ be τ-structures and let ϕ be a τ-formula which is *existential positive*, i.e., without universal quantifiers and without negation. Then $\phi^{\underline{A}}$ is preserved by all homomorphisms from $\underline{A}$ to $\underline{B}$.

(25) Show that two structures that are isomorphic satisfy the same first-order sentences.

(26) Show that two finite structures are isomorphic if and only if they have the same theory.

(27) Show that if a relation R has a first-order definition in a structure $\underline{A}$, then R is preserved by all *automorphisms* of $\underline{A}$. An automorphism of $\underline{A}$ is an isomorphism between $\underline{A}$ and $\underline{A}$.

(28) Show that the following are equivalent.
   (a) T is complete.
   (b) T is *maximally satisfiable*, i.e., T is satisfiable, and for every first-order sentence ϕ, either $T \models \phi$ or $T \cup \{\phi\}$ is unsatisfiable.
   (c) $T = \text{Th}(\underline{A})$ for some structure $\underline{A}$;
   (d) T is satisfiable, and $T = \text{Th}(\underline{A})$ for every $\underline{A} \models T$.

(29) Show that the usual ordering $<$ of the real numbers is definable in the structure $(\mathbb{R}; +, \cdot, 0, 1)$.

(30) Show that there is no linear order on the complex numbers which is definable in $(\mathbb{C}; +, \cdot, 0, 1)$.

(31) Find a model of $\text{Th}(\mathbb{Z}; <)$ which is not isomorphic to $(\mathbb{Z}; <)$.

CHAPTER 4

Set Theory

This chapter is a very short introduction to Zermelo–Fraenkel set theory. See [**14**] for a more detailed introduction to set theory. We use ZF to denote the axioms of Zermelo and Fraenkel, which is a first-order theory (Section 4.1). The acronym ZFC stands for the extension of ZF by the axiom of choice; ZFC suffices for practically all of mathematics (ironically, set theory itself is the major field that studies statements that do not follow from ZFC). In particular, we can formalise the concepts of *ordinal* (Section 4.2) and *cardinal* (Section 4.3) in ZFC. The system ZF builds on Zermelo set theory (Z) which we introduce first.

4.1. ZFC

We work with the signature $\tau = \{\in\}$ where $\in$ is a binary relation symbol that is used in infix notation. The elements $a \in U$ of a τ-structure $\underline{U}$ are called *sets*, and '$a \in b$' is intended to mean 'a is an element of b'. The domain U of $\underline{U}$ is called a *universe of sets*; it is a set in the naive sense (on the meta level), and not a set in the sense above. We use the following abbreviations for τ-formulas:

- $x \notin y$ for $\neg(x \in y)$
- $x \subseteq y$ for $\forall z\,(z \in x \Rightarrow z \in y)$.
- $x \subset y$ for $x \subseteq y \wedge x \neq y$.

The Zermelo–Fraenkel axioms (ZF) is the following set of τ-sentences.

(1) *Extensionality.* Sets containing the same elements are equal:

$$\forall x, y\,\big((x \subseteq y \wedge y \subseteq x) \Rightarrow x = y\big)$$

(2) *Empty set.* There exists an empty set (denoted by $\emptyset$):

$$\exists x\,\forall y\,(y \notin x)$$

(3) *Pairing.* For all sets a and b there is a set (denoted by $\{a, b\}$) which has exactly the elements a and b:

$$\forall a, b\,\exists c\,\forall x\,\big(x \in c \Leftrightarrow (x = a \vee x = b)\big)$$

(4) *Union.* For every set a there is a set (denoted by $\bigcup a$; we write $a \cup b$ for $\bigcup\{a, b\}$) that contains precisely the elements of the elements of a:

$$\forall a\,\exists b\,\forall x\,(x \in b \Leftrightarrow \exists y \in a.\, x \in y)$$

(5) *Power set.* For every set a there is a set (denoted by $\mathcal{P}(a)$) that consists of all subsets of a:

$$\forall a\,\exists b\,\forall x\,(x \in b \Leftrightarrow x \subseteq a)$$

(6) *Infinity.* There is an infinite set. One way to express this is to assert the existence of a set which contains the empty set and is closed under the *successor operation* $x \mapsto x^+ := x \cup \{x\}$:

$$\exists a\,\big(\emptyset \in a \wedge \forall x\,(x \in a \Rightarrow x \cup \{x\} \in a)\big).$$

(7) *Comprehension*: If $\phi(x, y_1, \dots, y_m)$ is a first-order $\{\in\}$-formula, for all sets $b_0, b_1, \dots, b_m$ there exists a subset of b_0, denoted by

$$\{x \in b_0 \mid \phi(x, b_1, \dots, b_m)\}$$

containing precisely those elements x of b_0 that satisfy $\phi(x, b_1, \dots, b_m)$. Written as a first-order formula, this is

$$\forall y_0, y_1, \dots, y_m \, \exists z \, \forall x \, \big(x \in z \Leftrightarrow (x \in y_0 \wedge \phi(x, y_1, \dots, y_m)\big).$$

Comprehension is in fact an infinite family of axioms (for every first-order formula ϕ we have one axiom); one therefore speaks of the *axiom scheme of comprehension*. The axioms presented so far are sometimes called *Zermelo's set theory*, abbreviated by Z.

REMARK 4.1.1. The infinity axiom jointly with comprehension implies that there is a smallest set with respect to inclusion that contains the empty set and is closed under the successor operation, which we will then denote by ω. Note that the property of a set to contain $\emptyset$ and to be closed under the successor operation can be expressed by a first-order formula $\phi(x)$. If a is any set that contains $\emptyset$ and is closed under the successor operation (which exists by the axiom of infinity) then $\{x \in a \mid \forall y \, \phi(y) \Rightarrow x \in y\}$ clearly satisfies ϕ, and is contained in any set that is closed under the successor operation and contains $\emptyset$. So $\omega = \{\emptyset, \emptyset^+, (\emptyset^+)^+, \dots\}$. We define $0 := \emptyset$, $1 := \{0\}$, $2 := \{0, 1\}$, and so on. Thus, $\omega = \{0, 1, 2, \dots\}$.

REMARK 4.1.2 (Russel's Antinomy). Comprehension implies that there is no set containing all sets. Indeed, suppose for contradiction that $\underline{U} \models \exists a \forall z. \, z \in a$. We may then apply comprehension with the formula $\phi(x)$ given by $x \notin x$ and obtain the existence of the set $b := \{x \in a \mid x \notin x\}$. Then $b \in b$ if and only if $b \notin b$, a contradiction. This contradiction is called *Russel's Antinomy*.

Fraenkel and independently Skolem **[5, 25]** pointed out that Z is too weak as an axiom system for the intuitive notion of sets that we have in mathematics. For example, in Z one cannot show the existence of the set

$$E := \{Z_0, Z_1, Z_2, \dots\} \tag{6}$$

where $Z_0 := \omega$ and $Z_{i+1} := \mathcal{P}(Z_i)$ for all $i \in \mathbb{N}$: One can prove that $\bigcup E$, equipped with the restriction of $\in$ to that set, provides a model of Z which does not have the element E. (Bonus exercise: verify this.)

Before proceeding with the axioms of ZF, we mention that the axioms presented so far are not independent: Axiom (2) about the existence of an empty set follows from Comprehension: for any set a, the set $\{b \in a \mid \neg(b = b)\}$ does not have elements (and by Extensionality, it is unique with this property); note that we use here the assumption that structures have non-empty domains.

To formulate the remaining axioms of ZF we need the set-theoretic concept of a function, which we introduce now. First, an *ordered pair* (also called *Kuratowski pair* of two sets a and b is the set

$$(a, b) := \big\{\{a\}, \{a, b\}\big\}.$$

Exercises.

(32) Show that if $(a, b) = (a', b')$ if and only if $a = a'$ and $b = b'$.

(33) Do we also get the property formulated in the previous exercise if we would define $(a, b) := \big\{a, \{a, b\}\big\}$. If yes: prove it. If not, why not?

(34) Derive from ZF that for any two sets a, b there exists a *Cartesian product* $a \times b$ of a and b:

$$a \times b := \{(x, y) \mid x \in a \wedge y \in b\}.$$

DEFINITION 4.1.3. A *(binary) relation* is a set of ordered pairs. A *function* is a relation f which satisfies

$$\forall x, y, z \big((x, y) \in f \wedge (x, z) \in f \Rightarrow y = z\big).$$

We write $f(x) = y$ instead of $(x, y) \in f$. Then

$$\mathrm{dom}(f) := \{x \in \bigcup\bigcup f \mid \exists y.\, f(x) = y\}$$

is called the *domain of* f. Likewise,

$$\mathrm{im}(f) := \{y \in \bigcup\bigcup f \mid \exists x.\, f(x) = y\}$$

is called the *image of* f. If $\mathrm{im}(f) \subseteq b$, we write $f\colon \mathrm{dom}(f) \to b$.

Note that if $\underline{U} \models Z$ and $S \subseteq U$ then there may be no element of U that contains precisely the elements of S. So S is not a set in the proper sense, only in the naive sense (on the meta level). Likewise, a function f in the naive sense might not be a function in the sense of Definition 4.1.3. The idea of the following axiom scheme is that if a function f in the meta sense is definable in $\underline{U}$, then the image $\{f(a) \mid a \in d\}$ of an arbitrary set d under f is a set.

(8) *Replacement*: For every first-order $\{\in\}$-formula $\phi(v_1, \dots, v_n, x, y)$ we have the axiom

$$\forall d, v_1, \dots, v_n \big(\forall x, y_1, y_2 (\phi(\bar{v}, x, y_1) \wedge \phi(\bar{v}, x, y_2) \Rightarrow y_1 = y_2) \\ \Rightarrow \exists u \forall y\, (y \in u \Leftrightarrow \exists x \in d.\, \phi(\bar{v}, x, y))\big).$$

EXAMPLE 18. Replacement implies the existence of the set E from (6). To see this, let $\underline{U}$ be a model of the axioms of set theory that we have presented so far. Let $G\colon U \to U$ be a function in the naive sense, defined by

- $G(\emptyset) := \omega$,
- $G(f) := \mathcal{P}(\bigcup \{f(y) \mid y \in \{0, 1, \dots, n\}\})$ if $f\colon \{0, 1, \dots, n\} \to U$, and
- $G(x) := \emptyset$ otherwise.

Claim. There is a $\{\in\}$-formula that defines in $\underline{U}$ a function $F\colon \omega \to U$ in the naive sense such that $F(n) = G(F|_{\{0,\dots,n-1\}})$ where $F|_{\{0,\dots,n-1\}}$ denotes the restriction of F to $\{0, \dots, n-1\}$. We first show that for every $n \in \omega$ there exists a function $f\colon \{0, \dots, n\} \to U$ satisfying

$$\forall i \in \{0, \dots, n-1\}(f(i) = G(f|_{\{1,\dots,i\}})). \tag{7}$$

This can be shown by induction on n: define $f_0 := \emptyset$ and define $f_{n+1} := f_n \cup \{(n, G(f_n))\}$. Now, $F(n) = x$ if and only if there exists a function f with domain $\{0, 1, \dots, n+1\}$ satisfying (7) and $f(n) = x$, and the claim follows. Note that

$$F(0) = G(\emptyset) = \omega$$
$$F(1) = G(F|_{\{0\}}) = \mathcal{P}(F(0)) = \mathcal{P}(\omega),$$
$$F(2) = G(F|_{\{0,1\}}) = \mathcal{P}(F(1)) = \mathcal{P}(\mathcal{P}(\omega)), \text{ etc.}$$

By replacement, the claim implies that

$$\{F(a) \mid a \in \omega\} = \{F(0), F(1), F(2), \dots\} = \{\omega, \mathcal{P}(\omega), \mathcal{P}(\mathcal{P}(\omega)), \dots\}$$

is a set. △

We want to exclude that sets can be their own element. The following axiom is even stronger:

(9) *Foundation.*

$$\forall x \left(x \neq \emptyset \Rightarrow \exists y \in x.\, \neg\exists z \left(z \in x \wedge z \in y\right)\right)$$

Foundation also excludes infinite sequences (a_n) such that a_{i+1} is an element of a_i for all i. Mathematics can be practised without this axiom, and some authors do not include this axiom in ZF. But assuming Foundation simplifies some proofs of fundamental properties of ordinals. This concludes the list of axioms of ZF.

For ZFC we add the *Axiom of Choice*.

(10) *Choice*: For every set a there is a function $c\colon \mathcal{P}(a) \setminus \{\emptyset\} \to a$ (called a *choice function*) such that $c(b) \in b$ for all non-empty $b \in \mathcal{P}(a)$.

Exercises.

(35) Let a and b be sets. Then $\{f \mid f\colon a \to b\}$ is a set.
(36) Show that Foundation indeed rules out the existence of a set a such that $a \in a$.
(37) Show that Replacement implies Comprehension.
(38) (*) Show that Pairing is a consequence of Power set and Replacement.
(39) Prove the claims that are made in Example 18 to prove that ZF implies the existence of the set $\{\omega, \mathcal{P}(\omega), \mathcal{P}(\mathcal{P}(\omega)), \dots\}$.

4.1.1. Sets and classes. In set theory we distinguish between sets and *classes*. This is motivated by the desire to speak e.g. about of the *class of all τ-structures* or the *class of all ordinals*. However, we know that these things cannot be sets (as we will see in the next section, if the class of all ordinals were a set, we could derive a contradiction).

Using first-order logic, we therefore want to give an axiomatic treatment of set theory that allows for the distinction between sets and (proper) classes. This can be done using ZF as follows. Let $\underline{U} = (U; \in)$ be a model of ZF. We now consider the expansion $\underline{U}_U$ of $(U; \in)$ (Definition 3.1.1). If $\phi(x)$ is a first-order $\{\in\} \cup \{c_u \mid u \in U\}$-formula and $S := \{u \in U \mid \underline{U}_U \models \phi(u)\}$, then S is called a *class*. Note that any set $a \in U$ gives rise to a class C_a, namely the class given by the formula '$x \in c_a$'. Note that by Extensionality, $a = b$ if and only if $C_a = C_b$. Classes that are not sets are called *proper classes*.

Remark 4.1.4. There are other axiom systems for set theory, for example *Bernays–Gödel set theory (BG)*. The difference concerns mainly how to implement the distinction between sets and classes. This distinction is more explicit in Bernays–Gödel set theory, which is formulated using 'two-sorted structures'. One sort of objects are *sets* and the other sort of objects are *proper classes*; sets can be elements of sets, and sets can be elements of proper classes, but proper classes cannot be elements of sets or proper classes.

4.1.2. Partial orders and Zorn's lemma. Let A be a set. A *(binary) relation over A* is a subset of $A^2 = A \times A$. A binary relation $\leq$ over A is called a *partial order (on A)* if it is

- *reflexive*: $\forall x \in A.\, x \leq x$.
- *antisymmetric*: $\forall x, y \in A \left((x \leq y \wedge y \leq x) \Rightarrow x = y\right)$.
- *transitive*: $\forall x, y, z \in A \left((x \leq y \wedge y \leq z) \Rightarrow x \leq z\right)$.

We write $x < y$ as a shortcut for $x \leq y \wedge x \neq y$. We sometimes refer to $<$ as a partial order, too; in this case, we mean that the relation $\leq$ defined from $<$ by $x \leq y$ iff $x < y$ or $x = y$ is a partial order.

A *chain* in a partial order $\leq$ on A is subset of A such that for all $x, y \in S$ we have $x \leq y$ or $y \leq x$. If all of A is a chain, then the partial order $\leq$ on A is called a *weak linear order*, and the relation $<$ on A is then called a *strict linear order*. Sometimes the addition of *'weak'* and *'strict'* is omitted, but since it is clear how to pass from weak to strict linear orders and vice versa, this should not cause problems. An element $a \in A$ is called an *upper bound* of $S \subseteq A$ if $s \leq a$ for all $s \in S$. An element $a \in A$ is called *a largest element* if it is an upper bound for all subsets of A and it is called *maximal* if there is no element $b \in A$ such that $a < b$. Note that a partial order can have at most one largest elements, but may have many maximal elements. *Lower bounds*, *smallest elements*, and *minimal elements* are defined analogously.

The following example shows how new partial orders can be constructed from known partial orders; this will be needed later in the section on ordinals to define ordinal addition, multiplication, and exponentiation (Section 4.2).

EXAMPLE 19. Let $<_1$ be a partial order on X_1 and $<_2$ be a partial order on X_2.

- The *ordered sum of* $<_1$ *and* $<_2$ is the partial order $<$ defined on the set $(X_1 \times \{1\}) \cup (X_2 \times \{2\})$ by $(a, i) < (b, j)$ if $i < j$ or if $i = j$ and $a <_i b$.
- The *reverse lexicographic product* of $<_1$ and $<_2$ is defined as the partial order $<$ on $X_1 \times X_2$ defined as follows: $(x, y) < (x', y')$ if $y <_2 y'$ or if $y = y'$ and $x <_1 x'$.
- If X_1 has a smallest element 0. Then the set $X_1^{(X_2)}$ of functions from X_2 to X_1 with finite *support* $\{x \in X_2 \mid f(x) \neq 0\}$ can be partially ordered by setting $f < g$ if there exists $x \in X_2$ such that $f(x) <_1 g(x)$ and $f(x') = g(x')$ for every $x' \in X_1$ with $x <_2 x'$. △

In ZFC one can prove Zorn's lemma, which will be used later in this text. We defer the proof to the next section when we have the concept of ordinals available.

THEOREM 4.1.5 (Zorn's Lemma). *Let $(P; \leq)$ be a partially ordered set with the property that every chain in P has an upper bound in P. Then P contains at least one maximal element.*

4.2. Ordinals

We freely follow Hils and Loeser [**10**]. Ordinals are a natural extension of the natural numbers. They classify well-ordered sets and are a natural device for using transfinite induction.

4.2.1. Well-orders. A partial order $\leq$ on a set X is *well-founded* if any non-empty subset of X contains a smallest element with respect to $\leq$. A *well-order* on a set X is a strict linear order $<$ on X such that the relation $\leq$, defined as usual by $x \leq y$ iff $x < y$ or $x = y$, is well-founded. The usual ordering $<$ of the set $\mathbb{N}$ is an example of a well-order, and the usual ordering $<$ of $\mathbb{Z}$ and the usual ordering $<$ of the non-negative rational numbers are non-examples.

LEMMA 4.2.1. *If X and Y are linearly ordered sets, then the ordered sum and the reverse lexicographic product of X and Y and the partial order defined on $X^{(Y)}$ in Example 19 are linear orders as well. If X and Y are well-founded partial orders, then the ordered sum, the reverse lexicographic product, and the partial order on $X^{(Y)}$ are well-founded as well.*

PROOF. The only non-trivial points to check are the well-foundedness of the reverse lexicographic product and of $X^{(Y)}$. Let Z be a non-empty subset of $X \times Y$. Since Y is well-founded, the set $\{y \mid (x, y) \in Z\}$ has a minimal element y'. Since X

is well-founded, the set $\{x \in X \mid (x,y) \in Z\}$ contains a minimal element x'. Then (x',y') is minimal in Z.

Let Z be a non-empty subset of $X^{(Y)}$. If the constant function with value 0 belongs to Z, then this function is a minimal element in Z. So we may assume that every $f \in Z$ has a non-empty support $\mathrm{Supp}(f)$. Let

$$\begin{aligned} Y_1 &:= \{\max(\mathrm{Supp}(f)) \mid f \in Z\} \\ y_1 &:= \min(Y_1) \\ Z_1' &:= \{f \in Z \mid y_1 = \max(\mathrm{Supp}(f))\} \\ x_1 &:= \min(\{f(y_1) \mid f \in Z_1'\} \\ Z_1 &:= \{f \in Z_1' \mid f(y_1) = x_1\} \end{aligned}$$

Now suppose inductively that for $n \geq 1$ we have already constructed $Y_1, \ldots, Y_n$, $y_1, \ldots, y_n$, $Z_1', \ldots, Z_n'$, $x_1, \ldots, x_n$, and $Z_1, \ldots, Z_n$. If Z_n contains the function with constant value 0 outside $\{y_1, \ldots, y_n\}$, then we have found our minimal element in Z. Otherwise, we have $\mathrm{Supp}(f) \setminus \{y_1, \ldots, y_n\} \neq \emptyset$ for every $f \in Z_n$. Let

$$\begin{aligned} Y_{n+1} &:= \{\max(\mathrm{Supp}(f) \setminus \{y_1, \ldots, y_n\}) \mid f \in Z_n\}, \\ y_{n+1} &:= \min(Y_{n+1}), \\ Z_{n+1}' &:= \{f \in Z_n \mid y_{n+1} = \max(\mathrm{Supp}(f) \setminus \{y_1, \ldots, y_n\})\} \\ x_{n+1} &:= \min(\{f(y_{n+1}) \mid f \in Z_{n+1}'\}) \\ Z_{n+1} &:= \{f \in Z_{n+1}' \mid f(y_{n+1}) = x_{n+1}\}. \end{aligned}$$

If Z_{n+1} contains the function with constant value 0 outside $\{y_1, \ldots, y_n\}$ then we have found our minimal element in Z. Since the sequence (y_i) is strictly decreasing in Y, this case must eventually arise for some finite $n \in \mathbb{N}$. □

4.2.2. Definition of ordinals.

Definition 4.2.2 (von Neumann). A set a is

- *transitive* if for all sets b and c, if $c \in b$ and $b \in a$ then $c \in a$.
- an *ordinal* if it is transitive and if the relation $\{(x,y) \in a \times a \mid x \in y\}$ on a defines a well-order on a.

Note that a is transitive if and only if it satisfies $\forall b\,(b \in a \Rightarrow b \subseteq a)$. The property that $\{(x,y) \in a \times a \mid x \in y\}$ is a well-order on a can be expressed in first-order logic as well; note that well-foundedness can be written as

$$\forall y \in \mathcal{P}(a)\,\big(y \neq \emptyset \Rightarrow \exists z \in y.\, \neg\exists x\,(x \in z \wedge x \in y)\big)$$

Hence, there is a $\{\in\}$-formula $\mathrm{Ord}(x)$ which holds for $c \in U$, for some model $(U;\in)$ of ZF, if and only if c is an ordinal; we may thus speak about the *class of all ordinals*.

From now on, we typically use α, β, γ, etc., to denote ordinals. An element β of an ordinal α is again an ordinal number: Since $\beta \subseteq \alpha$, we obtain a well-order of β by restricting the well-order of α to β.

Proposition 4.2.3. *Let α and β be ordinals. Then $\alpha \in \beta$ if and only if $\alpha \subset \beta$.*

Proof. Clearly, if $\alpha \in \beta$ then $\alpha \subseteq \beta$ by transitivity, and $\alpha \neq \beta$ because $\alpha \notin \alpha$. Conversely, if $\alpha \subset \beta$, then let γ be the smallest element of $\beta \setminus \alpha$ which exists because β is well-ordered.

Claim 1. $\gamma \subseteq \alpha$: let $\delta \in \gamma$. Since $\gamma \in \beta$, we have $\delta \in \beta$. The minimality of γ implies that $\delta \notin \beta \setminus \alpha$, hence, $\delta \in \alpha$.

Claim 2. $\alpha \subseteq \gamma$. Let $\delta \in \alpha$. If $\gamma \in \delta$ we would have $\gamma \in \alpha$ contradicting the definition of γ. Since $\delta, \gamma \in \beta$ and β is totally ordered, this implies that $\delta \in \gamma$.

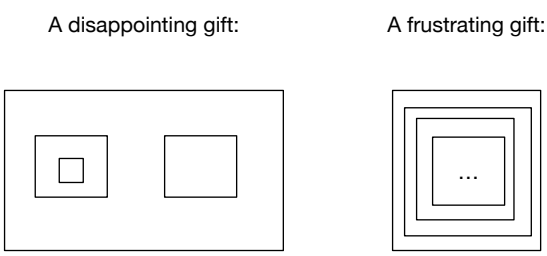


FIGURE 4.1. Moshovakis [**19**]: Ordinals are disappointing, but not frustrating gifts.

The claims imply that $\alpha = \gamma \in \beta$. □

PROPOSITION 4.2.4. *Let X be a non-empty set of ordinals. Then*

$$\bigcap X := \{\alpha \in \bigcup X \mid \forall \beta \in X \colon \alpha \in \beta\}$$

is an element of X which is smallest with respect to $\in$.

PROOF. Clearly, the intersection of transitive sets is transitive and the restriction of a well-order to a subset is a well-order. Hence, $\beta := \bigcap X$ is an ordinal. We have $\beta \subseteq \alpha$ for every $\alpha \in X$. If $\beta \notin X$, then $\beta \in \alpha$ for every $\alpha \in X$ by Proposition 4.2.3. It follows that $\beta \in \beta$, a contradiction. □

PROPOSITION 4.2.5. *Let α and β be ordinals. Then exactly one of the following holds:*

$$\alpha \in \beta, \quad \alpha = \beta, \quad \beta \in \alpha.$$

PROOF. It is clear that the three cases are mutually exclusive. We apply Proposition 4.2.4 to $X := \{\alpha, \beta\}$ and obtain that $\alpha \cap \beta = \alpha$ or $\alpha \cap \beta = \beta$. In the former case, we have $\alpha \subseteq \beta$ and hence $\alpha = \beta$ or $\alpha \in \beta$ by Proposition 4.2.3. In the latter case we conclude analogously that $\alpha = \beta$ or $\beta \in \alpha$. □

So the class of all ordinals is linearly ordered by $\in$. We have to be careful, though.

PROPOSITION 4.2.6 (Burali-Forti paradox). *There is no set that contains precisely the ordinal numbers.*

PROOF. If there were such a set α, then it would be an ordinal itself, and thus $\alpha \in \alpha$, contradicting the assumption that $\in$ is a well-order on the elements of α. □

If α and β are ordinals, then we also write

- $\alpha < \beta$ if $\alpha \in \beta$ and
- $\alpha \leq \beta$ if $\alpha \subseteq \beta$.

This notation is often more suggestive because it emphasises transitivity of $\in$ for ordinals. Moreover, if $(\alpha_i)_{i \in I}$ is a sequence of ordinals for some index set I, we write $\sup_{i \in I} \alpha_i$ for $\bigcup \{\alpha_i \mid i \in I\}$. A function $f \colon P \to Q$ between two ordered sets P and Q is called *strictly increasing* if $a, b \in P$ and $a < b$ implies $f(a) < f(b)$.

LEMMA 4.2.7. *Let $f \colon \alpha \to \beta$ be a map between two ordinals which is strictly increasing. Then $\gamma \leq f(\gamma)$ for every $\gamma \in \alpha$.*

PROOF. Suppose for contradiction that there is $\gamma \in \alpha$ such that $f(\gamma) < \gamma$; choose γ to be minimal. Since f is strictly increasing, $f(f(\gamma)) < f(\gamma)$. This contradicts the minimality of γ, because $\beta := f(\gamma) \in \alpha$ also satisfies $f(\beta) = f(f(\gamma)) < f(\gamma) = \beta$, but $\beta = f(\gamma) < \gamma$. □

PROPOSITION 4.2.8. *Every structure $(A;<)$ where A is well-ordered by $<$ is isomorphic to $(\alpha;<)$ for some ordinal α. Moreover, the ordinal α and the isomorphism are unique.*

PROOF. Define $f\colon A \to \alpha$ inductively by $f(y) := \{f(z) \mid z < y\}$. The image of f is an ordinal α such that $(\alpha;<)$ is isomorphic to $(A;<)$ via f.

Uniqueness: suppose that $f'\colon A \to \alpha'$ is an isomorphism between $(A;<)$ and $(\alpha';<)$ for some ordinal α'. Then $g := f' \circ f^{-1}\colon \alpha \to \alpha$ is an isomorphism as well. Lemma 4.2.7 implies that $g(\beta) \leq \beta$ for every $\beta \in \alpha$. We may argue analogously for g^{-1} and obtain that $g(\beta) = \beta$ and $\alpha = \alpha'$. □

4.2.3. Successor and limit ordinals. Recall the definition of the successor function from the infinity axiom. Note that for any ordinal α the successor $\alpha^+ := \alpha \cup \{\alpha\}$ of α is the smallest ordinal greater than α. Starting from the smallest ordinal $0 := \emptyset$, its successor is $1 := \{0\}$, then $2 := \{0, 1\}$, and so on, yielding the natural numbers $\mathbb{N}$. When we view $\mathbb{N}$ as an ordinal, we denote it by ω (see Remark 4.1.1). The next ordinal is $\omega^+ := \{0, 1, \ldots, \omega\}$, etc. By definition, a successor ordinal β contains a maximal element α (so $\beta = \alpha^+$). Ordinals greater than 0 which are not successor ordinals are called *limit ordinals*. We let $\mathrm{Lim}(x)$ be a formula which defines the class of all limit ordinals.

PROPOSITION 4.2.9. *A non-empty ordinal λ is a limit ordinal if and only if*

$$\lambda = \sup_{\alpha<\lambda} \alpha.$$

PROOF. If λ is a limit ordinal, let $\beta := \sup_{\alpha<\lambda} \alpha$. Clearly, $\beta \subseteq \lambda$. To show that $\lambda \subseteq \beta$, let $\alpha \in \lambda$. Then $\alpha^+ \leq \lambda$ and it follows that $\alpha^+ < \lambda$ because λ is not a successor ordinal. Thus, $\alpha^+ \subseteq \beta$ by the definition of β, and $\alpha \in \alpha^+ \subseteq \beta$.

If $\lambda = \gamma^+$, then $\sup_{\alpha<\lambda} \alpha = \sup_{\alpha\leq\gamma} \alpha = \gamma < \lambda$. □

Any ordinal can be written uniquely as $\lambda \underbrace{+\cdots+}_{n \text{ times}}$ where λ is a limit ordinal or 0.

PROPOSITION 4.2.10 (Transfinite induction). *Let $\underline{U}$ be a model of ZF, and let $\phi(x)$ be a $\{\in\} \cup U$-formula. Then $\underline{U}_U$ satisfies the following induction property:*

$$\big(\phi(0) \wedge \forall\gamma\,(\phi(\gamma) \Rightarrow \phi(\gamma^+)) \wedge \forall\gamma((\mathrm{Lim}(\gamma) \wedge \forall\delta \in \gamma.\, \phi(\delta)) \Rightarrow \phi(\gamma))\big) \Rightarrow \forall\gamma.\, \phi(\gamma)$$

PROOF. Suppose that $\underline{U}_U \models \neg\phi(\alpha)$ for some ordinal α. Choose α to be minimal with this property. Then either $\alpha = 0$, or α is a successor ordinal, or α is a limit ordinal, and in each of the cases one of the preconditions of the implication in the statement does not hold. □

Exercises.

(40) Show that if α, β are ordinals such that $\alpha < \beta$, then $\alpha^+ \subseteq \beta$.
(41) Show that if α, β are ordinals, and $\emptyset < \alpha < \beta$, then $\beta \setminus \alpha$ is not an ordinal.
(42) Let α and β be ordinals. Then $\alpha \cup \beta$ and $\bigcup(\beta \setminus \alpha)$ are ordinals.
(43) Let α be an ordinal. Show that
   - $\bigcup \alpha \subseteq \alpha$.
   - $\bigcup \alpha$ is an ordinal.
(44) If X is a set of ordinals, than $\bigcup X$ is an ordinal.

4.2.4. Ordinal arithmetic. If α and β are ordinals, then the ordered sum of α and β is a well-order by Lemma 4.2.1, and by Proposition 4.2.8 it is isomorphic to a unique ordinal, which is denoted by $\alpha+\beta$. Similarly, $\alpha\beta$ is defined as the unique ordinal isomorphic to the reverse lexicographic product of α and β, and α^β as the unique ordinal isomorphic to $\alpha^{(\beta)}$.

PROPOSITION 4.2.11 (Ordinal addition). *Let α,β be ordinals. Then $\alpha+0=\alpha$ and $\alpha+\beta^+=(\alpha+\beta)^+$. If λ is a limit ordinal, then $\alpha+\lambda=\sup_{\beta<\lambda}(\alpha+\beta)$.*

PROOF. Clearly, $\alpha+0=\alpha$ and $\alpha+1=\alpha^+$. Then $\alpha+\beta^+=(\alpha+\beta)^+$ follows from the easy fact that $\alpha+(\beta+\gamma)=(\alpha+\beta)+\gamma$. Now let λ be a limit ordinal. We first show that $\alpha+\lambda\geq\sup_{\beta<\lambda}(\alpha+\beta)$. This is immediate from the fact that $\alpha+\lambda\geq\alpha+\beta$ for every $\beta<\lambda$. The fact can be seen as follows: $\lambda\setminus\beta$ is a well-ordered set, and hence isomorphic to an ordinal δ by Proposition 4.2.8. Thus, $\alpha+\lambda=\alpha+\beta+\delta>\alpha+\beta$.

We finally show that $\alpha+\lambda\leq\sup_{\beta<\lambda}(\alpha+\beta)$. Let $\mu\in(\alpha+\lambda)$, i.e., $\alpha\leq\mu<\alpha+\lambda$. The $\mu=\alpha+\delta$ for some δ with $0\leq\delta<\lambda$. Since λ is a limit ordinal, one has $\delta^+<\lambda$, hence $\mu<\alpha+\delta^+\leq\sup_{\beta<\lambda}(\alpha+\beta)$. □

Note that by transfinite induction on β, the statements in Proposition 4.2.11 characterise $\alpha+\beta$ uniquely. One can show by induction that $1+\alpha=\alpha+1$ if α is finite. Otherwise, if α is infinite, then $1+\alpha=\alpha$. The reason is that

$$1+\omega=1+\sup_{\beta<\omega}\beta=\sup_{\beta<\omega}(1+\beta)=\sup_{\beta<\omega}\beta^+=\omega.$$

PROPOSITION 4.2.12 (Ordinal multiplication). *Let α,β be ordinals. Then $\alpha 0=0$ and $\alpha\beta^+=\alpha\beta+\alpha$. If λ is a limit ordinal, then $\alpha\lambda=\sup_{\beta<\lambda}(\alpha\beta)$.*

PROOF. The first statement is easy. Let λ be a limit ordinal. If $\alpha=0$ the statement is clear since $0\lambda=0$. So we assume that $\alpha\neq 0$. The inequality $\alpha\lambda\geq\sup_{\beta<\lambda}\alpha\beta$ can be shown similarly as in the proof of Proposition 4.2.11. Conversely, let $\gamma<\alpha\lambda$. We claim that then there are unique ordinals ρ,μ with $\rho<\mu$ such that $\gamma=\alpha\mu+\rho$. This is referred to as *Euclidean division*. If $\gamma=0$ there is nothing to prove. Otherwise, consider the mapping $f_0\colon\gamma\to\alpha\times\beta$ defined by $x\mapsto(0,x)$, which is strictly increasing. Hence, $\gamma\leq\alpha\gamma$ by Lemma 4.2.7. If $\gamma=\alpha\gamma$ one sets $\mu=\gamma$ and $\rho=0$. Otherwise, $\gamma\in\alpha\gamma$. Let f be the unique isomorphism of ordered sets between $\alpha\gamma$ and $\alpha\times\gamma$. One sets $(\rho,\mu)=f(\beta)$. Since $\{(\rho',\mu')\mid\alpha<(\rho',\mu')<(\rho,\mu)\}$ is isomorphic to $(\alpha\times\mu)+\rho$ it follows that $\gamma=\alpha\mu+\rho$, finishing the proof of the claim.

Since $\mu<\lambda$, we have $\mu^+<\lambda$ because λ is a limit ordinal, hence

$$\gamma=\alpha\mu+\rho<\alpha\mu+\alpha=\alpha\mu^+\leq\sup_{\beta<\lambda}\alpha\beta.$$ □

Again, transfinite induction on β shows that the statements in Proposition 4.2.12 characterise $\alpha\beta$ uniquely. Note that $2\omega=\omega<\omega+\omega=\omega 2$, because

$$2\omega=2\sup_{\beta<\omega}\beta=\sup_{\beta<\omega}2\beta=\sup_{\beta<\omega}\beta.$$

PROPOSITION 4.2.13 (Ordinal exponentiation). *Let α,β be ordinals. Then $\alpha^0=1$ and $\alpha^{\beta^+}=\alpha^\beta\alpha$. If λ is a limit ordinal, then $\alpha^\lambda=\sup_{\beta<\lambda}\alpha^\beta$.*

PROOF. The first statement can be checked directly. Let λ be a limit ordinal. Then inequality $\alpha^\lambda\geq\sup_{\beta\in\lambda}\alpha^\beta$ is easy to show. For the converse inequality, let $f\in\alpha^{(\lambda)}$. One may assume that f is not the constant function with value 0. Then $\max(\mathrm{Supp}(f))<\lambda$, and hence $\beta:=\max(\mathrm{Supp}(f))^+<\lambda$, which proves that there exists a strictly increasing function $\{g\in\alpha^{(\lambda)}\mid g\leq f\}\to\alpha^{(\beta)}$; hence, $f<\bigcup_{\beta\in\lambda}\alpha^\beta$ by Lemma 4.2.7. □

Transfinite induction on β shows that the statements in Proposition 4.2.13 characterise α^β uniquely.

4.2.5. The well-ordering theorem.

THEOREM 4.2.14 (Well-ordering theorem). *Every set has a well-ordering.*

PROOF. Let A be a set. Fix a set B which does not belong to A and define a function f from the class of all ordinals to $A \cup \{B\}$ as follows:

- if α is an ordinal such that $A \setminus \{f(\beta) \mid \beta < \alpha\} \neq \emptyset$ then set $f(\alpha)$ to be an element from this set (here we use the Axiom of Choice).
- Otherwise, $f(\alpha) := B$.

Then $\gamma := \{\alpha \mid f(\alpha) \neq B\}$ is an ordinal and the restriction of f to γ is bijection between γ and A. □

Note that the ordinal γ in the construction of the well-order of A is not unique, unless A is finite. The well-ordering theorem is in fact equivalent to the Axiom of Choice.

PROPOSITION 4.2.15. *ZF and the well-ordering theorem imply the Axiom of Choice.*

PROOF. Let A be a set. By hypothesis, A may be well-ordered, say by $<$. The function $f\colon \mathcal{P}(A)\setminus\{\emptyset\} \to A$ which associates to any non-empty subset of A its smallest element is a choice function on A. □

PROOF OF ZORN'S LEMMA (THEOREM 4.1.5). We define inductively a weakly increasing sequence (x_α) in P, indexed by the ordinals. Let x_0 be any element of P. Now suppose that α is an ordinal such that x_β has already been defined for $\beta < \alpha$. If $\alpha = \beta + 1$ then let x_α be x_β if x_β is maximal, and some element of P which is larger than x_β otherwise; such an element exists by the axiom of choice. Otherwise, we take x_α to be an upper bound of the chain $(x_\beta)_{\beta<\alpha}$, which exists by assumption. Note that if P has no maximal elements then the sequence x_α is strictly increasing. But it cannot be strictly increasing, because then the ordinals would be in bijection with a subset of P, contradicting the fact that one cannot form the set of all ordinals (Proposition 4.2.6). This proves that P has a maximal element and completes the proof of Zorn's lemma. □

4.3. Cardinals

Cardinals are used to compare the 'size' of sets, building on the notion of ordinals. We may use the existence of an injective, surjective, or bijective function between two sets to compare their size. The famous results of Cantor-Bernstein (Theorem 4.3.1) and Cantor (Theorem 4.3.2) illustrate this idea.

In the proof of the next theorem we use the following notation for functions $g\colon B \to B$: we define $g^0 := \mathrm{id}$ for $i \in \mathbb{N}$ we inductively define $g^{i+1} := g^{i+1} \circ g$ where $\circ$ denotes function composition.

THEOREM 4.3.1 (Cantor-Bernstein). *Let A and B be sets and let $f\colon A \to B$ and $g\colon B \to A$ be injective maps. Then there exists a bijection $h\colon B \to A$.*

PROOF. We may assume that $A \subseteq B$ and that f is the inclusion map. Now set $C := \{g^n(x) \mid n \in \mathbb{N}, x \in B \setminus A\}$. Note that $B \setminus C \subseteq A$ because $g^0(B \setminus A) = B \setminus A$. Define $h\colon B \to A$ by $h(x) := g(x) \in C$ if $x \in C$ and $h(x) := x$ if $x \in B \setminus C$. The map h is clearly injective and also surjective: indeed, any $y \in A \cap C$ is of the form $y = g(x)$ for some $x \in C$, and $x = h(x)$ for any $x \in A \setminus C$. □

We say that two sets A and B *have the same cardinality*, and write $A \sim B$, if there exists a bijection between them. We say that A *has at most the cardinality of* B, and write $A \preceq B$, if there exists an injection from A to B. The theorem of Cantor-Bernstein states that if $A \preceq B$ and $B \preceq A$, then $A \sim B$.

EXAMPLE 20. The sets $\mathbb{N}^2$ and $\mathbb{N}$ have the same cardinality. A bijection $\mathbb{N}^2 \to \mathbb{N}$ is given by $(x, y) \mapsto \frac{(x+y+1)(x+y)}{2} + x$. △

THEOREM 4.3.2 (Cantor). *Let A be a set. Then there is no surjection $A \to \mathcal{P}(A)$.*

PROOF. Let $f\colon A \to \mathcal{P}(A)$ be a function. Consider the set

$$B := \{x \in A \mid x \notin f(x)\}.$$

For every $x \in A$ with $f(x) = B$ we have that $x \in B$ if and only if $x \notin B$; hence, B does not belong to the image of f. □

By the well-ordering theorem, every set has the same cardinality as some ordinal. We call the smallest such ordinal *the cardinality of* A, denoted by $|A|$. Ordinals occurring in this way are called *cardinals*. An ordinal α is a cardinal if and only if all smaller ordinals do not have the same cardinality.

Notes.

- All natural numbers and ω are cardinals.
- $\omega + 1$ is the smallest ordinal that is not a cardinal.
- The cardinality of a finite set is a natural number.
- A set of cardinality ω is called *countably infinite*.

We write κ^+ for the smallest cardinal greater than κ, the *successor cardinal* of κ. To avoid confusion, from now on the ordinal successor of an ordinal α will be denoted by $\alpha + 1$. Positive cardinals which are not successor cardinals are called *limit cardinals*.

PROPOSITION 4.3.3. *Let X be a set of cardinals. Then $\lambda := \bigcup_{\kappa \in X} \kappa$ is a cardinal.*

PROOF. If there is a $\kappa_0 \in X$ such that $\kappa \leq \kappa_0$ for all $\kappa \in X$, then $\bigcup_{\kappa \in X} \kappa = \kappa_0$ and the statement is true. Otherwise, for every $\kappa \in X$ there is a $\kappa' \in X$ with $\kappa < \kappa'$. For each ordinal α with $\alpha < \lambda$ we have that $\alpha \in \lambda$ and hence $\alpha \in \kappa$ for some $\kappa \in X$. By the above, there is a $\kappa' \in X$ such that $|\alpha| \leq \kappa < \kappa' \leq |\lambda|$. Thus, every ordinal smaller than κ has smaller cardinality than λ, and λ is a cardinal. □

DEFINITION 4.3.4. The $\aleph$-*hierarchy* assigns to any ordinal α a cardinal $\aleph_\alpha$ as follows:

$$\aleph_\alpha := \begin{cases} \omega & \text{if } \alpha = 0 \\ \aleph_\beta^+ & \text{if } \alpha = \beta + 1 \\ \bigcup_{\beta < \alpha} \aleph_\beta & \text{if } \alpha \text{ is a limit ordinal.} \end{cases}$$

PROPOSITION 4.3.5. *Every infinite cardinal is of the form $\aleph_\alpha$ for some ordinal α.*

PROOF. Let κ be an infinite cardinal. It is easy to show by transfinite induction that the function from $\kappa + 1$ to $\aleph_{\kappa+1}$ given by $\beta \mapsto \aleph_\beta$ is strictly increasing. Thus, $\aleph_\kappa \geq \kappa$ by Lemma 4.2.7 and $\aleph_{\kappa+1} > \kappa$. Let $\alpha \leq \kappa + 1$ be minimal with $\aleph_\alpha > \kappa$. Since $\kappa \geq \aleph_0$, we have $\alpha > 0$. If α were a limit ordinal, by definition $\kappa \in \bigcup_{\beta < \alpha} \aleph_\beta$, and hence $\kappa \in \aleph_\beta$ for some $\beta < \alpha$, which would contradict the minimality of α. Thus, $\alpha = \beta + 1$ for some ordinal β, and $\aleph_\beta \leq \kappa < \aleph_{\beta+1} = \aleph_\beta^+$. It follows that $\aleph_\beta = \kappa$. □

We conclude that $\alpha \mapsto \aleph_\alpha$ is an isomorphism between the class of ordinals and the class of all infinite cardinals.

Sums, products, and powers of cardinals are defined as the cardinality of disjoint unions, Cartesian powers, and sets of functions:

$$|x| + |y| := |x \uplus y|$$
$$|x| \cdot |y| := |x \times y|$$
$$|x|^{|y|} := |x^y|$$

and likewise for infinite sums and products:

$$\sum_{x \in I} |x| := \Big| \biguplus_{x \in I} x \Big|$$
$$\prod_{x \in I} |x| := \Big| \prod_{x \in I} x \Big|.$$

Note that

$$(\kappa^\lambda)^\mu = \kappa^{\lambda \cdot \mu}.$$

By Theorem 4.3.2 we have

$$2^\kappa > \kappa\,.$$

In particular, there is no largest cardinal. Cantor's result also follows from König's theorem below for $\kappa_i := 1$ and $\lambda_i := 2$ for all $i \in I$.

THEOREM 4.3.6 (König's theorem). *Let $(\kappa_i)_{i \in I}$ and $(\lambda_i)_{i \in I}$ be sequences of cardinals. If $\kappa_i < \lambda_i$ for all $i \in I$, then*

$$\sum_{i \in I} \kappa_i < \prod_{i \in I} \lambda_i\,.$$

PROOF. We first show that $\sum_{i \in I} \kappa_i \leq \prod_{i \in I} \lambda_i$. Choose pairwise disjoint sets $(A_i)_{i \in I}$ and $(B_i)_{i \in I}$ such that $|A_i| = \kappa_i$, $|B_i| = \lambda_i$, and $A_i \subset B_i$ for all $i \in I$. We will construct an injection $f\colon \bigcup_{i \in I} A_i \to \prod_{i \in I} B_i$. Choose $d_i \in B_i \setminus A_i$ for each $i \in I$ (here we use the Axiom of Choice). For $x \in A := \bigcup_{i \in I} A_i$, define

$$f(x) := (a_i)_{i \in I} \text{ where } a_i := \begin{cases} x & \text{if } x \in A_i \\ d_i & \text{otherwise.} \end{cases}$$

To show the injectivity of f, let $x, y \in A$ be distinct. Let $i \in I$ be such that $x \in A_i$. If $y \in A_i$ then $f(x)_i = x \neq y = f(y)_i$. If $y \notin A_i$ then $f(x)_i = x \neq d_i = f(y)_i$ since $x \in A_i$ but $d_i \in B_i \setminus A_i$. So in both cases, $f(x) \neq f(y)$.

Suppose for contradiction that $\sum_{i \in I} \kappa_i = \prod_{i \in I} \lambda_i$. Then we can find sets $(X_i)_{i \in I}$ with $|X_i| = \kappa_i$ such that

$$B := \prod_{i \in I} B_i = \bigcup_{i \in I} X_i\,.$$

For each $i \in I$, define

$$Y_i := \{a_i \mid a \in X_i\}\,.$$

For every $i \in I$ there exists $b_i \in B_i \setminus Y_i$ because $|Y_i| \leq |X_i| = \kappa_i < \lambda_i = |B_i|$. Now define

$$b := (b_i)_{i \in I} \in \prod_{i \in I} B_i.$$

Let $j \in I$. Then $b_j \notin Y_j$ by the choice of b_j, and hence $b \notin X_j$ by the definition of Y_j. This shows that $b \notin \bigcup_{i \in I} X_i$, a contradiction. □

THEOREM 4.3.7. *Let κ be an infinite cardinal. Then*

(1) $\kappa \cdot \kappa = \kappa$.

(2) $\kappa + \lambda = \max(\kappa, \lambda)$.
(3) $\kappa^\kappa = 2^\kappa$.

PROOF. For ordinals $\alpha, \beta, \alpha', \beta'$, define $(\alpha, \beta) < (\alpha', \beta')$ iff

$$(\max(\alpha, \beta), \alpha, \beta) <_{\text{lex}} (\max(\alpha', \beta'), \alpha', \beta')$$

where lex is the lexicographical order on triples of ordinals. Since this is a well-order, there is a unique order-preserving bijection f between pairs of ordinals and ordinals by Proposition 4.2.8.

Claim. If κ is an infinite cardinal, then f maps $\kappa \times \kappa$ to κ, and hence $\kappa \cdot \kappa = \kappa$. The proof of the claim is by induction on κ. For $\alpha, \beta \in \kappa$ let $P_{\alpha,\beta}$ be the set of predecessors of (α, β). Note that:

- $P_{\alpha,\beta}$ is contained in $\delta \times \delta$ with $\delta = \max(\alpha, \beta) + 1$.
- Since κ is infinite and $\alpha, \beta < \kappa$, the cardinality of δ is smaller than κ.
- By inductive assumption $|P_{\alpha,\beta}| \leq |\delta \times \delta| = |\delta| \cdot |\delta| \overset{\text{IA}}{=} |\delta| < \kappa$.

Hence, $f(\alpha, \beta) < \kappa$ since f is an order isomorphism and thus $f(\alpha, \beta) \in \kappa$.

Now (2) and (3) are simple consequences. Let $\mu := \max(\kappa, \lambda)$.

$$\mu \leq \kappa + \lambda \leq \mu + \mu \leq 2 \cdot \mu \leq \mu \cdot \mu = \mu$$
$$2^\kappa \leq \kappa^\kappa \leq (2^\kappa)^\kappa = 2^{\kappa \cdot \kappa} = 2^\kappa$$ □

REMARK 4.3.8. The *Continuum Hypothesis (CH)* states that $\aleph_1 = 2^{\aleph_0}$, that is: there is no cardinal lying strictly between ω and the cardinality $|\mathbb{R}|$ of the continuum. The *Generalised Continuum Hypothesis (GCH)* states that $\kappa^+ = 2^\kappa$ for all infinite cardinals κ. As with CH, the GCH is known to be independent of ZFC, that is, there are models of ZFC where GCH is true, and models of ZFC where GCH is false (assuming that ZFC is consistent; see **[14]**).

Exercises.

(45) Prove the claim in Example 20 that the given function from $\mathbb{N}^2$ to $\mathbb{N}$ is bijective.

(46) (*) (Exercise 1.11.5 in Hils and Loeser **[10]**) An *Ulam matrix* is a family

$$(U_{\alpha,n})_{\alpha<\aleph_1, n<\omega}$$

of subsets of $\aleph_1$ such that
- $U_{\alpha,n} \cap U_{\beta,n} = \emptyset$ for every $n < \omega$ and all distinct $\alpha, \beta \in \aleph_1$.
- $\aleph_1 \setminus \bigcup_{n<\omega} U_{\alpha,n}$ is countable for every $\alpha < \aleph_1$.

For $\xi < \aleph_1$, let $f_\xi \colon \omega \to \aleph_1$ be such that $\xi \subseteq \text{im}(f_\xi)$. Define

$$U_{\alpha,n} := \{\xi < \aleph_1 \mid f_\xi(n) = \alpha\}.$$

Prove that $(U_{\alpha,n})$ is an Ulam matrix.

(47) (*) (Exercise 1.11.5 in Hils and Loeser **[10]**) Let $\mu \colon \mathcal{P}(\aleph_1) \to [0, 1]$ be a *σ-additive measure on* $\aleph_1$, that is,

$$\mu(\bigcup_{n<\omega} A_n) = \sum_{n<\omega} \mu(A_n)$$

for all families $(A_n)_{n<\omega}$ of pairwise disjoint subsets of $\aleph_1$. Prove that if $\mu(\{\alpha\}) = 0$ for every $\alpha < \aleph_1$, then μ is identically zero.

Hints:
- show that there are at most countably many pairwise disjoint subsets of $\aleph_1$ with positive measure.
- use the previous exercise.

(48) (*) (Exercise 1.11.10 in Hils and Loeser [**10**]) The goal of this exercise is to prove that the continuum hypothesis holds for closed subsets of the real line (Cantor's theorem), that is, if $S \subseteq \mathbb{R}$ is closed, then $|S| \in \{\aleph_0, 2^{\aleph_0}\}$. Suppose that $|S| > \alpha_0$. Let C be the set of $x \in S$ such that there is an open $U \subseteq \mathbb{R}$ which contains x and satisfies $|U \cap S| \leq \aleph_0$.

- Show that C is countable.
- Show that $S' := S \setminus C$ is a closed subset of $\mathbb{R}$.
- Show that any non-empty open subset of S' is uncountable.
- Let I be an open interval such that $I \cap S' \neq \emptyset$. Prove that for any $\epsilon > 0$ there are open intervals I_0 and I_1 of length at most ϵ such that $I_i \cap S' \neq \emptyset$ and $\bar{I}_i \subseteq I$ for $i \in \{0, 1\}$ and such that $\bar{I}_0 \cap \bar{I}_1 = \emptyset$.
- Prove that there is an injection from $2^{\aleph_0}$ into S'.
- Prove Cantor's theorem.

CHAPTER 5

The Completeness Theorem

In this section we formalise the notion of a proof. A *formal proof* consists of a sequence of sentences; each sentence in the sequence is either a *logical axiom* or derived from previous sentences by a deduction principle called *modus ponens*. The sequence is then viewed as a proof of the final sentence in the sequence. The key features of our notion of a formal proof are that

(1) every sentence that has a formal proof is valid *(soundness)*,
(2) every valid sentence has a formal proof *(completeness)*, and
(3) we can write a computer program that decides for a given sentence in the sequence whether it is a logical axiom or whether it can be derived from previously derived axioms via modus ponens *(effectivity)*.

It is easy to adapt the proof system so that the running time of the algorithm is even polynomial in the size of the proof *(efficiency)*.

Our set of axioms is infinite. It will be chosen so that the proof of the completeness theorem is as simple as possible. We essentially follow Hils and Loeser **[10]**.

5.1. Logical Axioms

All our logical axioms are sentences that are valid (so the logical axioms should not be confused with the axioms of set theory, which clearly do not hold in all structures). We start with axioms that describe properties of equality. It is clear that the following first-order sentences are valid.

$$\forall x.\ x = x \qquad \text{(reflexivity, E1)}$$

$$\forall x, y\ (x = y \Rightarrow y = x) \qquad \text{(symmetry, E2)}$$

$$\forall x, y, z\ \big((x = y \wedge y = z) \Rightarrow x = z\big) \qquad \text{(transitivity, E3)}$$

Let τ be a signature. For every function symbol $f \in \tau$ of arity n, the following *congruence condition for functions* is valid.

$$\forall x_1, \ldots, x_n, y_1, \ldots, y_n\ \Big(\bigwedge_{i=1}^{n} x_i = y_i \Rightarrow\ f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)\Big) \qquad \text{(E4)}$$

For every relation symbol $R \in \tau$ of arity n the following *congruence condition for relations* is valid.

$$\forall x_1, \ldots, x_n, y_1, \ldots, y_n\ \Big(\bigwedge_{i=1}^{n} x_i = y_i \Rightarrow\ (R(x_1, \ldots, x_n) \Leftrightarrow R(y_1, \ldots, y_n))\Big) \qquad \text{(E5)}$$

LEMMA 5.1.1. *Let $\phi(\bar{y})$ be a τ-formula and let x be a variable. Then the following sentence is valid.*

$$\forall \bar{y}.\ \phi \Rightarrow \forall \bar{y}, x.\ \phi \qquad \text{(Q1)}$$

EXAMPLE 21. For a unary relation symbol R, the following $\{R\}$-sentences

$$\forall y.\, R(y) \Rightarrow \forall y, x.\, R(y)$$
$$\forall y.\, R(y) \Rightarrow \forall x, y.\, R(y)$$

are *both* covered by (Q1): the first statement can be obtained by instantiating ϕ with $R(y)$, the second can be obtained by instantiating ϕ with $\forall y.R(y)$ (note that in this case, the tuple $\bar{y}$ in (Q1) has length 0). △

Note that $\forall x.\neg\phi$ is by definition the same as $\neg\exists x.\neg\neg\phi$, which is equivalent to $\neg\exists x.\phi$. This implies the following lemma.

LEMMA 5.1.2. *Let $\phi(\bar{y}, x)$ be a τ-formula. Then the following sentence is valid.*

$$\forall \bar{y}\, (\forall x.\, \neg\phi \Rightarrow \neg\exists x.\, \phi) \qquad \text{(Q2)}$$

DEFINITION 5.1.3 (Substitution in terms). Let s and t be τ-terms and x a variable. We write $s[x \mapsto t]$ for the τ-term obtained from s by replacing all occurrences of x by the term t.

Substitution in formulas is similar, but there is an important complication.

DEFINITION 5.1.4 (Substitution in formulas). If ϕ is a τ-formula, t is a τ-term, and x is a free variable in ϕ, then we write $\phi[x \mapsto t]$ for the τ-formula obtained from ϕ by replacing all the free occurrences of x by t; however, if one of the variables of t would be quantified by a quantifier of ϕ, then we first rename the variable of ϕ that is quantified so that this does not happen.

EXAMPLE 22. Let ϕ be the formula $\forall y.\, P(x, y)$ and let t be the term $f(y)$. If we simply replace x by $f(y)$ we would obtain $\forall y.\, P(f(y), y)$. But this is not what we want; we have to first rename y to z, and then obtain $\forall z.\, P(f(y), z)$. Note that this distinction has an effect on the free variables of the resulting formula. △

LEMMA 5.1.5. *Let $\phi(\bar{y}, x)$ be a τ-formula and let $t(\bar{y})$ be a τ-term. Then the following sentence is valid.*

$$\forall \bar{y}\, \big(\phi[x \mapsto t] \Rightarrow \exists x.\, \phi\big) \qquad \text{(Q3)}$$

PROOF. Let $\bar{y} = (y_1, \ldots, y_n)$. Suppose that there is a τ-structure $\underline{A}$ and $\bar{a} \in A^n$ such that $\underline{A} \models \phi[x \mapsto t](\bar{a})$. Then $t^{\underline{A}}(\bar{a})$ provides a witness for x that shows that $\underline{A} \models \exists x.\, \phi(\bar{a})$. □

LEMMA 5.1.6. *Let $\phi(\bar{y})$ and $\psi(\bar{y})$ be τ-formulas and suppose that the variable x is not free in ϕ. Then the following sentences are valid.*

$$\forall \bar{y}\, \big(\forall x\, (\phi \Rightarrow \psi) \Rightarrow (\phi \Rightarrow \forall x.\, \psi)\big) \qquad \text{(Q4)}$$
$$\forall \bar{y}\, \big((\phi \Rightarrow \forall x.\, \psi) \Rightarrow \forall x\, (\phi \Rightarrow \psi)\big) \qquad \text{(Q5)}$$

PROOF. Let $\underline{A}$ be a τ-structure, let $\bar{y} = (y_1, \ldots, y_n)$ be a sequence of variables, and let $\bar{a} \in A^n$. Then

$$\begin{aligned} & \underline{A} \models \forall x\, (\phi \Rightarrow \psi)(\bar{a}) \\ \text{if and only if } \ & \underline{A} \models (\phi \Rightarrow \psi)(\bar{a}, b) \text{ for every } b \in A \\ \text{if and only if } \ & \underline{A} \models \phi(\bar{a}) \Rightarrow \psi(\bar{a}, b) \text{ for every } b \in A \\ \text{if and only if } \ & \underline{A} \models (\phi \Rightarrow \forall x\, \psi)(\bar{a}). \end{aligned}$$

This shows both statements. □

LEMMA 5.1.7. *Let ϕ be a propositional tautology with propositional variables $X_1, \dots, X_m$ and let $\psi_1, \dots, \psi_m$ be first-order τ-formulas with free variables $x_1, \dots, x_n$. Then the sentence*

$$\forall x_1, \dots, x_n . \phi(\psi_1, \dots, \psi_m) \qquad \text{(TAUT)}$$

is valid where $\phi(\psi_1, \dots, \psi_m)$ is the τ-formula obtained from ϕ by replacing X_i by ψ_i, for all $i \in \{1, \dots, m\}$.

PROOF. Clear. □

Our logical axioms will be (E1)-(E5), (Q1)-(Q5), and (TAUT).

5.2. Formal Proofs

Our notion of a formal proof has only one type of deduction step: modus ponens. The following is clear.

LEMMA 5.2.1 (Modus Ponens (MP)). *Let $\phi(\bar{x}), \psi(\bar{x})$ be τ-formulas and T a τ-theory. If $T \models \forall \bar{x} . \phi$ and $T \models \forall \bar{x} (\phi \Rightarrow \psi)$, then $T \models \forall \bar{x} . \psi$.*

It will be convenient to define the concept of a formal proof with respect to some fixed first-order τ-theory T; the idea is that the sentences from T can be used like additional axioms in the proof. If $T = \emptyset$ we simply drop the reference to T in the notation. Note that in the formulas $\phi(\bar{x}), \psi(\bar{x})$ in Modus Ponens and all logical axioms the variable vector $\bar{x}$ might also have length zero!

DEFINITION 5.2.2. Let ϕ be a τ-sentence and T a τ-theory. A *formal proof of ϕ in T* is a sequence of τ-sentences $(\phi_1, \dots, \phi_n)$ with $\phi_n = \phi$ such that for every $i \in \{1, \dots, n\}$

- $\phi_i \in T$, or
- ϕ_i is a logical axiom, or
- ϕ_i can be deduced from some ϕ_j and ϕ_k with $j, k < i$ by MP, i.e.,

$$\begin{aligned} &\phi_j \text{ is of the form } \forall \bar{x} . \psi_j . \\ &\phi_k \text{ is of the form } \forall \bar{x} (\psi_j \Rightarrow \psi_i), \\ &\phi_i \text{ is of the form } \forall \bar{x} . \psi_i . \end{aligned}$$

If there exists a formal proof of ϕ in T, then we write $T \vdash_\tau \phi$, and otherwise we write $T \nvdash_\tau \phi$. It T is empty, we simply omit T in this notation, and write $\vdash_\tau \phi$.

REMARK 5.2.3. Note that a priori, the provability of a τ-sentence from a τ-theory T could depend on the signature allowed in the proofs, because if we add symbols to τ, there are more proofs; this is why we write $T \vdash_\tau \phi$ to indicate that only symbols from τ are allowed in the proof. In contrast, it is easy to see that the models relation $\models$ does not depend on the signature (note that here we use the assumption that structures have non-empty domains). It will therefore be a consequence of the completeness theorem (Theorem 5.4.1) that choosing a larger signature does not increase the set of τ-formulas that can be proved from T. This is why we later simply write $T \vdash \phi$ without reference to the signature.

LEMMA 5.2.4 (Soundness). *If $T \vdash_\tau \phi$ then $T \models \phi$.*

PROOF. Clearly, if $\phi \in T$, then $T \models \phi$. If ϕ is a logical axiom, then $T \models \phi$ (Section 5.1). The statement now follows from Lemma 5.2.1 by induction over the length of the proof. □

REMARK 5.2.5. It is straightforward to write a computer program that checks for a given string whether it is a formal proof. If we require that the logical axioms obtained from a propositional tautology ϕ, (TAUT), are presented together with a proof of ϕ (e.g., a resolution refutation of a transformation of $\neg\phi$ into CNF), then we can find a program whose running time is polynomial in the length of the string.

REMARK 5.2.6. Our notion of formal proof is an example of a family of deduction systems commonly referred to as a *Hilbert-style deductive system*. Such systems might vary in the precise choice of the deduction rules and the axioms; generally speaking, fewer axioms require more deduction rules, and fewer deduction rules require more axioms.

There are also different deductive systems, typically with the goal to allow for shorter proofs or to make the respective proofs easier to read for humans, like for example Gentzen's *sequent calculus*.

REMARK 5.2.7. There is also a (sound and complete) proof system for propositional logic which exclusively uses modus ponens as a proof step, and only requires six axiom schemes, called *Frege's propositional calculus*. So we might alternatively replace (TAUT) in our notion of formal proof by the Frege axioms.

Examples of formal proofs can be found in the proofs of the following lemmata.

LEMMA 5.2.8. *Let $\phi(\bar{y}, x)$ be a τ-formula and let $t(\bar{y})$ be a τ-term. Then*

$$\vdash_\tau \forall\bar{y}\,(\forall x.\,\phi \Rightarrow \phi[x \mapsto t]).$$

PROOF. The following is a formal proof of $\forall\bar{y}\,(\forall x.\,\phi \Rightarrow \phi[x \mapsto t])$.

$\forall\bar{y}\big(\neg\phi[x \mapsto t] \Rightarrow \exists x.\,\neg\phi\big)$	(Q3)
$\forall\bar{y}\big((\neg\phi[x \mapsto t] \Rightarrow \exists x.\,\neg\phi) \Rightarrow (\forall x.\,\phi \Rightarrow \phi[x \mapsto t])\big)$	(TAUT)
$\forall\bar{y}\,(\forall x.\,\phi \Rightarrow \phi[x \mapsto t])$	(MP) □

LEMMA 5.2.9. *Let $\phi(\bar{y}, x)$ be a τ-formula and suppose that z is a variable that does not occur in ϕ. Then*

$$\vdash_\tau \forall\bar{y}\,(\forall z.\,\phi[x \mapsto z] \Rightarrow \forall x.\,\phi).$$

PROOF. Let $\psi := \phi[x \mapsto z]$. Note that $\psi[z \mapsto x] = \phi$. We then have the following formal proof.

$\forall\bar{y}, x\,(\forall z.\,\psi \Rightarrow \psi[z \mapsto x])$	(Lemma 5.2.8)
$\forall\bar{y}\,\big(\forall x\,(\forall z.\,\phi[x \mapsto z] \Rightarrow \phi) \Rightarrow (\forall z.\,\phi[x \mapsto z] \Rightarrow \forall x.\,\phi)\big)$	(Q4)
$\forall\bar{y}\,(\forall z.\,\phi[x \mapsto z] \Rightarrow \forall x.\,\phi)$	(MP) □

REMARK 5.2.10. Some authors who use Hilbert-style proof systems need another deduction step, *generalisation*, and work with formulas instead of sentences, which we avoid here.

Exercises.

(49) Show that if $T \vdash_\tau \forall\bar{x}.\phi$ and $T \vdash_\tau \forall\bar{x}.\psi$ then $T \vdash_\tau \forall\bar{x}(\phi \wedge \psi)$.

(50) Let $\tau := \{R\}$ where R is a binary relation symbol. Find formals proofs for the following sentences.

$$\forall x, y.\,R(x, y) \Rightarrow \forall y, x.\,R(x, y)$$
$$\text{and}\quad \exists x\forall y.\,R(x, y) \Rightarrow \forall y\exists x.\,R(x, y)$$

5.3. Consistency

A τ-theory T is called *inconsistent* if there is a τ-sentence ϕ such that $T \vdash_\tau \phi$ and $T \vdash_\tau \neg\phi$, and it is called *consistent* otherwise. Clearly, every satisfiable theory is consistent.

LEMMA 5.3.1. *A τ-theory T is inconsistent if and only if $T \vdash_\tau \phi$ for any τ-sentence ϕ.*

PROOF. Let ϕ be a τ-sentence and suppose that ψ is a τ-sentence such that $T \vdash_\tau \psi$ and $T \vdash_\tau \neg\psi$. Note that $(X \Rightarrow (\neg X \Rightarrow Y))$ is a tautology, so use (TAUT) and two times (MP) to obtain $T \vdash_\tau \phi$. The converse implication is immediate. □

LEMMA 5.3.2. *If $T \vdash_\tau \phi$ then there exists a finite subset T_0 of T such that $T_0 \vdash_\tau \phi$.*

PROOF. Formal proofs are finite. □

COROLLARY 5.3.3. *Let T be a τ-theory such that all finite subsets of T are consistent. Then T is consistent as well.*

LEMMA 5.3.4 (Deduction Lemma). *Let χ and ϕ be τ-sentences and T a τ-theory. Then*

$$T \cup \{\chi\} \vdash_\tau \phi \text{ if and only if } T \vdash_\tau (\chi \Rightarrow \phi).$$

PROOF. Clearly, if $T \vdash_\tau (\chi \Rightarrow \phi)$ then $T \cup \{\chi\} \vdash_\tau \phi$ by (MP). Conversely, let $(\phi_1, \dots, \phi_n)$ be a formal proof of ϕ in $T \cup \{\chi\}$. We prove by induction on $i \in \{1, \dots, n\}$ that $T \vdash_\tau (\chi \Rightarrow \phi_i)$.

- If $\phi_i = \chi$ this follows from (TAUT).
- If ϕ_i is from T or a logical axiom, then the statement follows from the fact that $(\phi_i \Rightarrow (\chi \Rightarrow \phi_i))$ is a tautology and (MP).
- Otherwise, ϕ_i is deduced by (MP) applied to ϕ_j and ϕ_k for $j, k < i$. Suppose that ϕ_j is of the form $\forall\bar{y}.\, \psi_j$, and ϕ_i is of the form $\forall\bar{y}.\, \psi_i$, and that ϕ_k equals $\forall\bar{y}\,(\psi_j \Rightarrow \psi_i)$. By the inductive assumption we have that $T \vdash_\tau (\chi \Rightarrow \phi_j)$ and $T \vdash_\tau \big(\chi \Rightarrow \forall\bar{y}\,(\psi_j \Rightarrow \psi_i)\big)$. By (Q5) and (MP) we obtain that $T \vdash_\tau \forall\bar{y}\,(\chi \Rightarrow (\psi_j \Rightarrow \psi_i))$. Then we use the fact that

  $$(\chi \Rightarrow \psi_j) \Rightarrow \big((\chi \Rightarrow (\psi_j \Rightarrow \psi_i)) \Rightarrow (\chi \Rightarrow \psi_i)\big)$$

  is a tautology and two times (MP) to derive that $T \vdash_\tau \forall\bar{y}\,(\chi \Rightarrow \psi_i)$. Finally, by (Q4) and (MP) we obtain $T \vdash_\tau (\chi \Rightarrow \forall\bar{y}.\, \psi_i)$, i.e., $T \vdash_\tau (\chi \Rightarrow \phi_i)$. □

COROLLARY 5.3.5. *Let T be a τ-theory and ϕ a τ-sentence. Then $T \vdash_\tau \phi$ if and only if $T \cup \{\neg\phi\}$ is inconsistent.*

PROOF. If $T \vdash_\tau \phi$, then in particular $T \cup \{\neg\phi\} \vdash_\tau \phi$, but clearly $T \cup \{\neg\phi\} \vdash_\tau \neg\phi$, so $T \cup \{\neg\phi\}$ is inconsistent.

Conversely, if $T \cup \{\neg\phi\}$ is inconsistent, then $T \cup \{\neg\phi\} \vdash_\tau \phi$ by Lemma 5.3.1. It follows from the deduction lemma (Lemma 5.3.4) that $T \vdash_\tau (\neg\phi \Rightarrow \phi)$. Since $(\neg\phi \Rightarrow \phi) \Rightarrow \phi$ is a tautology, we may use (TAUT) and (MP) to deduce that $T \vdash_\tau \phi$. □

Exercises.

(51) Let $\tau := \{R, S\}$ where R and S are unary relation symbols. Show the following (without using the completeness theorem).

$$\big\{\exists x.R(x), \forall y(R(y) \Rightarrow S(y))\big\} \vdash_\tau \quad \exists x.\, S(x)$$

(52) (*) Find a formal proof that shows that the pairing axiom implies

$$\forall a \exists c (x \in c \Leftrightarrow x = a).$$

(53) (*) Give a formal proof that the pairing axiom and the axiom of foundation imply $\forall a.\neg(a \in a)$.

The previous two exercises illustrate the severe complications we are facing with writing formal proofs in our proof system for some simple mathematical facts. In the light of these exercises it is surprising how easy the proof of the completeness theorem in the next section is.

5.4. Henkin Theories

In this section, we prove the completeness theorem of first-order logic, first proved by Gödel [**7**].

THEOREM 5.4.1 (Completeness). *Let T be a τ-theory and let ϕ be a τ-sentence. Then $T \models \phi$ if and only if $T \vdash_\tau \phi$.*

It is easy to see that the completeness theorem is equivalent to the following theorem, which we will prove at the end of this section.

THEOREM 5.4.2. *A theory has a model if and only if it is consistent.*

PROOF OF THEOREM 5.4.1.

$T \not\models \phi$ if and only if $T \cup \{\neg\phi\}$ has a model (by definition)
if and only if $T \cup \{\neg\phi\}$ is consistent (by Theorem 5.4.2)
if and only if $T \not\vdash_\tau \phi$ (by Corollary 5.3.5). □

The method that we use to prove Theorem 5.4.2 is due to Henkin. One of the ideas of the proof is to work with a signature that has additional new constant symbols. We first state an easy lemma concerning new constant symbols.

LEMMA 5.4.3. *Let $\psi(\bar{y}, x)$ be a τ-formula, T a τ-theory, and let c be a constant symbol not contained in τ. The the following are equivalent:*

(1) $T \vdash_\tau \forall \bar{y}, x.\, \psi$;
(2) $T \vdash_{\tau\cup\{c\}} \forall \bar{y}, x.\, \psi$;
(3) $T \vdash_{\tau\cup\{c\}} \forall \bar{y}.\, \psi[x \mapsto c]$.

PROOF. (1) ⇒ (2) is clear since every τ-proof is a $\tau \cup \{c\}$-proof.
(2) ⇒ (3). A special case of Lemma 5.2.8.
(3) ⇒ (1). Let $\phi_1, \ldots, \phi_m$ be a formal proof of $\psi[x \mapsto c]$ in the signature $\tau \cup \{c\}$. Let z be a variable that does not appear in the entire proof $\phi_1, \ldots, \phi_m$. We have to find a formal proof of $\forall \bar{y}, x.\, \psi$ in the signature of τ.

Let ϕ_i' be the τ-formula obtained from ϕ_i by replacing all occurrences of c by z. If $\phi_i \in T$, then the symbol c does not appear in ϕ_i, and thus $T \vdash_\tau \forall \bar{y}, z.\, \phi_i'$ by (Q1) and (MP). If ϕ_i is an axiom, then so is $\forall z.\, \phi_i'$: this is straightforward to check for (E1)-(E5), (TAUT), and (Q1),(Q2),(Q4),(Q5). For (Q3), suppose that ϕ_i is of the form $\forall \bar{y}(\delta[x \mapsto t] \Rightarrow \exists x.\, \delta)$. Note that in this case ϕ_i' is of the form $\forall \bar{y}\,(\delta'[x \mapsto t'] \Rightarrow \exists x.\, \delta')$ where δ' and t are obtained from δ and t by replacing all occurrences of c by z. Then $\forall z, \bar{y}(\delta'[x \mapsto t'] \Rightarrow \exists x.\, \delta')$ is an instance of (Q3). If ϕ_i is obtained from ϕ_j and ϕ_k by MP, then $\forall z.\, \phi_i'$ is obtained from $\forall z.\, \phi_j'$ and $\forall z.\, \phi_k'$ by MP. This shows that $T \vdash \forall z.\, \phi_m'$. Note that $\phi_m = \psi[x \mapsto c]$ and that $\phi_m' = \psi[x \mapsto z]$. Using Lemma 5.2.9 and (MP) we finally deduce that $\vdash_\tau \forall \bar{y}, x.\, \psi$. □

LEMMA 5.4.4. *Let T be a consistent τ-theory, ϕ a τ-sentence, and $c \in \tau$ a constant symbol not occurring in $T \cup \{\phi\}$. Then $T' := T \cup \{\exists x.\, \phi \Rightarrow \phi[x \mapsto c]\}$ is a consistent τ-theory.*

PROOF. If T' is inconsistent then $T \vdash_\tau (\exists x.\, \phi \wedge \neg\phi[x \mapsto c])$ by Corollary 5.3.5. In particular, $T \vdash_\tau \exists x.\, \phi$. The implication (3) $\Rightarrow$ (1) of Lemma 5.4.3 implies that $T \vdash_\tau \forall x.\, \neg\phi$. We have $T \vdash \forall x.\neg\phi \Rightarrow \neg\exists x.\phi$ by (Q2) and obtain $T \vdash_\tau \neg\exists x.\, \phi$ by (MP), so T is inconsistent. □

DEFINITION 5.4.5. Let τ be a signature and let ρ be a set of *new* constant symbols (i.e., $\rho \cap \tau = \emptyset$). A $(\tau \cup \rho)$-theory T is called a *Henkin theory* if for every $(\tau \cup \rho)$-formula $\phi(x)$ there is a constant $c \in \rho$ such that

$$\big(\exists x.\phi(x) \Rightarrow \phi(c)\big) \in T.$$

The elements of ρ are called *Henkin constants* of T.

If ρ is a set of constant symbols and $\underline{A}$ is a $(\tau \cup \rho)$-structure such that $A = \{c^{\underline{A}} \mid c \in \rho\}$ then $\mathrm{Th}(\underline{A})$ is a (complete) Henkin theory. To formulate a converse of this observation, we introduce the concept of *finite completeness*.

DEFINITION 5.4.6. A τ-theory T is called *finitely complete* if it is consistent and for every τ-sentence ϕ

$$\text{either } T \vdash_\tau \phi \text{ or } T \vdash_\tau \neg\phi.$$

Hils and Loeser [**10**] simply write *complete* instead of *finitely complete*. However, we have already defined completeness. The completeness theorem implies that finite completeness is equivalent to completeness, and once we have completely[1] proved the completeness theorem we will simply write *complete* instead of *finitely complete*.

LEMMA 5.4.7. *Every finitely complete Henkin $(\tau \cup \rho)$-theory T with Henkin constants ρ has a model $\underline{A}$ such that*

$$A = \{c^{\underline{A}} \mid c \in \rho\}.$$

PROOF. Replacing T by the set of $(\tau \cup \rho)$-sentences ϕ such that $T \vdash_{\tau\cup\rho} \phi$ does not change the assumptions. We may thus assume that T is *deductively closed*, that is, $T \vdash_{\tau\cup\rho} \phi$ if and only if $\phi \in T$.

We define for $c, d \in \rho$ the relation $c \sim d$ iff $(c = d) \in T$. It follows from (E1)-(E3) that $\sim$ is an equivalence relation. We define a $(\tau \cup \rho)$-structure $\underline{A}$ on $A := \rho/\!\sim$ by setting

$$([c_1]_\sim, \dots, [c_k]_\sim) \in R^{\underline{A}} \text{ iff } R(c_1, \dots, c_k) \in T \text{ for } R \in \tau$$
$$f^{\underline{A}}([c_1]_\sim, \dots, [c_k]_\sim) = [c_0]_\sim \text{ iff } f(c_1, \dots, c_k) = c_0 \in T \text{ for } f \in (\tau \cup \rho).$$

It follows from the equality axioms (E4) and (E5) that this is well-defined. Also note that f is defined on all of A^k. Indeed, $\forall \bar{y}, x.\, x = x \in T$ by (E1) and (Q1), and therefore $\forall \bar{y}.\, f(\bar{y}) = f(\bar{y}) \in T$ by Lemma 5.2.8. Thus, $\forall \bar{y} \exists x.\, x = f(\bar{y}) \in T$ by (Q3). Since T admits Henkin witnesses in ρ, there exists $c \in \rho$ such that $\forall \bar{y}(c = f(\bar{y})) \in T$.

We claim that $\mathrm{Th}(\underline{A}) = T$, and show by induction on the number of symbols in a first-order $(\tau \cup \rho)$-sentences ϕ that

$$\underline{A} \models \phi \text{ if and only if } \phi \in T.$$

- ϕ is atomic. If ϕ has the form $c = d$ or $R(c_1, \dots, c_n)$, for $c, d, c_1, \dots, c_n \in \rho$, then the statement follows from the construction of $\underline{A}$. Otherwise, ϕ contains a function symbol $f \in \tau$, so ϕ can be written as $\psi(f(c_1, \dots, c_k))$ for some $(\tau \cup \rho)$-formula $\psi(x)$ and $c_1, \dots, c_k \in \rho$ (note that k might be 0, in which

[1] I could not resist to use this word here; here, 'completely' is part of the meta language.

case f is a constant symbol from τ). By construction, there exists $c \in \rho$ such that T contains $f(c_1, \ldots, c_k) = c$, so $\underline{A} \models f(c_1, \ldots, c_k) = c$. Thus,

$$\begin{aligned} \underline{A} \models \phi &\text{ iff } \underline{A} \models \psi(c) \\ &\text{iff } \psi(c) \in T \\ &\text{iff } \phi \in T. \end{aligned}$$

Here, the second equivalence is by inductive assumption since $\psi(c)$ has less symbols than ϕ.

- ϕ is of the form $\neg\psi$. Then

$$\begin{aligned} \underline{A} \models \neg\psi &\text{ iff } \underline{A} \not\models \psi \\ &\text{iff } \psi \notin T && \text{(by inductive assumption)} \\ &\text{iff } \neg\psi \in T && \text{(by finite completeness).} \end{aligned}$$

- ϕ is of the form $\psi_1 \wedge \psi_2$. Then

$$\begin{aligned} &\underline{A} \models \psi_1 \wedge \psi_2 \\ &\text{iff } \underline{A} \models \psi_1 \text{ and } \underline{A} \models \psi_2 \\ &\text{iff } \psi_1 \in T \text{ and } \psi_2 \in T && \text{(by inductive assumption)} \\ &\text{iff } \psi_1 \wedge \psi_2 \in T && (T \text{ is deductively closed}). \end{aligned}$$

- ϕ is of the form $\exists x.\, \psi(x)$. Then

$$\begin{aligned} &\underline{A} \models \exists x.\, \psi(x) \\ &\text{iff } \underline{A} \models \psi(c) \text{ for some } c \in \rho && \text{(by construction)} \\ &\text{iff } \psi(c) \in T \text{ for some } c \in \rho && \text{(by inductive assumption)} \\ &\text{iff } \exists x.\, \psi(x) \in T. \end{aligned}$$

To see the final equivalence, note that $\psi(c) \in T$ implies that $\exists x.\psi(x) \in T$ by (Q3) and (MP). For the converse, we use the assumption that T admits Henkin witnesses in ρ.

This concludes the proof that $\underline{A} \models T$; by definition, $A = \rho/\!\sim = \{c^{\underline{A}} \mid c \in \rho\}$. □

PROPOSITION 5.4.8. *Let T be a consistent τ-theory. Then T can be extended to a finitely complete Henkin theory T^*.*

PROOF. We define an increasing sequence $\emptyset = \rho_0 \subseteq \rho_1 \subseteq \cdots$ of sets of constant symbols by introducing for every $(\tau \cup \rho_i)$-formula $\phi(x)$ a new constant symbol c_ϕ and setting

$$\rho_{i+1} := \rho_i \cup \{c_\phi \mid \phi(x) \text{ a } (\tau \cup \rho_i)\text{-formula}\}.$$

Let $\rho := \bigcup_{i \in \mathbb{N}} \rho_i$. Define $T_0 := T$, and for $i \in \mathbb{N}$ define the $(\tau \cup \rho_i)$-theory

$$T_{i+1} := T_i \cup \{\exists x.\phi(x) \Rightarrow \phi(c_\phi) \mid \phi(x) \text{ a } (\tau \cup \rho_i)\text{-formula}\}.$$

We prove by induction over i that T_i is consistent. For $i = 0$ this holds by assumption. For $i > 0$, it suffices by Corollary 5.3.3 to show that all finite subsets S of T_i are consistent. Also note that T_{i-1} is also consistent as a $(\tau \cup \rho_i)$-theory, as a consequence of Lemma 5.4.3. Then Lemma 5.4.4 and an induction over the size of S show that $T_{i-1} \cup S$ is a consistent $(\tau \cup \rho_i)$-theory.

Using the fact that the union of a chain of consistent theories is consistent, we can apply Zorn's lemma (Theorem 4.1.5) to the set of consistent $(\tau \cup \rho)$-theories that contain $\bigcup_{i \in \mathbb{N}} T_i$, partially ordered by inclusion, and obtain a maximal consistent $(\tau \cup \rho)$-theory T^* which contains $\bigcup_{i \in \mathbb{N}} T_i$. We show that T^* is finitely complete. Let ϕ

be some $(\tau \cup \rho)$-sentence. If $T^* \nvdash_{\tau\cup\rho} \phi$, then $T^* \cup \{\neg\phi\}$ is consistent by Corollary 5.3.5. Maximality then implies that $\neg\phi \in T^*$, which finishes the proof. □

Proof of Theorem 5.4.2. If T has a model, then it is consistent. Conversely, suppose that T is consistent. By Proposition 5.4.8, T is contained in a finitely complete Henkin theory T^*. By Lemma 5.4.7, T^* has a model. □

Exercises.

(54) Prove Theorem 5.4.2 using the completeness theorem (Theorem 5.4.1).
(55) Show that if τ is countable, then there is a proof of Proposition 5.4.8 and hence a proof of Theorem 5.4.1 which does not require Zorn's lemma, and can be carried out in ZF (without the axiom of choice).

5.5. Compactness

Gödels completeness theorem implies the compactness theorem of first-order logic, which has found many applications in mathematics, e.g. in topology, set theory, and algebra. It is one of the most often used theorems in model theory.

Theorem 5.5.1. *A theory T is satisfiable if and only if T' is satisfiable for all finite $T' \subseteq T$.*

Proof. The completeness theorem (Theorem 5.4.1) shows that T is satisfiable if and only if T is not inconsistent. Since proofs are finite, this is the case if and only if all finite subsets of T' of T are not inconsistent, i.e., satisfiable. □

Remark 5.5.2. The name *compactness theorem* comes from the fact that the compactness theorem is equivalent to the statement that the following natural topological space is compact: the space is the set $\mathcal{T}(\tau)$ of all complete τ-theories, and the basic open sets are the sets T_ϕ of the form $\{T \in \mathcal{T}(\tau) \mid \phi \in T\}$.

To see the equivalence, let $\mathcal{C}$ be a covering of $\mathcal{T}(\tau)$ by open subsets of $\mathcal{T}(\tau)$. We may assume that $\mathcal{C}$ is of the form $\{T_\phi \mid \phi \in S\}$ for some set S of τ-sentences. Note that $S' := \{\neg\phi \mid \phi \in S\}$ is unsatisfiable. The compactness theorem of first-order logic implies that there is a finite subset F of S' which is unsatisfiable. But then $\{T_{\neg\phi} \mid \phi \in F\}$ is a finite subset of $\mathcal{C}$ covering $\mathcal{T}(\tau)$, showing that $\mathcal{T}(\tau)$ is compact.

Conversely, suppose that $\mathcal{T}(\tau)$ is compact, and that the τ-theory S is inconsistent. Then $\{T_{\neg\phi} \mid \phi \in S\}$ is an open covering of $\mathcal{T}(\tau)$. So by compactness it has a finite subcovering, i.e., there is a finite subset F of S such that $\bigcup\{T_{\neg\phi} \mid \phi \in F\} = \mathcal{T}(\tau)$. Hence, F is inconsistent, which is the statement of the compactness theorem. □

The following corollaries present well-known consequences of the compactness theorem.

Corollary 5.5.3. *Let T be a first-order theory with arbitrarily large finite models. Then T has an infinite model.*

Proof. By assumption, every finite subset of

$$T' := T \cup \Big\{\exists x_1, \dots, x_k \bigwedge_{1\le i<j\le k} x_i \neq x_j \mid k \in \mathbb{N}\Big\}$$

has a model. By the compactness theorem, T' has a model, and every model of T' must be infinite. □

Recall that a graph G is k-colourable if and only if it has a homomorphism to K_k, the clique with k vertices (Examples 9 and 10).

COROLLARY 5.5.4. *A graph $\underline{G} = (V; E)$ is k-colourable if and only if all its finite subgraphs are k-colourable.*

PROOF. Let τ be the signature $\{f, c_1, \dots, c_k\} \cup \{c_v \mid v \in V\}$ where f is a unary function symbol and all other symbols are distinct constant symbols. Consider the following τ-theory T.

$$\begin{aligned} T :=& \{f(c_u) \neq f(c_v) \mid (u, v) \in E^{\underline{G}}\} \\ & \cup \{\bigwedge_{i \neq j} c_i \neq c_j, \forall x \bigvee_{i \in \{1,\dots,k\}} f(x) = c_i\} \end{aligned}$$

By assumption, every finite subset of T is satisfiable. By the compactness theorem, T has a model $\underline{M}$, and $f^{\underline{M}}$ gives the desired k-colouring of $\underline{G}$. □

An *unfriendly partition* of a graph (V, E) is a partition of the vertices $V = V_1 \uplus V_2$ such that for every $i \in \{1, 2\}$, every $x \in V_i$ has at least as many neighbours in V_{3-i} as in V_i. Every finite graph has an unfriendly partition since every partition $V = V_1 \uplus V_2$ with maximally many edges between V_1 and V_2 is unfriendly.

CONJECTURE 5.1 (Cowan and Emerson, unpublished). *Every countable graph has an unfriendly partition.*

The conjecture is false for uncountable graphs (Shelah and Milner [**24**]). But the conjecture is true for *locally finite* graphs: a graph $(V; E)$ is *locally finite* if for every vertex $x \in V$ the set of neighbours $N(x) := \{y \in V \mid (x, y) \in E\}$ is finite.

COROLLARY 5.5.5. *Every locally finite graph $\underline{G}$ has an unfriendly partition.*

PROOF. Let $\{c_v \mid v \in G\}$ be a set of pairwise distinct constant symbols and P a unary relation symbol. Let $\phi_{v,n}$ be the sentence

$$\left(\bigwedge_{\text{distinct } u_1,\dots,u_n \in N(u)} \bigwedge_{i \leq n} P(c_{u_i}) \right) \Rightarrow \bigvee_{\text{distinct } v_1,\dots,v_n \in N(u)} \bigwedge_{i \leq n} \neg P(c_{v_i}).$$

Analogously, there is a first-order sentence $\psi_{v,n}$ expressing that if c_v has n neighbours in $\underline{G}$ in the complement of P, then it has in $\underline{G}$ also n neighbours in P. Consider the theory

$$T := \{\phi_{v,n} \mid v \in G, n \in \mathbb{N}\} \cup \{\psi_{v,n} \mid v \in G, n \in \mathbb{N}\}.$$

Observe that every finite subset of T has a model, since every finite graph has an unfriendly partition (Why?!). The compactness theorem therefore implies that T has a model $\underline{M}$. Then $G_1 := P^{\underline{M}} \cap G$ and $G_2 := G \setminus G_1$ is an unfriendly partition of $\underline{G}$. □

EXAMPLE 23. Note that every linearly ordered group $\underline{G}$ (see Example 7) is *torsion-free*, i.e., if $a \in G$ is such that $a^m = e$ for some $m \in \mathbb{N}$, then $a = e$. Conversely, a torsion-free abelian group $\underline{G}$ can be expanded to a linearly ordered group (Levi). To see this, let T be the theory of torsion-free ordered abelian groups; note that the definition of torsion-freeness above, and the definition of ordered abelian groups can be phrased using *universal* first-order sentences (i.e., a sentence that starts with universal quantifiers in front of a quantifier-free part), so we assume that T is universal. Let S be the set of all atomic first-order sentences that hold in $\underline{G}_G$ (see Section 3.1.4). Note that the $\{+, -, 0, <\}$-reduct of a model of $S \cup T$ is an ordered abelian group $\underline{H}$, and that G is the domain of a substructure $\underline{G}'$ of $\underline{H}$. Since T is universal, $\underline{G}'$ satisfies all sentences of T, too, and hence $\underline{G}'$ is an expansion of $\underline{G}$ which is an ordered abelian group, as desired. So all we have to prove is that $T \cup S$ is satisfiable. By the compactness theorem, it suffices to show that $T \cup F$ is satisfiable for every finite subset F of S.

Let F be a finite subset of S. Only finitely many constant symbols can be mentioned in F; consider the smallest subgroup $\underline{S}$ of $\underline{G}$ that contains the elements denoted by these constants. Then $\underline{S}$ is a finitely generated abelian group; those groups are classified, by the *fundamental theorem for finitely generated abelian groups*. Since $\underline{S}$ is torsion-free, they must be of the form $(\mathbb{Z}^n; +, -, 0)$. Hence, $\underline{S}$ can be linearly ordered, for example by the *lexicographic ordering*, defined by $(x_1, \ldots, x_n) < (y_1, \ldots, y_n)$ if $x_1 = y_1, \ldots, x_{i-1} = y_{i-1}$ and $x_i < y_i$, for some $i \in \{1, \ldots, n\}$. This shows that $T \cup F$ is satisfiable. △

Exercises.

(56) Let T be a first-order theory and $\phi(x)$ a formula. Show that if T has for every $n \in \mathbb{N}$ a model $\underline{A}$ with $|\phi^{\underline{A}}| \geq n$, then T has a model $\underline{A}$ such that $\phi^{\underline{A}}$ is infinite.

(57) Show that the compactness theorem does not hold if we allow infinite disjunctions as sentences.

(58) Show that the compactness theorem for first-order logic implies the compactness theorem for propositional logic: if S is a set of propositional formulas such that every finite subset of S is satisfiable, then all of S is satisfiable, i.e., there is a mapping from the variables that appear in formulas from S to $\{0, 1\}$ which satisfies all formulas in S.

(59) Let $\underline{G}$ be a 2-colorable graph with color classes $A, B \subseteq G$ such that every vertex in A has only finitely many neighbours. Suppose that every finite subset A' of A has a *matching*, i.e., a subset M of the edges of $\underline{G}$ such that any two distinct edges in M are disjoint and every $a \in A'$ appears in an edge of M. Show that then all of A has a matching.

(60) Show that the compactness theorem is equivalent to the following statement. If a first-order theory T has the same models as a single first-order sentence ϕ, there is already a finite subset of T which has the same models as ϕ.

(61) (Exercise 2.2.3 in [**27**]) A class $\mathcal{C}$ of τ-structures is called
- *elementary* if there exists a first-order τ-theory T such that $\mathcal{C}$ are precisely the models of T;
- *finitely axiomatisable* if it is the class of models of a finite theory.

Show that $\mathcal{C}$ is finitely axiomatisable if and only if $\mathcal{C}$ is an elementary class and the complement of $\mathcal{C}$ is an elementary class.

(62) (Exercise 2.2.5 in [**27**]) Let T be a τ_{Ring}-theory containing T_{Field}. Show that
- if T has models of arbitrarily large characteristic, then it has a model of characteristic 0.
- The class of fields of characteristic 0 is not finitely axiomatisable.

CHAPTER 6

Computability

This chapter presents the basics of the theory of computable functions (also called *recursive functions*); the results will be important for proving the incompleteness theorems in the next chapter. We follow the classical approach to first introduce the class of *primitive recursive functions*, which is already quite large, but does not contain some recursive functions that grow too fast for being primitive recursive. The Ackermann function is an example of such a function. The recursive functions extend the primitive recursive functions by so-called μ-recursion.

Recursive functions can be characterised in many different ways. A particularly useful characterisation is via an abstract machine model, *Turing machines*. Turing machines can be represented by bit strings, and one can show that there is a *universal Turing machine* that, given a string that describes a Turing machine, simulates the computation of that Turing machine for any given input. We will also present a concrete function that is not recursive, derived from the Halting problem for Turing machines. It will then be easy to also prove the first incompleteness theorem of Gödel in the next chapter.

6.1. Primitive Recursion

For $n \in \mathbb{N}$, we define $\mathscr{O}^{(n)} := \{f \colon \mathbb{N}^n \to \mathbb{N}\}$ and $\mathscr{O} := \bigcup_{n\in\mathbb{N}} \mathscr{O}^{(n)}$.

Definition 6.1.1. A subset $\mathscr{C}$ of $\mathscr{O}$ is called a *clone* if

- $\mathscr{C}$ contains for every $n \in \mathbb{N}^+$ and every $i \in \{1, \dots, n\}$ the i-th projection of arity n, i.e., the operation $\pi_i^n \in \mathscr{O}^{(n)}$ defined by $(x_1, \dots, x_n) \mapsto x_i$.
- $\mathscr{C}$ is *closed under composition*: if $f_1, \dots, f_n \in \mathscr{C} \cap \mathscr{O}^{(p)}$ and $g \in \mathscr{C} \cap \mathscr{O}^{(n)}$, then the operation $g(f_1, \dots, f_n) \in \mathscr{O}^{(p)}$ given by

  $$(x_1, \dots, x_p) \mapsto g\big(f_1(x_1, \dots, x_p), \dots, f_n(x_1, \dots, x_p)\big)$$

  is also contained in $\mathscr{C}$.

We also write $\mathscr{C}^{(n)}$ for $\mathscr{C} \cap \mathscr{O}^{(n)}$.

6.1.1. Primitive recursive functions. The set of *primitive recursive* functions $\mathscr{P}$ is the smallest subset of $\mathscr{O}$ which satisfies the following properties:

- $\mathscr{P}$ is a clone;
- $\mathscr{P}$ contains the 0-ary constant operation $c_0^0 \in \mathscr{O}^{(0)}$ that is constant 0;
- $\mathscr{P}$ contains the *successor function* $s \in \mathscr{O}^{(1)}$ defined by $x \mapsto x + 1$;
- $\mathscr{P}$ is *closed under recursion*: if $g \in \mathscr{P}^{(n)}$ and $h \in \mathscr{P}^{(n+2)}$, then the operation $\rho(g, h) := f \in \mathscr{O}^{(n+1)}$ defined by

  $$f(x_1, \dots, x_n, 0) := g(x_1, \dots, x_n)$$
  $$f(x_1, \dots, x_n, y+1) := h\big(x_1, \dots, x_n, y, f(x_1, \dots, x_n, y)\big)$$

  is also in $\mathscr{P}$.

Formally, $\mathscr{P}$ is the intersection over all subsets of $\mathscr{O}$ that have the closure properties above; the definition of $\mathscr{P}$ makes sense because the intersection then also has the closure properties.

We use the following compact notation for specifying operations in $\mathscr{O}$: the function of arity n that maps $(x_1, \dots, x_n)$ to $f(x_1, \dots, x_n)$ is denoted by $\lambda x_1, \dots, x_n. f$. For example, the i-th projection of arity n equals $\lambda x_1, \dots, x_n. x_i$, and the successor function equals $\lambda x. x + 1$.

LEMMA 6.1.2. *The operations $\lambda x, y. x + y$ and $\lambda x, y. x \cdot y$ are primitive recursive.*

PROOF. Let $h := s(\pi^3_3)$ and $g = \pi^1_1$. Then $\rho(g, h) = \lambda x, y. x+y$, because $x+0 = x$ and $x + (y + 1) = (x + y) + 1$. The proof for $\lambda x, y. x \cdot y$ is similar. □

DEFINITION 6.1.3. The operation $\dot{-} \colon \mathbb{N}^2 \to \mathbb{N}$ is defined as follows.

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

LEMMA 6.1.4. *The operation $\dot{-}$ is primitive recursive.*

PROOF. We first construct $\lambda y. y \dot{-} 1$ using $0 \dot{-} 1 = 0$ and $(y + 1) \dot{-} 1 = y$ by recursion. Then $\dot{-}$ can be constructed using $x \dot{-} 0 = x$ and $x \cdot (y + 1) = (x \dot{-} y) \dot{-} 1$ by recursion. □

LEMMA 6.1.5. *If $f \in \mathscr{P}^{(n+1)}$, then $\lambda x_1, \dots, x_n, y. \sum_{t=0}^{y} f(\bar{x}, t) \in \mathscr{P}^{(n)}$.*

PROOF. By recursion. □

Exercises.

(63) Show that for every constant $k \in \mathbb{N}$ the operation $c^n_k := \lambda x_1, \dots, x_n.k \in \mathscr{O}^{(n)}$ that always returns k is primitive recursive.
(64) Show that the smallest clone that contains s and c^0_0 is not closed under recursion.
(65) Show that $\lambda x. x!$ and $\lambda x, y. x^y$ are primitive recursive.

6.1.2. Primitive recursive sets. If $X \subseteq \mathbb{N}^n$ then the *characteristic function for X*, denoted by 1_X, is the function defined by

$$1_X(x_1, \dots, x_n) := \begin{cases} 1 & \text{if } (x_1, \dots, x_n) \in X; \\ 0 & \text{else.} \end{cases}$$

A subset X of $\mathbb{N}^n$ is called *primitive recursive* if its characteristic function 1_X is primitive recursive.

LEMMA 6.1.6. *The set $\{(x, y) \mid x < y\} \subseteq \mathbb{N}^2$ is primitive recursive.*

PROOF. We have $1_<(x, y) = 1 \dot{-} (1 \dot{-} (y \dot{-} x))$. □

LEMMA 6.1.7. *The set of primitive recursive subsets of $\mathbb{N}^n$ is closed under $\cup$, $\cap$, and complementation.*

PROOF. Let X be a primitive recursive subset of $\mathbb{N}^n$. Then $1_{\mathbb{N}^n \setminus X} = 1 \dot{-} 1_X$, and hence $\mathbb{N}^n \setminus X$ is primitive recursive. If X and Y are primitive recursive subsets of $\mathbb{N}^n$, then $1_{X \cap Y} = 1_X \cdot 1_Y$ and hence $X \cap Y$ is primitive recursive; this proves the lemma since $X \cup Y = \mathbb{N}^n \setminus \big((\mathbb{N}^n \setminus X) \cap (\mathbb{N}^n \setminus Y)\big)$. □

LEMMA 6.1.8 (Definition by cases). *Let $\mathbb{N}^n = A_1 \uplus \cdots \uplus A_k$ be a partition of $\mathbb{N}^n$ into primitive recursive sets, and let $f_1, \dots, f_k \in \mathscr{P}^{(n)}$. Then the function f defined by $f(\bar{x}) := f_i(\bar{x})$ if $\bar{x} \in A_i$, for all $i \in \{1, \dots, k\}$, is primitive recursive.*

PROOF. One has $f = 1_{A_1} \cdot f_1 + \cdots + 1_{A_k} \cdot f_k$. □

We will see later (Exercise 76) that if $X \subseteq \mathbb{N}^{n+1}$ is primitive recursive, then

$$\{(x_1, \ldots, x_n) \in \mathbb{N}^n \mid \text{ there exists } x_{n+1} \text{ such that } (x_1, \ldots, x_n, x_{n+1}) \in X\}$$

need not be primitive recursive. However, the following lemma shows that the primitive recursive sets are closed under a bounded version of quantification.

LEMMA 6.1.9. *If $X \subseteq \mathbb{N}^{n+1}$ is a primitive recursive set, so are the sets*

$$X_\exists := \{(x_1, \ldots, x_n, z) \in \mathbb{N}^{n+1} \mid \textit{there exists } t \leq z \textit{ such that } (\bar{x}, t) \in X\}$$
$$\textit{and } X_\forall := \{(x_1, \ldots, x_n, z) \in \mathbb{N}^{n+1} \mid \textit{for all } t \leq z \textit{ we have } (\bar{x}, t) \in X\}.$$

PROOF. By Lemma 6.1.7, it is enough to treat the first case. The operation $\lambda x_1, \ldots, x_n, z \sum_{t=0}^{z} 1_X(\bar{x}, t)$ is primitive recursive by Lemma 6.1.5. Note that $1_{X_\exists} = 1$ if $\sum_{t=0}^{z} 1_X(\bar{x}, t) \geq 1$ and $1_{X_\exists}(\bar{x}, z) = 0$ otherwise. Hence, the statement follows by Lemma 6.1.8. □

Exercises.

(66) Show that the binary operation $1_=$ is primitive recursive.
(67) Prove that the set of primitive recursive subsets of $\mathbb{N}$ is not closed under infinite intersections and unions.

6.1.3. Bounded μ-recursion. Let $X \subseteq \mathbb{N}^{n+1}$. Then the function $f\colon \mathbb{N}^{n+1} \to \mathbb{N}$ defined by

$$f(\bar{x}, z) := \begin{cases} 0 & \text{if there is no } t \leq z \text{ with } (\bar{x}, t) \in X \\ t & \text{if } t \in \{0, \ldots, z\} \text{ is smallest such that } (\bar{x}, t) \in X \end{cases}$$

is denoted by

$$(\mu t \leq z) X(\bar{x}, t).$$

LEMMA 6.1.10. *Let $X \subseteq \mathbb{N}^{n+1}$ be primitive recursive. Then $f := (\mu t \leq z) X(\bar{x}, t)$ is primitive recursive as well.*

PROOF. We have $f(\bar{x}, 0) = 0$ and

$$f(\bar{x}, z+1) = \begin{cases} f(\bar{x}, z) & \text{if } \sum_{t=0}^{z} 1_X(\bar{x}, t) \geq 1, \\ z+1 & \text{if } \sum_{t=0}^{z} 1_X(\bar{x}, t) = 0 \text{ and } (\bar{x}, z+1) \in X, \\ 0 & \text{otherwise.} \end{cases}$$

The statement then follows from Lemma 6.1.8 and recursion. □

EXAMPLE 24. Let $q\colon \mathbb{N}^2 \to \mathbb{N}$ be the function which maps (x, y) to the integer part of x/y if $y \neq 0$, and to 0 otherwise. Then q is primitive recursive; indeed, $q(x, y) = (\mu t \leq x)((t+1) \cdot y > x)$. △

6.1.4. Gödel numbers. To formulate more complex functions we need a coding trick, due to Gödel. We start with the easy observation that pairs of natural numbers can be coded by a single natural number, in the following sense.

LEMMA 6.1.11. *There exists a bijection $\alpha_n\colon \mathbb{N}^n \to \mathbb{N}$ which is primitive recursive. Moreover, there are primitive recursive functions $\beta_1^n, \ldots, \beta_n^n\colon \mathbb{N} \to \mathbb{N}$ such that the function $x \mapsto \alpha_n(\beta_1^n(x), \ldots, \beta_n^n(x))$ is the identity on $\mathbb{N}$.*

PROOF. We show the statement for $n = 2$; the general case can be shown by induction. The function α given by $(x, y) \mapsto \frac{(x+y+1)(x+y)}{2} + x$ from Example 20 is clearly primitive recursive (using Example 24). Define

$$\beta_1^2 := \lambda x(\mu z \leq x)(\exists t \leq x)(\alpha_2(z, t) = x)$$
$$\text{and } \beta_2^2 := \lambda y(\mu z \leq y)(\exists t \leq y)(\alpha_2(t, z) = y).$$

Since $\alpha_2(x, y) \geq \min(x, y)$, for every $x \in \mathbb{N}$ we have $\alpha_2(\beta_1^2(x), \beta_2^2(x)) = x$. □

The coding trick of Gödel is based on representing arbitrarily long finite sequences of natural numbers as natural numbers. We first need some preparatory lemmas.

LEMMA 6.1.12. *The divisibility relation*

$$D := \{(x, y) \in \mathbb{N}^2 : y|x\} = \{(x, y) \in \mathbb{N}^2 : \text{ there exists } z \in \mathbb{N} \text{ such that } y \cdot z = x\}$$

is primitive recursive.

PROOF. First define the modulo function $m \colon \mathbb{N}^2 \to \mathbb{N}$ recursively by $m(0, y) = 0$ and

$$m(x + 1, y) = \begin{cases} 0 & \text{if } m(x, y) + 1 = y \\ m(x, y) + 1 & \text{otherwise.} \end{cases}$$

It follows from Lemma 6.1.8 that $m \in \mathscr{P}$. Then $1_D(x, y) = 1 \dot{-} m(x, y)$. □

LEMMA 6.1.13. *The set $P \subseteq \mathbb{N}$ of prime numbers is primitive recursive. The function $\pi \colon \mathbb{N} \to \mathbb{N}$ that maps $n \in \mathbb{N}$ to the n-th prime number is primitive recursive.*

PROOF. We have that x is a prime number if and only if it satisfies

$$x \geq 2 \wedge \forall y \leq x \big(y|x \Rightarrow (y = 1 \vee y = x)\big).$$

It follows from Lemma 6.1.6, Lemma 6.1.7, and bounded quantification (Lemma 6.1.9) that the set of all natural numbers that satisfy this formula is primitive recursive. For the second statement, note that $\pi(0) = 2$ and that

$$\pi(x + 1) = (\mu x \leq \pi(x)! + 1)\big(z > \pi(x) \wedge P(z)\big).$$ □

DEFINITION 6.1.14. Let $(x_0, \ldots, x_{n-1})$ be a finite sequence of natural numbers. Then the *Gödel number* of $(x_0, \ldots, x_{n-1})$ is defined to be

$$\langle x_0, \ldots, x_{n-1} \rangle := \pi(0)^{x_0} \cdot \cdots \cdot \pi(n - 2)^{x_{n-2}} \cdot \pi(n - 1)^{x_{n-1}+1} - 1.$$

LEMMA 6.1.15. *The map $\langle . \rangle$ defines a bijection between the set of finite sequences of natural numbers and $\mathbb{N}$ and has the following properties:*

(1) The length function *ℓ defined by*

$$\ell(\langle x_0, \ldots, x_{n-1} \rangle) = n$$

is primitive recursive.

(2) The binary component function *$(x)_i$ defined by*

$$(\langle x_0, \ldots, x_{n-1} \rangle)_i := \begin{cases} x_i & \text{if } i < n, \\ 0 & \text{else .} \end{cases}$$

is primitive recursive.

$\xi(n,x)$	0	1	2	3	4	x
0	1	2	4	8	16	
1	1	2	4	8	16	
2	1	2	4	8	16	
3	1	2	4	16	2^{16}	
4	1	2	4	2^{16}	$2^{2^{16}}$	
n						

PROOF. The set

$$X := \big\{(x,y) \mid \neg\pi(z)|(x+1) \text{ for all } z \in \{y,\dots,x\}\big\}$$

is primitive recursive by Lemmas 6.1.12, 6.1.13, 6.1.6, 6.1.7, and 6.1.9. The function ℓ can be written as

$$\ell(x) = \begin{cases} 0 & \text{if } x = 0 \\ (\mu y \le x)\, X(x,y) & \text{if } x > 0 \end{cases}$$

and hence is primitive recursive by Lemma 6.1.8.

To prove (2), consider the primitive recursive set $Y := \{(i,x,y) \mid \neg\pi(x)^y|(x+1)\}$. Then the function $(x)_i$ can be written as

$$(x)_i := \begin{cases} 0 & \text{if } i+1 > \ell(x) \\ (\mu y \le x)\, Y(i,y+2,x) & \text{if } i+1 = \ell(x) \\ (\mu y \le x)\, Y(i,y+1,x) & \text{if } i+1 < \ell(x) \end{cases}$$

and hence is primitive recursive by Lemma 6.1.8. □

6.1.5. The Ackermann function. We present a famous example of an operation which is not primitive recursive, the *Ackermann function* $\xi \in \mathscr{O}^{(2)}$. It is the simplest and earliest-discovered function of this type. The original function had three arguments; we present a modification of a formulation of Péter that is used by Hils and Loeser **[10]** and by Cori and Lascar **[21]**; the motivation of the modification is to simplify the proof that the Ackermann function is not primitive recursive.

DEFINITION 6.1.16. The Ackermann function is defined inductively as follows.

- $\xi(0,x) := 2^x$
- $\xi(n,0) := 1$
- $\xi(n+1,x+1) := \xi(n,\xi(n+1,x))$.

This is well-defined, since the lexicographic order of $\mathbb{N}^2$ is a well-ordering (in the inductive step, either the first argument decreases, or the first argument remains equal, in which case the second argument decreases). In the following we write $\xi_n(x)$ for $\xi(n,x)$ and first prove some simple properties of $\xi_n\colon \mathbb{N} \to \mathbb{N}$.

LEMMA 6.1.17. *Let $n,x \in \mathbb{N}$. Then $\xi_n(x) > x$. If $y > x$, then $\xi_n(y) > \xi_n(x)$. Moreover, $\xi_{n+1}(x) \ge \xi_n(x)$.*

PROOF. We prove the first and second statement by induction on n. For $n = 0$, we have $\xi_0(x) = \xi(0,x) = 2^x > x$, and for $y > x$ we have $\xi_0(y) = 2^y > 2^x = \xi_0(x)$. Now suppose that the statements hold for n; then they imply

$$\xi_{n+1}(x+1) = \xi_n(\xi_{n+1}(x)) > \xi_{n+1}(x)$$

which shows the second statement for $n+1$. The first statement for $n+1$ follows from the second statement: $\xi_{n+1}(x) > \xi_{n+1}(x-1) > \cdots > \xi_{n+1}(0) = 1$, so $\xi_{n+1}(x) > x$.

To prove the the third statement, note that $\xi_{n+1}(0) = 1 = \xi_n(0)$ and that

$$\begin{aligned}\xi_{n+1}(x+1) &= \xi_n(\xi_{n+1}(x)) && \text{(by the definition of } \xi_{n+1}\text{)}\\ &\geq \xi_n(x+1) && \text{(by the second statement)}\end{aligned}$$

where the second statement of the lemma can be applied because $\xi_{n+1}(x) > x$ by the first statement. □

Define $\xi_n^k := \underbrace{\xi_n \circ \cdots \circ \xi_n}_{k \text{ times}}$ for $k \geq 1$ and $\xi_n^0(x) := x$.

LEMMA 6.1.18. *For all $k, n, x \in \mathbb{N}$ we have $\xi_n^k(x) \leq \xi_{n+1}(x+k)$.*

PROOF. Our proof is by induction on k. The base case $k = 0$ follows from Lemma 6.1.17. Now suppose that the result holds for k. Then

$$\xi_n^{k+1}(x) = \xi_n(\xi_n^k(x)) \leq \xi_n(\xi_n(x+k)) \leq \xi_n(\xi_{n+1}(x+k)) = \xi_{n+1}(x+k+1). \quad \square$$

Note that for each fixed n, the function ξ_n is primitive recursive (Exercise 69). However, the function ξ is not. To prove this, we need a better understanding of the maximal growth of primitive recursive functions.

DEFINITION 6.1.19. A function $f \in \mathscr{O}^{(1)}$ *dominates* a function $g \in \mathscr{O}^{(k)}$ if there exists $\ell \in \mathbb{N}$ such that for all $(x_1, \ldots, x_k) \in \mathbb{N}^k$

$$g(x_1, \ldots, x_k) \leq f\big(\max(x_1, \ldots, x_k, \ell)\big).$$

For $n \in \mathbb{N}$, we denote by $\mathscr{B}_n$ the set of all operations in $\mathscr{O}$ that are dominated by ξ_n^k for at least one $k \in \mathbb{N}$.

LEMMA 6.1.20. *For every $n \in \mathbb{N}$ the set $\mathscr{B}_n$ is a clone.*

PROOF. Let $f_1, \ldots, f_m \in \mathscr{B}_n^{(k)}$ and $g \in \mathscr{B}_n^{(m)}$. Then there are $\ell_0, \ell_1, \ldots, \ell_m$, $k_0, k_1, \ldots, k_m \in \mathbb{N}$ such that for all $x_1, \ldots, x_n \in \mathbb{N}$ and $i \in \{1, \ldots, m\}$

$$\begin{aligned}g(x_1, \ldots, x_m) &\leq \xi_n^{k_0}(\max(x_1, \ldots, x_m, \ell_0))\\ \text{and } f_i(x_1, \ldots, x_k) &\leq \xi_n^{k_i}(\max(x_1, \ldots, x_k, \ell_i)).\end{aligned}$$

Define $\ell := \max(\ell_0, \ell_1, \ldots, \ell_m)$ and $k := k_0 + \max(k_1, \ldots, k_m)$. Let $\bar{x} \in \mathbb{N}^k$. Then

$$\begin{aligned}g\big(f_1(\bar{x}), \ldots, f_m(\bar{x})\big) &\leq g\big(\xi_n^{k_1}(\max(\bar{x}, \ell_1)), \ldots, \xi_n^{k_m}(\max(\bar{x}, \ell_m))\big)\\ &\leq \xi_n^{k_0}\big(\max(\xi_n^{k_1}(\max(\bar{x}, \ell_1)), \ldots, \xi_n^{k_m}(\max(\bar{x}, \ell_m), \ell_0)\big)\\ &\leq \xi_n^{k_0}(\xi_n^{\max(k_1, \ldots, k_m)}(\max(\bar{x}, \ell_0, \ell_1, \ldots, \ell_m)))\\ &\leq \xi_n^{k}(\max(\bar{x}, \ell))\end{aligned}$$

which proves that $\mathscr{B}_n$ is closed under composition. □

LEMMA 6.1.21. *Let $g \in \mathscr{B}_n^{(p)}$ and $h \in \mathscr{B}_n^{(p+2)}$. Then $\rho(g, h) \in \mathscr{B}_{n+1}^{(p)}$. In particular,*

$$\mathscr{P} \subseteq \bigcup_{n \in \mathbb{N}} \mathscr{B}_n.$$

PROOF. By assumption, there are $k_1, \ell_1, k_2, \ell_2 \in \mathbb{N}$ such that for all $x_1, \ldots, x_p \in \mathbb{N}$

$$\begin{aligned}g(\bar{x}) &\leq \xi_n^{k_1}(\max(x_1, \ldots, x_p, \ell_1))\\ h(\bar{x}, y, t) &\leq \xi_n^{k_2}(\max(x_1, \ldots, x_p, y, t, \ell_2)).\end{aligned}$$

By induction on $y \in \mathbb{N}$ one can show that

$$\rho(g, h)(\bar{x}, y) \leq \xi_n^{k_1 + y k_2}(\max(x_1, \ldots, x_p, y, \ell_1, \ell_2))$$

which is at most $\xi_{n+1}(\max(x_1, \ldots, x_p, y, \ell_1, \ell_2) + k_1 + yk_2)$ by Lemma 6.1.18. The upper bound is in $\mathscr{B}_{n+1}$ by Lemma 6.1.20, which proves the claim. □

The following result was conjectured by Hilbert and proved by Ackermann.

THEOREM 6.1.22. *The Ackermann function ξ is not primitive recursive.*

PROOF. Suppose for contradiction that ξ is primitive recursive. Then $\lambda x.\, \xi_x(2x)$ is primitive recursive as well and by Lemma 6.1.21 there exist $n, k, \ell \in \mathbb{N}$ such that for all $x \in \mathbb{N}$ with $x \geq \ell$

$$\xi_x(2x) \leq \xi_n^k(x).$$

For $x > \max(k, n+1, \ell)$ we thus obtain

$$\begin{aligned} \xi_{n+1}(x+k) < \xi_x(2x) &\leq \xi_n^k(x) \\ &\leq \xi_{n+1}(x+k) \qquad \text{(by Lemma 6.1.18),} \end{aligned}$$

a contradiction. □

REMARK 6.1.23. The so-called *inverse Ackermann function*, often denoted by α, is not strictly speaking the inverse of the Ackermann function, but roughly speaking it grows as slowly as the Ackermann function grows quickly. It has many applications in the analysis of algorithms (for example for the disjoint-set data structure). One may assume that $\alpha(n) \leq 5$ for any practical input size n.

Exercises.

(68) The *Fibonacci function* $f \in \mathscr{O}^{(1)}$ is defined via $f(0) := 0$, $f(1) := 1$, and $f(n+2) := f(n+1) + f(n)$ for all $n \in \mathbb{N}$. Show that f is primitive recursive.

(69) Show that for every fixed n the operation ξ_n is primitive recursive.

(70) (from Hils and Loeser [**10**]) The set of *elementary recursive functions* $\mathscr{E}$ is the smallest clone that contains c_0^0, addition $+$, multiplication $\cdot$, and the binary $\dot{-}$ from Definition 6.1.3, and that is closed under *bounded sum* and *bounded product*: if $f \in \mathscr{E}^{(n+1)}$ then the operations

$$(x_1, \ldots, x_n, x) \mapsto \sum_{i=0}^{x} f(x_1, \ldots, x_n, i)$$

$$(x_1, \ldots, x_n, x) \mapsto \prod_{i=0}^{x} f(x_1, \ldots, x_n, i)$$

are contained in $\mathscr{E}^{(n)}$. Show that the following operations are elementary recursive:

- for every $n, k \in \mathbb{N}$, the constant operation c_k^n;
- $\lambda x.\, 2^x$.
- $\lambda x, y.\, x^y$.

(71) (from Hils and Loeser [**10**]) Let $t \in \mathscr{O}^{(2)}$ be the operation defined by $t(m, 0) := m$ and $t(m, n+1) = 2^{t(m,n)}$.

- Prove that t is primitive recursive.
- Prove that for every $n \in \mathbb{N}$ the function $t_n := \lambda x.\, t(x, n)$ is strictly increasing.
- Prove that for every elementary recursive function f there exists $n \in \mathbb{N}$ such that f is dominated by t_n.
- Prove that t is not elementary recursive.

6.2. Recursive Functions

The Ackermann function illustrates that there are functions that can be computed (in a still informal sense) but that are not primitive recursive; we need to strengthen our recursion mechanism to also capture such functions.

DEFINITION 6.2.1 (total μ-operator). Let $g\colon \mathbb{N}^{n+1} \to \mathbb{N}$ be such that for all $x_1, \dots, x_n \in \mathbb{N}$ there exists $y \in \mathbb{N}$ with $g(x_1, \dots, x_n, y) = 0$, then the function $\mu g\colon \mathbb{N}^n \to \mathbb{N}$ is defined as

$$(\mu g)(x_1, \dots, x_n) := \min\{y \in \mathbb{N} \mid g(x_1, \dots, x_n, y) = 0\}.$$

This definition has an important technical disadvantage: the definition only works for operations such that for all $x_1, \dots, x_n \in \mathbb{N}$ there exists $y \in \mathbb{N}$ with $g(x_1, \dots, x_n, y) = 0$. This restriction can become a problem later, because the condition is very difficult to check already if g is a primitive recursive function. We therefore work in the following with *partial operations*.

DEFINITION 6.2.2. A *partial operation of arity n over* $\mathbb{N}$ is a function $f\colon A \to \mathbb{N}$ for some $A \subseteq \mathbb{N}^n$. We also write $\mathrm{dom}(f)$ for A. If $\bar{t} \in \mathbb{N}^n \setminus A$, then we say that f *is undefined on* $\bar{t}$. A partial function is *total* if $\mathrm{dom}(f) = \mathbb{N}^n$.

DEFINITION 6.2.3 (general μ-operator). Let g be a partial operation of arity $n+1$ on $\mathbb{N}$. Then μg denotes the partial operation of arity n on $\mathbb{N}$ that is defined as follows. If $x_1, \dots, x_n \in \mathbb{N}$ are such that there exists $y \in \mathbb{N}$ with $g(\bar{x}, y) = 0$ and $(\bar{x}, z) \in \mathrm{dom}(g)$ for all $z \leq y$, then $\mu g(\bar{x})$ is defined to be the *minimal* such y; otherwise, $\mu g(\bar{x})$ is undefined.

If we have a set of partial operations over $\mathbb{N}$, then closure under composition and closure under recursion are defined analogously to the case of total operations (Definition 6.1.1):

- if g is a partial operation of arity n over $\mathbb{N}$ and $f_1, \dots, f_n$ are partial operations of arity m over $\mathbb{N}$, then $g(f_1, \dots, f_n)$ is the partial operation of arity m over $\mathbb{N}$ which maps $\bar{x} \in \mathbb{N}^m$ to $g(f_1(\bar{x}), \dots, f_n(\bar{x}))$ if $\bar{x} \in \mathrm{dom}(f_i)$ for all i and $(f_1(\bar{x}), \dots, f_n(\bar{x})) \in \mathrm{dom}(g)$, and is undefined otherwise.
- if g is a partial operation of arity p over $\mathbb{N}$ and h is a partial operation of arity $p+2$ over $\mathbb{N}$, then $\rho(g, h)$ is the partial operation of arity $p+1$ over $\mathbb{N}$ defined by
  - $f(\bar{x}, 0) := g(\bar{x})$ if $\bar{x} \in \mathrm{dom}(g)$, and otherwise f is not defined for $(\bar{x}, 0)$;
  - $f(\bar{x}, y+1) := h(\bar{x}, y, f(\bar{x}, y))$ if f is defined for $(\bar{x}, y)$ (this property is defined simultaneously by induction on y) and if $(\bar{x}, y, f(\bar{x}, y)) \in \mathrm{dom}(h)$, and otherwise f is not defined for $(\bar{x}, y+1)$.

DEFINITION 6.2.4. The set of *partial recursive functions* is defined to be the smallest set of partial operations over $\mathbb{N}$ that contains the basic operations c_0^0 and s, the projections, and is closed under composition, recursion, and μ-recursion. A *(total) recursive function* is a total function which is partial recursive. We write $\mathscr{R}$ for the clone of all recursive operations.

Note that $\mathscr{R}$ contains the set of primitive recursive operations $\mathscr{P}$.

Church's thesis is the (non-mathematical) statement that every partial operation over $\mathbb{N}$ which is computable in an intuitive sense is recursive.

Exercises.

(72) (*) Show that the Ackermann function is recursive. If this exercise is too difficult, read on, it will become an easy exercise later.

6.3. Recursively Enumerable Sets

DEFINITION 6.3.1. A set $X \subseteq \mathbb{N}^n$ is called

- *decidable* (or *recursive*) if 1_X is recursive;
- *recursively enumerable* (or *semi-decidable*) if there is a decidable set $Y \subseteq \mathbb{N}^{n+1}$ such that

$$X = \pi(Y) := \{(x_1, \ldots, x_n) \mid (x_1, \ldots, x_n, y) \in Y\}.$$

Every decidable set is recursively enumerable (Exercise 73), but the converse does not hold, as we will prove in Section 6.4.5 (Remark 6.4.24). The concept of recursively enumerable sets can be used to establish results about decidable sets, via the following theorem.

THEOREM 6.3.2 (Theorem of the Complement). *A set $X \subseteq \mathbb{N}^n$ is decidable if and only if both X and $\mathbb{N}^n \setminus X$ are recursively enumerable.*

PROOF. The forward implication is clear. For the converse, assume that $X \subseteq \mathbb{N}^n$ is such that there are decidable sets $Y, Y' \subseteq \mathbb{N}^{n+1}$ such that $X = \pi(Y)$ and $\mathbb{N}^n \setminus X = \pi(Y')$. Let $g := \lambda z_1, \ldots, z_n, t\,\big(1 - 1_Y(\bar{z}, t) - 1_{Y'}(\bar{z}, t)\big)$. Then $1_X(\bar{z}) = 1_Y\big(\bar{z}, (\mu g)(\bar{z})\big)$. □

The set of recursively enumerable sets is a rather robust set of subsets of $\mathbb{N}$, as we will see in the following.

PROPOSITION 6.3.3. *Let $f_1, \ldots, f_m \in \mathscr{R}^{(n)}$ and let $X \subseteq \mathbb{N}^n$ be recursively enumerable. Then*

$$X' := \{\bar{x} \in \mathbb{N}^m \mid (f_1(\bar{x}), \ldots, f_m(\bar{x})) \in X\}$$

is recursively enumerable as well.

PROOF. There exists a decidable set $Y \subseteq \mathbb{N}^{n+1}$ such that $X = \pi(Y)$. Note that $Y' := \{(\bar{x}, y) \in \mathbb{N}^{n+1} \mid (f_1(\bar{x}), \ldots, f_m(\bar{x}), y) \in Y\}$ is decidable. Then $X' = \pi(Y')$ and hence is recursively enumerable. □

PROPOSITION 6.3.4. *The set of recursively enumerable sets is closed under projection, intersection, and union.*

PROOF. To show closure under projections, suppose that $n \geq 1$ and let $R \subseteq \mathbb{N}^n$ be recursively enumerable, so that there exists $S \subseteq \mathbb{N}^{n+1}$ such that $R = \pi(S)$. Note that

$$S' := \{(\bar{x}, t) \in \mathbb{N}^n \mid (\bar{x}, \beta_1^2(t), \beta_2^2(t)) \in S\}$$

is decidable (where β_i^2 is the function from Lemma 6.1.11). Thus, $\pi(R) = \pi(\pi(S)) = \pi(S')$ is recursively enumerable.

To show closure under intersection and union, let $R_1, R_2 \subseteq \mathbb{N}^n$ be recursively enumerable; so there are decidable sets $S_1, S_2 \subseteq \mathbb{N}^{n+1}$ such that $R_i = \pi(S_i)$ for $i \in \{1, 2\}$. Define

$$S_i' := \big\{(\bar{x}, y) \in \mathbb{N}^{n+1} \mid (\bar{x}, \beta_i^2(y)) \in S_i\big\}.$$

We then have

$$R_1 \cup R_2 = \pi(S_1 \cup S_2)$$
$$R_1 \cap R_2 = \pi(S_1' \cap S_2').$$ □

EXAMPLE 25. Let $p(x_1, \ldots, x_n, y_1, \ldots, y_k)$ be a polynomial. Let X_p be the set $\{(a_1, \ldots, a_k) \in \mathbb{N}^k \mid \text{there are } b_1, \ldots, b_n \in \mathbb{N}^n \text{ such that } p(a_1, \ldots, a_k, b_1, \ldots, b_n) = 0\}$. It is now easy to see that the set X_p is recursively enumerable. However, there are polynomials p such that X_p is undecidable [**18**]; this will not be shown in this course. △

The second item of the following theorem motivates the name *recursively enumerable*: the recursive functions $f_1, \dots, f_n$ 'enumerate' the elements of the recursively enumerable set X. The third item, on the other hand, motivates the name *semi-decidable*.

THEOREM 6.3.5. *For $X \subseteq \mathbb{N}^n$ the following are equivalent.*

(1) X is recursively enumerable.
(2) $X = \emptyset$ or $X = \{(f_1(x), \dots, f_n(x)) \mid x \in \mathbb{N}\}$ where $f_1, \dots, f_n \in \mathscr{R}^{(1)}$.
(3) $X = \mathrm{dom}(g)$ for a partial recursive function g.
(4) There exists a primitive recursive set $Y \subseteq \mathbb{N}^{n+1}$ such that $X = \pi(Y)$.

PROOF. Lemma 6.1.11 implies that it suffices to show the statement for $n = 1$.

The implication from (4) to (1) is trivial. For the implication from (1) to (2), let $Y \subseteq \mathbb{N}^2$ be decidable such that $X = \pi(Y)$. If $X = \emptyset$ then there is nothing to be shown; otherwise, let $r \in X$. Define $f \colon \mathbb{N} \to \mathbb{N}$ by

$$f(x) := \begin{cases} \beta_1^2(x) & \text{if } (\beta_1^2(x), \beta_2^2(x)) \in Y \\ r & \text{else.} \end{cases}$$

Clearly, $f \in \mathscr{R}^{(1)}$ and the image of f equals X.

For the implication from (2) to (3), let $f \in \mathscr{R}^{(1)}$. Let $g := \mu(\lambda x, t.\, f(x) - t)$. Then g is a partial recursive function such that $\mathrm{dom}(g) = \mathrm{im}(f)$. Moreover, $X = \emptyset = \mathrm{dom}(\mu\pi_2^2)$.

The implication from (3) to (4) is at this point of the notes outside of our comfort zone. However, it will be an easy consequence of the results in the next section. The proof can be found at the end of Section 6.4.5. □

We are now in an awkward situation:

- We did not provide good arguments for Church's thesis (end of Section 6.2).
- Exercise 72 was too difficult.
- In the proof of Theorem 6.3.5 there is an important gap.

All of this illustrates that we don't really understand the class of recursive functions yet. The following section will improve the situation a lot.

Exercises.

(73) Show that every decidable set is recursively enumerable.

6.4. Turing Computable Functions

Turing machines were invented by Alan Turing in 1936 [**28**]. They are horrible to program. So the reader might wonder: why do we introduce Turing machines? Aren't there machine models that are more convenient for programming, after more than 80 years have passed since their discovery? My answer to the reader is: *yes*, there are such machine models, but for the purposes of this course they will probably lead to more work! The reason is that more advanced machine models require more work to simulate computation in these models. Recall from the beginning of this section: we need a machine model such that there is a machine in that model that can simulate any other machine in the same model. So we would like to keep the machine model simple.

On the other hand, there are many ways to simplify my definition of Turing machines even further (we will comment on a few simplifications later), and there are also completely different simplistic models of computation (see, e.g., Exercise 78). But then programming in such restricted models of computation is even more painful than it is with my definition of Turing machines.

To summarise: Turing machines strike a good balance between the following two opposite requirements that a theoretician has for a good machine model:

- the model should be simple, so that it can be simulated easily;
- the model should be powerful, so that it is easy to simulate other machines.

6.4.1. Turing machines. Our presentation is based on [**10**], but deviates in several details. A *Turing machine* is a tuple $M = (n, \ell, Q, s, t, \Sigma, \delta)$ consisting of

- an integer $n \geq 0$ (the number of *input tapes* of the machine),
- an integer $\ell \geq 0$ (the number of *work tapes* of the machine),
- a finite set Q (the *states* of the machine),
- $s, t \in Q$ (where s is called the *start state* and t is called the *terminal state*),
- the 3-element set $\Sigma := \{\$, |, \square\}$ (called the *alphabet*; \$ is the so-called *tape start symbol* and $\square$ is the so-called *blank symbol*) and
- a *transition function*

$$\delta \colon Q \times \Sigma^{1+n+\ell} \to Q \times \Sigma^{1+n+\ell} \times \{-1, 0, 1\}$$

such that

(1) $\delta(t, \bar{a}) = (t, \bar{a}, 0)$ and
(2) for every $q \in Q$ there exists $m \in \{0, 1\}$ and $q' \in Q$ such that

$$\delta(q, \$, \dots, \$) = (q', \$, \dots, \$, m).$$

Remark 6.4.1. The **idea** is that the machine has $1 + n + \ell$ tapes: an output tape, n input tapes, and ℓ work tapes. It has a read-write head which is at a position p above the tapes, can look at the $1 + n + \ell$ symbols at this position and modify these symbols; then it changes the position by $+1$, by -1, or leaves it unchanged; moreover, it can change to another state. How this is done precisely is specified by δ. The extra conditions on δ express that when the machine enters the terminal state, it stays there forever, and that when the machine reads the start symbol, it cannot move further to the left and must not overwrite the start symbol.

To formally define how M operates, we need to introduce *words*. The set of finite sequences of elements of Σ is denoted by Σ^*; the elements of Σ^* are also called *words over the alphabet* Σ. The *length* of a word w is the length of the sequence and denoted by $|w|$. There is a unique word of length 0 (also called the *empty word*), which is denoted by ϵ. If $w = (w_1, \dots, w_k)$ and $i \in \mathbb{N}$ with $i \geq 1$ then

$$w[i] := \begin{cases} w_i & \text{if } i \in \{1, \dots, k\} \\ \square & \text{otherwise.} \end{cases}$$

If $a \in \Sigma$, then $w[i \mapsto a]$ is the word of length $k := \max(i, |w|)$ defined as follows: for $j \leq k$, define

$$w[i \mapsto a][j] := \begin{cases} w[j] & \text{if } i \neq j \text{ and } j \in \{1, \dots, |w|\} \\ a & \text{if } i = j \\ \square & \text{otherwise.} \end{cases}$$

For the following definitions, we fix a Turing machine M.

Definition 6.4.2 (Configuration). A *configuration of* M is a tuple

$$(q, w_0, \dots, w_{n+\ell}, p) \in Q \times (\Sigma^*)^{1+n+\ell} \times \{1, 2, \dots\}.$$

Definition 6.4.3 (Successor configuration). Let $C = (q, w_0, \dots, w_{n+\ell}, p)$ be a configuration of M and let $(q', a_0, \dots, a_{n+\ell}, m) := \delta(q, w_0[p], \dots, w_{n+\ell}[p])$. Note that by condition (2) on δ, we must have $p' := p + m \geq 1$. The successor configuration of C equals $(q', w_0[p \mapsto a_0], \dots, w_{n+\ell}[p \mapsto a_{n+\ell}], p')$.

If w is the word that starts with \$ followed by the word $||\cdots|$ of length exactly $x \in \mathbb{N}$, possibly followed by arbitrarily many blank symbols, then we say that w *represents* x. Note that we represent numbers *'in unary'*; this suffices for our purposes because we are not concerned with computational complexity or practical feasibility at this point.

DEFINITION 6.4.4 (Computation). A *computation of* M is a sequence $C_0, \ldots, C_r$ of configurations of M such that C_i is the successor configuration of C_{i-1} for every $i \in \{1, \ldots, r\}$. The machine M *computes* a partial operation f of arity n over $\mathbb{N}$ if for every $x_1, \ldots, x_n \in \mathbb{N}$ we have that $(x_1, \ldots, x_n) \in \mathrm{dom}(f)$ if and only if there exists a computation $C_0, \ldots, C_r$ of M such that

$$C_0 = (s, \epsilon, w_1, \ldots, w_n, \underbrace{\epsilon, \ldots, \epsilon}_{\ell \text{ times}}, 0)$$

where w_i, for $i \in \{1, \ldots, n\}$, is the word that starts with \$ and is follows by the word $||\cdots|$ of length exactly x_i (so w_i represents x_i; the configuration C_0 is called the *start configuration of* M *on input* $\bar{x}$) and

$$C_r = (t, u_0, u_1, \ldots, u_n, v_1, \ldots, v_\ell, p)$$

for some $u_1, \ldots, u_n, v_1, \ldots, v_\ell \in \Sigma^*$ and $p \in \mathbb{N}$. Moreover, if $(x_1, \ldots, x_n) \in \mathrm{dom}(f)$ then u_0 represents $f(x_1, \ldots, x_n)$.

If M computes a partial operation f of arity n over $\mathbb{N}$ and $(x_1, \ldots, x_n) \in \mathrm{dom}(f)$, then we say that M *halts* on $x_1, \ldots, x_n$.

LEMMA 6.4.5. *The constant function* c_0^0*, the projections* π_i^n*, and the successor function* s *can be computed by a Turing machine.*

PROOF. The function c_0^0 can be computed by a Turing machine with a single tape, states $Q := \{s, t\}$, and a transition function δ such that $\delta(s, \$) = (t, \$, 0)$.

The function π_i^n can be computed by a Turing machine with $n + 1$ tapes, states $Q := \{s, t\}$, and a transition function δ such that

$$\delta(s, \$, \ldots, \$) = (s, \$, \ldots, \$, 1)$$

$$\delta(s, \square, a_1, \ldots, a_n) = \begin{cases} (s, |, a_1, \ldots, a_n, 1) & \text{if } a_i = | \\ (t, \square, a_1, \ldots, a_n, 0) & \text{otherwise.} \end{cases}$$

The successor function on $\mathbb{N}$ can be computed by a 2-tape Turing machine with states $Q := \{s, t\}$ and a transition function δ such that

$$\begin{aligned} \delta(s, \$, \$) &= (s, \$, \$, 1) \\ \delta(s, \square, |) &= (s, |, |, 1) && \text{(copy all |'s)} \\ \delta(s, \square, \square) &= (t, |, \square, 0) && \text{(add another |).} \end{aligned}$$

□

REMARK 6.4.6. There are many variations of the definition of a Turing machine that lead to the same set of partial functions. For example, we may restrict the machines to a single tape, we may restrict Σ to $\{|, \square\}$, or we may restrict the tape head to only move left or right, not staying in place. These restrictions also do not affect the length of the computations up to some polynomial factor. Only if we are interested in low time complexities, the differences become important: for example, the problem of deciding whether an input string is a palindrome (such as *1001*) can be solved in linear time on a two-tape Turing machine, but requires quadratic time on a one-tape Turing machine.

Exercises.

(74) Show that the addition operation $+\colon \mathbb{N}^2 \to \mathbb{N}$ can be computed by a Turing machine.

6.4.2. Partial recursive functions are Turing computable. In this section we show that every partial recursive function is Turing computable (Theorem 6.4.10).

LEMMA 6.4.7. *The set of Turing computable partial operations over* $\mathbb{N}$ *is closed under composition.*

PROOF. Let $M_1, \ldots, M_q$, for $q \geq 1$, be Turing machines that compute the partial operations $f_1, \ldots, f_q$ of arity p over $\mathbb{N}$, and let M' be a Turing machine that computes a partial operation g of arity q over $\mathbb{N}$. We have to show that the operation $g(f_1, \ldots, f_q)$ is Turing computatable. Rename the states of $M', M_1, \ldots, M_q$ so that these sets are pairwise disjoint. Let Q be the union over all those state sets; we will define a Turing machine M with state set Q. We may assume that $s \in Q$ equals the start state of M_1 and that $t \in Q$ equals the terminal state of M'. If M_i has n_i tapes, for $i \in \{1, \ldots, q\}$, and M' has n' tapes, then M has

$$p + (n' - q) + \sum_{i=1}^{q} (n_i - p) \tag{8}$$

tapes. The idea is that M first executes the computation of M_1 using the states from M_1, then executes the computation of M_2 using the states from M_2, and so on, until M_q, all on separate work tapes (this accounts for the summand $\sum_{i=1}^{q}(n_i - p)$ in (8)). These machines write their output on the input tapes of M' (this accounts for the summand $n' - q$ in (8)). We may have to modify the machines so that after they have finished their computation they return to position 0. Finally, we execute the computation of M' on the output of the machines $M_1, \ldots, M_p$. The resulting output will be the output of M. We leave the laborious details of the definition of the transition function for M to the reader. □

LEMMA 6.4.8. *The set of Turing computable partial functions is closed under recursion.*

PROOF. Let M be a Turing machine with $1 + n + \ell$ tapes and a set of states Q computing a partial operation g of arity n over $\mathbb{N}$. Let M' be a Turing machine with $3 + n + \ell'$ tapes and a set of states Q' computing a partial operation h of arity $n + 2$ over $\mathbb{N}$. We have to construct a Turing machine N that computes the partial function $f := \rho(g, h)$ of arity $n + 1$ over $\mathbb{N}$. Our machine has $n + 4 + \ell + \ell'$ tapes, and its set of states is given by the disjoint union of Q and Q' and some additional states. During the entire computation, tape $n + 3$ will represent a natural number which is at most x_{n+1}; this is true initially where tape $x + 3$ represents 0. For input $\bar{x} = (x_1, \ldots, x_{n+1}) \in \mathbb{N}^{n+1}$ given on tapes $1, \ldots, n, n + 1$, the machine proceeds as follows:

(1) the machine computes $g(x_1, \ldots, x_n)$ with input tapes $1, \ldots, n$, output tape $n + 2$ and work tapes $n + 4, \ldots, n + 4 + \ell$, operating as M would do, up to renumbering the tapes.
(2) Compare x_{n+1} and the natural number y which is represented on tape $n+3$: if $x_{n+1} = y$, then go into the terminal state t.
(3) Compute $h(\bar{x}, y, f(\bar{x}, y)) = f(\bar{x}, y + 1)$ by operating as M' would do, but on the input tapes $1, \ldots, n, n + 3, n + 2$, output tape 0, and the final ℓ' tapes as auxiliary tapes.
(4) Copy the content of tape 0 onto tape $n+2$, then clear tape 0, and increment the content of tape $n+3$ by one, that is, pass from y to $y+1$. Then go back to step (2).

The final output $f(\bar{x})$ can then be found on tape $n+2$. □

It follows that every primitive recursive operation is Turing computable. Note that the program in Lemma 6.4.8 implements a *'For $x = 1$-to-n loop'* known to those who are familiar with imperative programming languages. The proof of the next lemma is simpler than the proof that we have just seen; instead of the *'For-loop'* we have to implement a *'While-loop'*.

LEMMA 6.4.9. *The set of Turing computable partial functions is closed under μ-recursion.*

PROOF. Let M be a Turing machine with $n+\ell+2$ tapes and a set of states Q computing a partial operation g of arity $n+1$ over $\mathbb{N}$. We have to construct a Turing machine N that computes the partial function μg of arity n over $\mathbb{N}$. Our machine has $n+\ell+2$ tapes, and its set of states is given by the states of Q and some additional states. For input $\bar{x} = (x_1, \ldots, x_n) \in \mathbb{N}^n$ given on tapes $1, \ldots, n$, the machine proceeds as follows:

(1) the machine computes $g(x_1, \ldots, x_n, y)$ as M would do.
(2) If output tape 0 represents 0 then go into the terminal state t.
(3) Clear the work tapes of M. Increment y to $y+1$ represented on tape $n+1$. Then go back to step (1).

The final output $\mu g(\bar{x})$ can then be found on tape $n+1$. □

Note that the Turing machine in the proof of Lemma 6.4.9 might not terminate.

THEOREM 6.4.10. *Every partial recursive function is Turing computable.*

PROOF. By Lemma 6.4.5, the operations c_1^1, s, and the projections are Turing computable. Lemma 6.4.7, Lemma 6.4.8, and Lemma 6.4.9 imply that the Turing-computable functions are closed under composition, recursion, and μ-recursion. Since the partial recursive functions are the smallest such class, the statement follows. □

The reader should by now be convinced by the (non-mathematical) fact that every function which is computable in an intuitive sense is also Turing computable. Quite surprisingly, we can establish the converse of Theorem 6.4.10 – so this will finally provide some evidence for Church's thesis.

6.4.3. Turing computable functions are partial recursive. To show that every partial operation over $\mathbb{N}$ that is Turing computable is also partial recursive, we first have to describe how to code a Turing machine as a natural number.

We identify the symbol □ with 0 and the symbol | with 1. A word $w \in \Sigma^*$ is coded as

$$\ulcorner w \urcorner := \sum_{i \in \{1,\ldots,|w|\}} w[i] 2^i.$$

Let $M = (n, \ell, Q, s, t, \Sigma, \delta)$ be a Turing machine. To define a code for M, we use for fixed $k \in \mathbb{N}$ the operation $\alpha_k \colon \mathbb{N}^k \to \mathbb{N}$ introduced in Lemma 6.1.11.

- If $U = (q, s_0, \ldots, s_{n+\ell}) \in Q \times \Sigma^{n+\ell+1}$, then
$$\ulcorner U \urcorner := \alpha_2(q, \ulcorner s_0 \ldots s_{n+\ell} \urcorner).$$
- If $V = (q, s_0, \ldots, s_{n+\ell}, m) \in Q \times \Sigma^{1+n+\ell} \times \{-1, 0, 1\}$, then
$$\ulcorner V \urcorner := \alpha_3(q, \ulcorner s_0 \ldots s_{n+\ell} \urcorner, m).$$

- The transition function of M is then coded as
$$\ulcorner\delta\urcorner := \prod_{U \in Q \times \Sigma^{1+n+\ell}} \pi(\ulcorner U\urcorner)^{\ulcorner\delta(U)\urcorner}$$
where $\pi\colon \mathbb{N} \to \mathbb{N}$ was introduced in Lemma 6.1.13.
- Finally, $\ulcorner M\urcorner := \alpha_4(n, \ell, |Q|, \ulcorner\delta\urcorner)$.

Note that if M and N are distinct Turing machines, then $\ulcorner M\urcorner \neq \ulcorner N\urcorner$. Also note that the transition function can be decoded from $\ulcorner M\urcorner$ in the following formal sense: there is a primitive recursive function that, on input $(\ulcorner M\urcorner, \ulcorner U\urcorner)$ where $U \in Q \times \Sigma^{1+n+\ell}$, computes $\ulcorner\delta(U)\urcorner$. This can be shown as in the proof of Lemma 6.1.15.

LEMMA 6.4.11. *The set*
$$\mathcal{M} := \{\ulcorner M\urcorner \mid M \text{ is a Turing machine}\}$$
is primitive recursive.

PROOF SKETCH. We use the primitive recursive operations from Lemma 6.1.11 and primitive recursive operations similar to the ones constructed in Lemma 6.1.15 to compute from a given $k \in \mathbb{N}$ a set of numbers that might correspond to a Turing machine M with $\ulcorner M\urcorner = k$. We have to check whether δ is indeed a function, etc. The full details of this proof are lengthy to work out, but entirely straightforward and therefore omitted. □

Not only Turing machines M, but also their computations can be coded. If $C = (q, w_0, \ldots, w_{n+\ell}, p) \in Q \times (\Sigma^*)^{1+n+\ell} \times \mathbb{N}$ is a configuration of M, then
$$\ulcorner C\urcorner := \alpha_3(q, \langle\ulcorner w_0\urcorner, \ldots, \ulcorner w_{n+\ell}\urcorner\rangle, p).$$
where $\langle . \rangle$ is defined in Definition 6.1.14.

LEMMA 6.4.12. *There is an operation $g_1 \in \mathscr{P}^{(2)}$ such that for all $i, c \in \mathbb{N}$:*

- *If $i \notin \mathcal{M}$, then $g_1(i, c) := 0$.*
- *Otherwise, $i = \ulcorner M\urcorner$ for some unique Turing machine M. If there is no configuration C of M such that $c = \ulcorner C\urcorner$, then $g_1(i, c) := 0$.*
- *Otherwise, $c = \ulcorner C\urcorner$ for some unique configuration C of M. If C' is the successor configuration of C, then $g_1(i, c) := \ulcorner C'\urcorner$.*

PROOF. Work. □

LEMMA 6.4.13. *There is an operation $g_2 \in \mathscr{P}^{(3)}$ such that for all $i, x, t \in \mathbb{N}$:*

- *If $i \notin \mathcal{M}$, then $g_2(i, x, t) := 0$.*
- *Otherwise, $i = \ulcorner M\urcorner$ for some unique Turing machine M with n input tapes. If x is not of the form $\langle x_1, \ldots, x_n\rangle$, then $g_2(i, x, t) := 0$.*
- *Otherwise, g_2 returns $\ulcorner C'\urcorner$ for t-th successor configuration C' of the start configuration of M on input $x_1, \ldots, x_n$.*

PROOF. Let g_5 be the operation of arity $2 + n$ that returns for given numbers $n, \ell, x_1, \ldots, x_n \in \mathbb{N}$ the (unique) code for the starting configuration of any Turing machine M with n input tapes and ℓ work tapes where the input tapes represent $x_1, \ldots, x_n$; it is easy to see that g_5 is primitive recursive. The statement now follows using the operation g_1 from Lemma 6.4.12 and recursion. □

LEMMA 6.4.14. *There exists a binary partial recursive operation g_3 over $\mathbb{N}$ such that for all $i, x \in \mathbb{N}$:*

- *If there is no $t \in \mathbb{N}$ such that $g_2(i, x, t)$ returns the code of a configuration of M in an accepting state, then $g_3(i, x)$ is undefined.*

- *Otherwise, $g_3(i,x)$ equals the smallest $t \in \mathbb{N}$ such that $g_2(i,x,t)$ returns the code of a configuration of M in an accepting state.*

PROOF. Clearly the set of Gödel numbers of configurations with an accepting state is decidable; the statement follows using μ-recursion and the primitive recursive function g_2 from Lemma 6.4.13. □

THEOREM 6.4.15. *There exists a binary partial recursive operation g_4 over $\mathbb{N}$ such that for all $i, x \in \mathbb{N}$:*

- *If $i \notin \mathcal{M}$, then $g_4(i,x)$ is undefined.*
- *Otherwise, $i = \ulcorner M \urcorner$ for some unique Turing machine M with n input tapes, computing a partial operation f of arity n over $\mathbb{N}$. If x is not of the form $\langle x_1, \ldots, x_n \rangle$, then $g_4(i,x)$ is undefined;*
- *Otherwise, $g_4(i,x) := f(x_1, \ldots, x_n)$.*

PROOF. It is straightforward to implement the case distinctions in the definition of the operation g_4, using Lemma 6.4.11 and Lemma 6.1.15. Suppose now that $i = \ulcorner M \urcorner$ for some unique Turing machine with n input tapes that computes a partial operation f of arity n over $\mathbb{N}$ and that $x = \langle x_1, \ldots, x_n \rangle$ for some $x_1, \ldots, x_n \in \mathbb{N}$.

Let g_2 be the operation from Lemma 6.4.13 and let g_3 be the operation from Lemma 6.4.14. Then

$$f(x_1, \ldots, x_n) = g_4(i,x) := g_2(i, x, g_3(i,x)).$$

Moreover, g_4 is partial recursive since g_2 is primitive recursive and g_3 is partial recursive. □

COROLLARY 6.4.16. *Every partial Turing computable function is recursive.*

PROOF. Let f be a partial operation of arity n over $\mathbb{N}$ that is computed by a Turing machine M. Then for all $x_1, \ldots, x_n \in \mathbb{N}$ we have

$$f(\bar{x}) = g_4(\ulcorner M \urcorner, \langle x_1, \ldots, x_n \rangle)$$

where g_4 is the partial recursive function from Theorem 6.4.15. □

COROLLARY 6.4.17. *A set $X \subseteq \mathbb{N}^n$ is decidable if and only if there exists a Turing machine that computes 1_X.*

PROOF. An immediate consequence of Corollary 6.4.16 and Theorem 6.4.10. □

Exercises.

(75) Show that the Ackermann function is recursive.

6.4.4. Universal machines and universal functions. By combining the results of the previous two sections, we obtain a *universal machine*, i.e., a Turing machine U that for given $\ulcorner M \urcorner$ for some Turing machine M, simulates the computation of M on some given input $\bar{x}$, and returns the number that M would output on $\bar{x}$.

COROLLARY 6.4.18 (A universal machine). *There exists a 3-tape Turing machine U that computes the following binary partial operation u over $\mathbb{N}$. For $i, x \in \mathbb{N}$:*

- *If $i \notin \mathcal{M}$ then $u(i,x)$ is undefined.*
- *Otherwise, $i = \ulcorner M \urcorner$ for some unique Turing machine M that computes a partial operation f of arity n over $\mathbb{N}$. If x is not of the form $\langle x_1, \ldots, x_n \rangle$, then $u(i,x)$ is undefined.*
- *Otherwise, $u(i,x) := f(x_1, \ldots, x_n)$.*

PROOF. Let g_4 be the operation from Theorem 6.4.15. Theorem 6.4.10 implies that there exists a Turing machine U that computes g_4. Then the statement follows from Theorem 6.4.15. □

Similarly, we obtain a *universal partial recursive function.*

THEOREM 6.4.19. *Let $n \in \mathbb{N}$. Then there exists a partial recursive function u^n of arity $n+1$ over $\mathbb{N}$ such that for every partial recursive function f of arity n there exists $i \in \mathbb{N}$ such that $f = u^n_i := \lambda \bar{x}.\, u^n(i, \bar{x})$.*

PROOF. Let g_4 be the partial recursive operation from Theorem 6.4.15, and consider $u^n(i, x_1, \ldots, x_n) := g_4(i, \langle x_1, \ldots, x_n \rangle)$. If f is a partial recursive function, then there exists a Turing machine M that computes f by Theorem 6.4.10. Then we have $f(\bar{x}) = g_4(\ulcorner M \urcorner, \langle x_1, \ldots, x_n \rangle)$. Since $\langle . \rangle$ is primitive recursive, this implies the statement. □

The proofs of Lemma 6.4.13 and Theorem 6.4.15 show as a by-product that if we have the total μ-operator, we don't need recursion in the definition of total recursive functions.

COROLLARY 6.4.20. *The set of total recursive functions is the smallest subset of $\mathscr{O}$ which contains the primitive recursive operations and which is closed under composition and the total μ-operator.*

PROOF. Clearly, primitive recursive functions are total and recursive, and this set is closed under composition and the total μ-operator. Conversely, if f is total recursive, then by Theorem 6.4.10, there exists a Turing machine M that computes f. Since f is total, the operation defined by $x \mapsto g_3(\ulcorner M \urcorner, x)$ is total, and can be defined from the primitive recursive operation g_2 from Lemma 6.4.13 by the total μ-operator. Hence, for the operation g_4 from Theorem 6.4.15 we have that $x \mapsto g_4(\ulcorner M \urcorner, x)$ can be constructed from the primitive recursive operations using composition and the total μ-operator. Note that this operation equals f. □

We can now first finish our proof of a statement about recursively enumerable sets that we have already announced in Section 6.3.

PROOF OF THEOREM 6.3.5, (3) ⇒ (4). Let g be a partial recursive function of arity 1 over $\mathbb{N}$. By Theorem 6.4.19 there exists $i \in \mathbb{N}$ with $g = u^1_i$, that is,

$$X = \mathrm{dom}(g) = \{y \in \mathbb{N} \mid \exists t.\, g_2(i, \langle y \rangle, t) \neq 0\}$$

where g_2 is the primitive recursive operation from Lemma 6.4.13, which proves that $X = \pi(Y)$ for some primitive recursive set Y. □

REMARK 6.4.21. Note that the equivalence of (1) and (4) in Theorem 6.3.5 and the proof of (1) ⇒ (2) show that a subset of $\mathbb{N}^n$ is recursively enumerable if and only if $X = \emptyset$ or $X = \big\{(f_1(x), \ldots, f_n(x)) \mid x \in \mathbb{N}\big\}$ where $f_1, \ldots, f_n$ may even be chosen to be *primitive* recursive.

COROLLARY 6.4.22. *The set $S := \mathrm{dom}(u^n) \subseteq \mathbb{N}^{n+1}$ is* universal recursively enumerable *in the sense that S is recursively enumerable and every recursively enumerable set $X \subseteq \mathbb{N}^n$ is of the form $\{\bar{x} \mid (i, \bar{x}) \in S\}$ for some $i \in \mathbb{N}$.*

6.4.5. The halting problem. Note that there are sets $X \subseteq \mathbb{N}$ that are not decidable: the reason is that there are uncountably many subsets of $\mathbb{N}$, but only countably many Turing machines. So by Cantor's theorem (Theorem 4.3.2) there exist subsets of $\mathbb{N}$ that cannot be computed by Turing machines, and hence are not decidable by Corollary 6.4.17. In this section we present a *concrete* subset of $\mathbb{N}$ that is not recursive; in fact, the subset we present will even be recursively enumerable. The proof is based on a *diagonal argument* like the proof of Theorem 4.3.2.

THEOREM 6.4.23. *Let* $D := \operatorname{dom}\big(\lambda x.\, u^1(x,x)\big)$. *Then* $C := \mathbb{N} \setminus D$ *is not recursively enumerable.*

PROOF. Suppose for contradiction that C is recursively enumerable; then $C = \operatorname{dom}(u_i^1)$ for some $i \in \mathbb{N}$ by Corollary 6.4.22. Then $i \in C$ if and only if $u^1(i,i)$ is defined. On the other hand, $i \in D$ if and only if $u^1(i,i)$ is defined, a contradiction. □

REMARK 6.4.24. Note that D is recursively enumerable, but not recursive, because otherwise $C := \mathbb{N} \setminus D$ would be recursive, and hence also recursively enumerable, contrary to Theorem 6.4.23.

We prove the following theorem by reducing to a situation where we can apply a similar idea as in the proof of Theorem 6.4.23 (but presented differently).

THEOREM 6.4.25 (Undecidability of the halting problem). *The set*

$$H_0 := \big\{ i \in \mathcal{M} \mid i = \ulcorner M \urcorner \text{ and } M \text{ has no input tapes and halts} \big\}$$

is not recursive.

PROOF SKETCH. It suffices to show that the set

$$H := \big\{ (i,x) \in \mathbb{N}^2 \mid i = \ulcorner M \urcorner \text{ and } M \text{ has 1 input tape and } M \text{ halts on } x \big\}$$

is not recursive (i.e., *undecidable*). The reason is that if M is a Turing machine with one input tape and $x \in \mathbb{N}$, then it is straightforward to compute from $\ulcorner M \urcorner$ and x the number $\ulcorner N \urcorner$ for some Turing machine N such that M halts if and only if N halts on input x.

Clearly, if H is decidable, then the *special halting problem*

$$H_s := \big\{ \ulcorner M \urcorner \mid M \text{ has one input tape and } M \text{ halts on } \ulcorner M \urcorner \big\}$$

is decidable as well. So it suffices to show that H_s is undecidable. Suppose for contradiction that H_s is decidable, i.e., 1_{H_s} is recursive. By Theorem 6.4.10 there is a Turing machine M that computes 1_{H_s}. We construct a new Turing machine M' as follows:

(1) Execute M on input w.
(2) If M reaches a terminal state with 0 on the output tape, stop.
(3) Otherwise: loop forever.

How does M' behave on input $\ulcorner M' \urcorner$?

$$\begin{aligned} M' \text{ halts on } \ulcorner M' \urcorner \quad &\text{iff } M \text{ computes 0 on input } \ulcorner M' \urcorner \\ &\text{iff } \ulcorner M' \urcorner \notin H_s \\ &\text{iff } M' \text{ executed on } \ulcorner M' \urcorner \text{ does not halt.} \end{aligned}$$

A contradiction! □

THEOREM 6.4.26 (Rice's theorem). *Let* $\mathcal{X}$ *be a set of unary partial recursive operations. Suppose that* $\mathcal{X}$ *is neither empty nor equal to the set of all unary partial recursive functions over* $\mathbb{N}$. *Then*

$$I := \{ \ulcorner M \urcorner \mid M \text{ computes a partial funtion in } \mathcal{X} \}$$

is undecidable.

PROOF SKETCH. Since I is decidable if and only if $\mathbb{N} \setminus I$ is decidable, we may assume that the function with the empty domain is not in $\mathcal{X}$, because otherwise we may pass to the complement. Choose $\ulcorner L \urcorner \in I$. Assume for contradiction that there is a Turing machine N that decides I. Then we create a new machine N' that decides the halting problem H_0, which will be a contradiction to Theorem 6.4.25.

On input $\ulcorner M \urcorner$ for some Turing machine M with no input tapes,

(1) N' computes $\ulcorner M' \urcorner$ where M' is the following Turing machine:
   - on input x, the machine M' first simulates the machine M.
   - Then M' simulates L on input x and returns the output of L.

(2) Then N' simulates N on $\ulcorner M' \urcorner$.

(3) If N accepts, then N' returns 1, otherwise N' returns 0.

We verify that N' indeed decides the halting problem:

$$\begin{aligned} N' \text{ returns 1 on input } \ulcorner M \urcorner \quad & \text{iff } N \text{ accepts } \ulcorner M' \urcorner \\ & \text{iff } \ulcorner M' \urcorner \in I \\ & \text{iff } M \text{ halts.} \end{aligned}$$

The last equivalence holds because if M halts, then M' computes the same function as L, and since $\ulcorner L \urcorner \in I$ we have $\ulcorner M' \urcorner \in I$. Conversely, if M does not halt, then M' computes the function with the empty domain. Since this function is by assumption not in $\mathcal{X}$, we conclude that $\ulcorner M' \urcorner \notin I$. □

Exercises.

(76) Show that the set of primitive recursive sets is not closed under projections.

(77) Find a subset S of $\mathbb{N}$ such that neither S nor its complement $\mathbb{N} \setminus S$ is recursively enumerable.

(78) While Turing machines can be viewed as the origin of imperative programming languages, the following formalism, which goes back to Alonzo Church and Stephen Kleene in the 1930s, can be viewed as the origin of all functional programming languages. A *λ-term* is defined inductively as follows:
   - variables are λ-terms;
   - if M is a λ-term and x is a variable, then so is $\lambda x.\, M$ *(Abstraction)*;
   - if M and N are λ-terms, then so is (MN) *(Application)*.

   If x is a variable and M, N are λ-terms, then the λ-term $M[x := N]$ obtained from M by substituting x with N is defined inductively as follows:
   - if M is of the form x, then $M[x \mapsto N] := N$;
   - if M is of the form $\lambda y.\, M'$ and x and y are distinct variable symbols, then
   $$M[x \mapsto N] := \lambda y (M[x \mapsto N]);$$
   - if M is of the form $\lambda x.\, M'$ then
   $$M[x \mapsto N] := (\lambda x.\, M')$$
   (all occurrences of x in M are bound, 'no substitution');
   - if M is of the form $(M_1 M_2)$ then
   $$M[x \mapsto N] := (M_1[x \mapsto N] M_2[x \mapsto N]).$$

   A *β-reduction step* replaces a λ-term of the form $((\lambda x.M)N)$ by the λ-term $M[x \mapsto N]$. We then also write $((\lambda x.M)N) \xrightarrow{\beta} M[x \mapsto N]$. For example,
   $$(\lambda x.\, x)(\lambda x.\, x) \quad \xrightarrow{\beta} \quad x[x \mapsto (\lambda x.\, x)] = \lambda x.\, x$$

Note that for example in the term $\lambda x.\, x$ no further β-reduction steps can be applied. Show that there are λ-terms such that repeatedly performing β-reduction does not terminate.

(79) Show that the set of pairs (s, t) of λ-terms, coded as a set of natural numbers, such that t can be reached from s by repeated application of β-reduction steps is recursively enumerable.

(80) (*) Show that the set from the previous exercise it not recursive.

(81) (*) Let τ be a signature that only contains unary relation symbols.
- Show that τ-sentences have the *finite model property*: if a τ-sentence ϕ has a model, then it also has a finite model.
- Derive that there exists an algorithm that tests whether a given τ-sentence is satisfiable.

(82) Show the existence of a *quine*, which is a Turing machine M with no input tapes which writes $\ulcorner M \urcorner$ on its output tape.

CHAPTER 7

The Incompleteness Theorems

In this chapter we prove Gödel's first incompleteness theorem: every consistent and recursive theory that contains the axioms of *weak Peano arithmetic* is necessarily incomplete (Theorem 7.5.1). The same holds for ZF instead of weak Peano arithmetic.

We then also sketch a proof of Gödel's second incompleteness theorem: every consistent and recursive theory that contains *Peano arithmetic* cannot prove its own consistency (Theorem 7.7.11). It follows that if ZF is consistent, then there is no formal proof in ZF proving the consistency of ZF.

7.1. Coding Formulas and Proofs

Let $\tau = \{z_1, \dots, z_n\}$ be a finite signature; for each $i \in \{1, \dots, n\}$ the symbol z_i is either a relation or a function symbol. We assign to each τ-formula ϕ a *Gödel number* $\ulcorner \phi \urcorner \in \mathbb{N}$ as follows. The set of *logical symbols* is the union of τ, a countably infinite set of variables $v_1, v_2, \dots$, and the set of symbols containing $=$, $\wedge$, $\neg$, $\exists$, (, and). We start by assigning Gödel numbers to the logical symbols as follows ($\langle \cdot \rangle$ has been defined in Lemma 6.1.15).

z	$=$	$\wedge$	$\neg$	$\exists$	(	)	z_i	v_i
$\ulcorner z \urcorner$	$\langle 0,0\rangle$	$\langle 0,1\rangle$	$\langle 0,2\rangle$	$\langle 0,3\rangle$	$\langle 0,4\rangle$	$\langle 0,5\rangle$	$\langle 0, i+5\rangle$	$\langle 1, i\rangle$

The Gödel number $\ulcorner w \urcorner$ of a word $w = w_1 \dots w_n \in \Sigma^*$ is defined as

$$\langle \ulcorner w_1 \urcorner, \dots, \ulcorner w_n \urcorner \rangle.$$

The proofs of the facts in Lemma 7.1.1, Lemma 7.1.2, and Theorem 7.1.3 can be seen as programming exercises.

LEMMA 7.1.1. *The following sets are primitive recursive.*

- *Term*$(\tau) := \{\ulcorner t \urcorner \mid t \text{ is } \tau\text{-term}\}$.
- *Formula*$(\tau) := \{\ulcorner \phi \urcorner \mid \phi \text{ is } \tau\text{-formula}\}$.
- *Sentence*$(\tau) := \{\ulcorner \phi \urcorner \mid \phi \text{ is } \tau\text{-sentence}\}$.

LEMMA 7.1.2. *There exists a primitive recursive function* $\mathrm{tsubst} \in \mathscr{P}^3$ *such that for every* $n \in \mathbb{N}$ *and all* τ*-terms* s, t

$$\mathrm{tsubst}(n, \ulcorner s \urcorner, \ulcorner t \urcorner) = \ulcorner t[v_n \mapsto s] \urcorner.$$

There exists a primitive recursive function $\mathrm{fsubst} \in \mathscr{P}^3$ *such that for every* $n \in \mathbb{N}$*, every* τ*-term* s*, and every* τ*-formula* ϕ

$$\mathrm{fsubst}(n, \ulcorner s \urcorner, \ulcorner \phi \urcorner) = \ulcorner \phi[v_n \mapsto s] \urcorner.$$

THEOREM 7.1.3. *The set* $\{\ulcorner \phi \urcorner \mid \phi \text{ is a logical axiom for } \tau\}$ *is primitive recursive.*

For a theory T, define

$$\#T := \{\ulcorner \phi \urcorner \mid \phi \in T\}$$
$$\mathrm{Prf}(T) := \{\langle \ulcorner \phi_1 \urcorner, \dots, \ulcorner \phi_n \urcorner \rangle \mid (\phi_1, \dots, \phi_n) \text{ is a formal proof in } T\}.$$

LEMMA 7.1.4. *Let T be a theory. If $\#T$ is recursive, then* $\mathrm{Prf}(T)$ *is recursive, and if $\#T$ is primitive recursive, then* $\mathrm{Prf}(T)$ *is primitive recursive. In particular,* $\mathrm{Prf}(\emptyset)$ *is primitive recursive.*

PROPOSITION 7.1.5. *The set $\{\ulcorner\phi\urcorner \mid \vdash \phi\}$ is recursively enumerable.*

PROOF. We have $\vdash \phi$ if and only if there exist $n \in \mathbb{N}$ and first-order sentences $\phi_1, \ldots, \phi_n$ such that $(\phi_1, \ldots, \phi_n) \in \mathrm{Prf}(\emptyset)$ and $\phi_n = \phi$. □

7.2. Decidable Theories

Let T be a τ-theory. The *deductive closure of* T is the set

$$\mathrm{Thm}(T) := \{\phi \text{ a } \tau\text{-sentence} \mid T \vdash \phi\}.$$

A theory is called *finitely axiomatisable* if there exists a finite theory T' such that $\mathrm{Thm}(T) = \mathrm{Thm}(T')$.

EXAMPLE 26. The theory $\mathrm{Th}(\mathbb{Q}; <)$ is finitely axiomatisable. Clearly, $(\mathbb{Q}; <)$ is an unbounded and dense linear order. Note that the fact that $<$ is an unbounded and dense linear order can be expressed by a first-order sentence ψ. Conversely, if ϕ is a first-order sentence that holds in $(\mathbb{Q}; <)$, then it must hold in all countable dense linear orders: this follows from the well-known and easy fact that all countable dense linear orders without endpoints are isomorphic (this was observed already by Cantor **[3]**). In fact, it can also be shown that ϕ must hold in *all* countable dense linear orders (we refer to a model theory course for more details) and hence $\mathrm{Th}(\mathbb{Q}; <) = \mathrm{Thm}(\{\psi\})$. △

We say that T is *recursively* (or *effectively*) *axiomatisable* if there exists a τ-theory T' such that $\mathrm{Thm}(T) = \mathrm{Thm}(T')$ and $\#T'$ is recursive.

EXAMPLE 27. Every finitely axiomatisable theory is effectively axiomatisable. △

PROPOSITION 7.2.1. *A theory T is effectively axiomatisable if and only if* $\#\,\mathrm{Thm}(T)$ *is recursively enumerable.*

PROOF. First suppose that there exists a theory T' such that $\mathrm{Thm}(T) = \mathrm{Thm}(T')$ and $\#T'$ is recursive. We have $\phi \in \mathrm{Thm}(T)$ if and only if there exists a formal proof of ϕ in T'. The statement follows because $\mathrm{Prf}(T')$ is recursive by Lemma 7.1.4.

Conversely, suppose that $\#\,\mathrm{Thm}(T)$ is recursively enumerable. By Theorem 6.3.5 (1) ⇒ (2) there exists a recursive function $f \in \mathscr{R}^{(1)}$ such that $\#\,\mathrm{Thm}(T) = \{f(i) \mid i \in \mathbb{N}\}$. Let ϕ_i be such that $\ulcorner\phi_i\urcorner = f(i)$. The function $g\colon \mathbb{N} \to \mathbb{N}$ given by

$$g(n) := \ulcorner \textstyle\bigwedge_{i\in\{0,\ldots,n\}} \phi_i \urcorner$$

is recursive and we have that $g(n) \geq n$ for every $n \in \mathbb{N}$. The image of g is therefore a recursive set. Since

$$\mathrm{Thm}(\{\bigwedge_{i\in\{0,\ldots,n\}} \phi_i \mid n \in \mathbb{N}\}) = \mathrm{Thm}(T)$$

this proves that T is effectively axiomatisable. □

DEFINITION 7.2.2. A theory T is called

- *recursive* if $\#T$ is recursive.
- *decidable* if $\#\,\mathrm{Thm}(T)$ is recursive.

THEOREM 7.2.3. *Let T be a complete τ-theory which is effectively axiomatisable. Then T is decidable.*

PROOF. By Proposition 7.2.1 we know that $\#\operatorname{Thm}(T)$ is recursively enumerable. Its complement $\mathbb{N} \setminus \#\operatorname{Thm}(T)$ equals

$$\{\ulcorner\phi\urcorner \mid \neg\phi \in \operatorname{Thm}(T)\} \cup \{a \mid a \notin \operatorname{Sentence}(\tau)\}.$$

The first set in this union is recursively enumerable since $\ulcorner\phi\urcorner \mapsto \ulcorner\neg\phi\urcorner$ is given by a (primitive) recursive function. The second set is primitive recursive by Lemma 7.1.1. Hence, the union is recursively enumerable as well by Proposition 6.3.4. Theorem 6.3.2 therefore implies that $\#\operatorname{Thm}(T)$ is recursive. □

EXAMPLE 28. The theory $T := \operatorname{Th}(\mathbb{Q}; <)$ is finitely axiomatisable (Example 26), and hence it is effectively axiomatisable (Example 27). Of course, T is complete and thus Theorem 7.2.3 implies that T is decidable. △

Exercises.

(83) (Exercise 5.2.6 in Hils and Loeser [**10**]) Let T be a τ-theory. If T is decidable and S is a finite set of first-order τ-sentences, then $T \cup S$ is decidable.
(84) Is $\operatorname{Th}(\mathbb{N}; \neq)$ finitely axiomatisable?

7.3. (Weak) Peano Arithmetic

In this section we work with the signature $\tau_{\text{Arithm}} := \{0, s, +, \cdot, <\}$ of arithmetic. For every $n \in \mathbb{N}$ we define the term $\underline{n}$ by induction, setting $\underline{0} := 0$ and $\underline{n+1} := s(\underline{n})$.

DEFINITION 7.3.1. The set PA_0 of *weak Peano axioms*, also known as *Robinson's theory Q*, is the first-order theory consisting of the following eight sentences:

$A1$	$\forall v_0.\, s(v_0) \neq 0$	(0 is not a successor)
$A2$	$\forall v_0\, \exists v_1\, (v_0 \neq 0 \Rightarrow s(v_1) = v_0)$	(every non-zero element has a predecessor)
$A3$	$\forall v_0, v_1\, (s(v_0) = s(v_1) \Rightarrow v_0 = v_1)$	(injectivity of s)
$A4$	$\forall v_0\, (v_0 + 0 = v_0)$	(0 is additively neutral)
$A5$	$\forall v_0, v_1\, (v_0 + s(v_1) = s(v_0 + v_1))$	(distributivity of $+$ over s)
$A6$	$\forall v_0\, (v_0 \cdot 0 = 0)$	(0 is multiplicative zero)
$A7$	$\forall v_0, v_1\, (v_0 \cdot s(v_1) = (v_0 \cdot v_1) + v_0)$	(distributivity of $\cdot$ over s)
$A8$	$\forall v_0, v_1\, \big(v_0 < v_1 \Leftrightarrow$ $\exists v_2(v_2 + v_0 = v_1 \wedge v_0 \neq v_1)\big)$	(definition of $<$)

Note that we could alternatively work with the signature without $<$ and the first seven axioms only, because $<$ is first-order definable from the other symbols in the signature (Axiom 8 provides such a definition).

At first sight, weak Peano arithmetic appears to be very weak: there are even models of PA_0 in which multiplication is not commutative and where there are elements x such that $s(x) = x$, as the following example illustrates.

EXAMPLE 29. Consider the structure $\underline{A}$ whose domain A is $\mathbb{N} \cup \{a\}$ where a is some new element, not contained in $\mathbb{N}$. Define

$$0^{\underline{A}} := 0$$

$$s^{\underline{A}}(x) := \begin{cases} x+1 & \text{if } x \in \mathbb{N}, \\ a & \text{if } x = a \end{cases}$$

$$x +^{\underline{A}} y := \begin{cases} x+y & \text{if } x, y \in \mathbb{N}, \\ a & \text{otherwise} \end{cases}$$

$$x \cdot^{\underline{A}} y := \begin{cases} xy & \text{if } x, y \in \mathbb{N}, \\ 0 & \text{if } y = 0, \\ a & \text{otherwise.} \end{cases}$$

This finishes our description of $\underline{A}$ because $0^{\underline{A}}$ and $+^{\underline{A}}$ uniquely determine $<^{\underline{A}}$. Note that $s^{\underline{A}}(a) = a$ and that $x \cdot^{\underline{A}} 0 = 0 \neq a = 0 \cdot^{\underline{A}} a$. △

But we will see later that even in such a weak theory we can express surprisingly strong things with first-order sentences (Theorem 7.3.8).

DEFINITION 7.3.2. The set PA of *Peano axioms* consists of PA_0 together with the set of first-order sentences of the form

$$\forall v_1, \ldots, v_n \left((\phi(0, \bar{v}) \wedge \forall v_0 \left(\phi(v_0, \bar{v}) \Rightarrow \phi(s(v_0), \bar{v}) \right)) \Rightarrow \forall v_0 . \, \phi(v_0, \bar{v}) \right)$$

for a first-order τ_{Arithm}-formula $\phi(v_0, v_1, \ldots, v_n)$.

Clearly, the structure $\underline{N}_{\mathrm{st}} := (\mathbb{N}; 0, s, +, \cdot, <)$ with the usual definition of the successor function, addition, multiplication, and the order, is a model of PA. Note that the natural numbers, the successor function, addition, multiplication, and the order were all formalised in ZF in Chapter 4; it follows that if ZF is consistent, then so is PA. If $\underline{A}$ is a model of PA, then an element $a \in A$ is called *non-standard* if $a \neq \underline{n}^{\underline{A}}$ for every $n \in \mathbb{N}$. By compactness of first-order logic, there are models of PA with non-standard elements.

LEMMA 7.3.3. *For every* $n \in \mathbb{N}$

$$\mathrm{PA}_0 \models \forall x (x < \underline{n} \Leftrightarrow \bigvee_{i=0}^{n-1} x = \underline{i}).$$

PROOF. Let $\underline{A} \models \mathrm{PA}_0$. We show the statement by induction on n. For $n = 0$, suppose for contradiction that $\underline{A} \models c < 0$ for some $c \in A$. Then by A8 there exists $a \in A$ such that $\underline{A} \models a + c = 0 \wedge c \neq 0$. Hence $c = s(d)$ for some d by A2, and

$$\begin{aligned} 0 &= a + s(d) \\ &= s(a + d) && \text{(by A5)} \end{aligned}$$

in contradiction to A1. For the induction step, we first prove that

$$\mathrm{PA}_0 \models \forall x, y \left(x < y \Leftrightarrow s(x) < s(y) \right). \tag{9}$$

This follows from A8, using that if $a, b, c \in A$ then

$$\begin{aligned} \underline{A} \models c + a = b \wedge a \neq b &\Leftrightarrow \underline{A} \models s(c + a) = s(b) \wedge s(a) \neq s(b) && \text{(by A3)} \\ &\Leftrightarrow \underline{A} \models s(c) + a = s(b) \wedge s(a) \neq s(b) && \text{(by A5).} \end{aligned}$$

Now let $a \in A$. We have to show that $a < \underline{n+1}^{\underline{A}}$ if and only if $a = \underline{i}$ for some $i < n+1$. The statement holds for $a = 0$ because A8 implies that $0 < c$ for any $c \neq 0$. Otherwise, there exists $b \in A$ such that $s(b) = a$ by A2. Then

$$\begin{aligned} a < \underline{n+1}^{\underline{A}} &\Leftrightarrow b < n^{\underline{A}} && \text{(by (9))} \\ &\Leftrightarrow b = \underline{i}^{\underline{A}} \text{ for some } i < n && \text{(by inductive assumption)} \\ &\Leftrightarrow a = \underline{i}^{\underline{A}} \text{ for some } i < n+1 && \text{(by (9)).} \end{aligned}$$

□

LEMMA 7.3.4. *Let $\underline{A}$ be a model of* PA_0. *Then $N := \{\underline{n}^{\underline{A}} \mid n \in \mathbb{N}\}$ is the domain of a substructure of $\underline{A}$ which is isomorphic to $\underline{N}_{st}$.*

PROOF. Let $i\colon \mathbb{N} \to N$ be the map $n \mapsto \underline{n}^{\underline{A}}$. One verifies easily by (naive) induction on $n \in \mathbb{N}$ that i preserves 0, s, $+$, and $\cdot$. For example, for any $m, n \in \mathbb{N}$ one has $\mathrm{PA}_0 \models \underline{m} + \underline{n} = \underline{m+n}$, using (A4) and (A5). The injectivity of i follows from Lemma 7.3.3 (Exercise 85). It follows from (A8) that i preserves $<$. The verification that i also preserves the complement of $<$ is left as an exercise. □

DEFINITION 7.3.5. Let $f \in \mathscr{O}^{(k)}$. A formula $\phi(x_0, x_1, \ldots, x_k)$ *represents* f if for all $n_1, \ldots, n_k \in \mathbb{N}$ one has

$$\mathrm{PA}_0 \models \forall y\big(\phi(y, \underline{n_1}, \ldots, \underline{n_k}) \Leftrightarrow y = \underline{f(n_1, \ldots, n_k)}\big).$$

Let $R \subseteq \mathbb{N}^k$. A formula $\phi(x_1, \ldots, x_k)$ *represents* R if for all $n_1, \ldots, n_k \in \mathbb{N}$ one has

$$\begin{aligned} &\text{if } (n_1, \ldots, n_k) \in R \text{ then } \mathrm{PA}_0 \models \phi(\underline{n_1}, \ldots, \underline{n_k}) \\ &\text{if } (n_1, \ldots, n_k) \notin R \text{ then } \mathrm{PA}_0 \models \neg\phi(\underline{n_1}, \ldots, \underline{n_k}). \end{aligned}$$

DEFINITION 7.3.6. The set of Σ_1-*formulas* is the smallest set of τ_{Arithm}-formulas that contains all quantifier-free formulas and which is stable under existential quantification $\exists x$, conjunction $\wedge$, disjunction $\vee$, bounded universal quantification of the form $\forall x(x < t \Rightarrow \phi)$ where t is a term not depending on the variable x. We also write $\forall x < t.\, \phi$ instead of $\forall x(x < t \Rightarrow \phi)$.

The set of *strict* Σ_1-*formulas* is the smallest set of τ_{Arithm}-formulas containing the formulas $0 = x$, $s(x) = y$, $x + y = z$, $x \cdot y = z$, $x = y$, $\neg(x = y)$, $x < y$, $\neg(x < y)$ and which is stable under conjunction, disjunction, existential quantification, and universal quantification of the form $\forall x < y.\, \phi$, assuming that x and y are two distinct variables.

LEMMA 7.3.7. *Any Σ_1-formula is equivalent to a strict Σ_1-formula.*

PROOF. One may eliminate complex terms by using existential quantifiers. □

For example, $s(x) + y = z$ is equivalent to

$$\exists x'(s(x) = x' \wedge x' + y = z).$$

THEOREM 7.3.8. *Every Σ_1-sentence that holds in $\underline{N}_{st}$ is a consequence of* PA_0.

PROOF. By Lemma 7.3.7, it suffices to prove the statement for strict Σ_1-sentences. We prove that for any strict Σ_1-formula $\phi(x_1, \ldots, x_n)$ and $m_1, \ldots, m_n \in \mathbb{N}$ we have $\underline{N}_{\mathrm{st}} \models \phi(m_1, \ldots, m_n) \Rightarrow \mathrm{PA}_0 \models \phi(\underline{m_1}, \ldots, \underline{m_n})$. We prove this by structural induction over the definition of formulas, starting from the case when ϕ is an atomic formula or a negated atomic formula which follows from Lemma 7.3.4.

Assume that the statement holds for $\phi(x_0, x_1, \ldots, x_n)$ and for $\psi(x_0, x_1, \ldots, x_n)$. Clearly it then also holds for $\phi \wedge \psi$ and $\phi \vee \psi$. To prove that it holds for $\exists x_0.\, \phi$, suppose that $\underline{N}_{\mathrm{st}} \models (\exists x_0.\, \phi)(m_1, \ldots, m_n)$. Then there exists $m_0 \in \mathbb{N}$ such that $\underline{N}_{\mathrm{st}} \models \phi(m_0, m_1, \ldots, m_n)$, which by the inductive assumption implies that $\mathrm{PA}_0 \models \phi(\underline{m_0}, \underline{m_1}, \ldots, \underline{m_n})$. Hence, $\mathrm{PA}_0 \models (\exists x_0.\phi)(\underline{m_1}, \ldots, \underline{m_n})$.

Finally, suppose that $\underline{N}_{\mathrm{st}} \models (\forall x_0 < x_1.\phi)(m_1, \dots, m_n)$. Then by definition of bounded universal quantification one has $\underline{N}_{\mathrm{st}} \models \phi(m_0, m_1, \dots, m_n)$ for every $m_0 < m_1$, and by inductive assumption $\mathrm{PA}_0 \models \phi(\underline{m_0}, \underline{m_1}, \dots, \underline{m_n})$ for every $m_0 < m_1$. In other words, $\mathrm{PA}_0 \models \bigvee_{i=0}^{m_1-1} \phi(\underline{i}, \underline{m_1}, \dots, \underline{m_n})$. Lemma 7.3.3 then implies that $\mathrm{PA}_0 \models \forall x_0 < \underline{m_1}.\, \phi(x_0, \underline{m_1}, \dots, \underline{m_n})$. □

The next result would be easy to prove, but there is an important technical complication.

THEOREM 7.3.9 (Representability theorem). *Every recursive function can be represented by a Σ_1-formula.*

(BEGINNING OF PROOF). We already know from Corollary 6.4.20 that every recursive function can be constructed from the primitive recursive operations by composition and the total μ-operator. It is clear that the successor function s, the constant function c_0^0, and the projections are representable. By the trick from Lemma 7.3.7 we may also represent compositions of representable functions. Suppose now that $f(\bar{y}, x)$ is representable. Let

$$A := \{(\bar{y}, x) \mid f(\bar{y}, x) = 0\}.$$

Then 1_A is recursive and representable by what we have proved above; let $\phi(z, \bar{x}, y)$ be a Σ_1-formula that represents 1_A. Then μf has the representation

$$\phi(\underline{1}, \bar{y}, x) \wedge \forall x' < x.\, \phi(\underline{0}, \bar{y}, x').$$

We are left with the task to verify that the class of representable functions is also closed under recursion (cf. Definition 6.2.4); this is the mentioned complication. However, it turns out that recursion is not needed in the definition of recursive functions! The reason is that recursion can be simulated by total μ-recursion; proving this requires some efforts, though. □

As a temporary definition, let $\mathscr{R}_0$ be the smallest clone that contains s, c_0^0, and is closed under total μ-recursion. To prove that $\mathscr{R}_0 = \mathscr{R}$ (Lemma 7.3.12) we basically follow Section 6.1, replacing recursion by total μ-recursion. Subsets of $\mathbb{N}^n$ whose indicator function is in $\mathscr{R}_0$ are called 0-recursive (temporarily, since we are about to prove that 0-recursive sets and recursive sets are the same thing).

LEMMA 7.3.10. *$\mathscr{R}_0^{(3)}$ contains $\dot{-}$ (see Lemma 6.1.4). The set of 0-recursive sets is closed under Boolean combinations, bounded quantification, and contains*

$$\{(x, y, z) \in \mathbb{N}^3 \mid x \equiv y \mod z\}.$$

Moreover, the set $\mathscr{R}_0$ is stable under definition by cases (see Lemma 6.1.8).

The final ingredient to the proof of Theorem 7.3.9 is an ingenious idea of Gödel.

LEMMA 7.3.11 (Gödel's β-function). *There exists an operation $\beta \in \mathscr{R}_0^{(3)}$ such that for any finite sequence $(c_0, \dots, c_{n-1})$ of natural numbers there are $a, b \in \mathbb{N}$ with $\beta(a, b, i) = c_i$ for $i \in \{0, \dots, n-1\}$.*

PROOF SKETCH. Define

$$\beta(a, b, i) := \min\{z \in \mathbb{N} \mid z \equiv a \mod ((i+1)b + 1)\}.$$

It is easy to write this definition using the total μ-operator using functions whose containment in $\mathscr{R}_0$ we already know. To prove that β has the required properties, let $c_0, \dots, c_{n-1}$ be given and let $b \in \mathbb{N}$ be such that b is divisible by $n!$ and $b > c_i$ for all i. Then $b+1$, $2b+1$, $3b+1$, …, $nb+1$ is a sequence of pairwise prime integers. By the Chinese Remainder Theorem there exists $a \in \mathbb{N}$ such that for all $i \in \{0, \dots, n-1\}$ it

holds that $a \equiv c_i \mod ((i+1)b+1)$. As $c_i < (i+1)b+1$ for all i, it is the smallest natural number which is congruent to a modulo $((i+1)b+1)$. □

LEMMA 7.3.12 (Elimination of recursion). $\mathscr{R} = \mathscr{R}_0$.

PROOF. By Corollary 6.4.20 it suffices to show that $\mathscr{R}_0$ is closed under recursion, because then by definition $\mathscr{R}_0$ contains the primitive recursive operations, and hence equals $\mathscr{R}$ by Corollary 6.4.20. Let $g \in \mathscr{R}_0^{(n)}$ and $h \in \mathscr{R}_0^{(n+2)}$ and let $f \in \mathscr{O}^{(n+1)}$ be the operation obtained from g and h by recursion. Consider the set

$$Z := \big\{(\bar{x}, y, a, b) \in \mathbb{N}^{n+3} \mid \beta(a,b,0) = g(\bar{x}) \text{ and for all } i \leq y \\ \beta(a,b,i+1) = h(\bar{x}, i, \beta(a,b,i))\big\}$$

whose indicator function is in $\mathscr{R}_0$ by Lemma 7.3.11. Moreover, the property of β implies that for any $(\bar{x}, y) \in \mathbb{N}^{n+1}$ there are $a, b \in \mathbb{N}$ such that $(\bar{x}, y, a, b) \in Z$. Thus, $e := \lambda \bar{x}, y. \min\{s \mid \exists a, b \leq s. (\bar{x}, y, a, b) \in Z\}$ is in $\mathscr{R}_0$. Finally, $f(\bar{x}, y)$ equals

$$\min\{z \mid \exists a, b \leq e(\bar{x}, y).\ (\bar{x}, y, a, b) \in Z \text{ and } z = \beta(a, b, y)\}$$

proving that $f \in \mathscr{R}_0$. □

This completes the proof of Theorem 7.3.9.

COROLLARY 7.3.13. *Every recursively enumerable set can be represented by a* Σ_1*-formula.*

PROOF. Let $R \subseteq \mathbb{N}^n$ be of the form $\pi(Y)$ for some recursive set Y. By Theorem 7.3.9 there exists a Σ_1-formula ϕ that represents 1_Y. Then $\exists y.\, \phi(\underline{0}, x_1, \ldots, x_n, y)$ represents R. □

Exercises.

(85) Prove that the map i in the proof of Lemma 7.3.4 is injective.
(86) Prove that the map i in the proof of Lemma 7.3.4 also preserves the complement of $<$.
(87) A relation $R \subseteq \mathbb{N}^n$ is called *arithmetical* if it is first-order definable[1] in the structure $\underline{N}_{\text{st}} = (\mathbb{N}; 0, s, +, \cdot, <)$. Show that every recursively enumerable set is arithmetical.
(88) Show that the theory $\text{Th}(\underline{N}_{\text{st}})$ is undecidable (Definition 7.2.2).
(89) Is $\text{Th}(\underline{N}_{\text{st}})$ finitely axiomatisable?
(90) Is PA a complete theory? Try to answer this question without using results that come later in the course notes.
(91) Show that
$$\text{PA} \vdash \forall x, y.\ x + y = y + x,$$
indicating carefully which axioms of Peano arithmetic are used, and where.
(92) Do the ordinal numbers with respect to $<$ form a model of WPA? Is this a properly phrased question?

7.4. The Theorems of Tarski and Church

We start this section with a proof of the fixed point theorem in logic, which states that for every first-order definable property E of the natural numbers there exists a first-order sentence ψ in the language of arithmetic which states that

"My Gödel number has property E."

[1] First-order definability was introduced in Section 3.2.4.

We prove this with a diagonalisation argument, similar to the one used in the proof of Cantor's theorem (Theorem 4.3.2), the existence of recursively enumerable sets that are not recursive (Theorem 6.4.23), and the undecidability of the halting problem (Theorem 6.4.25).

Lemma 7.1.2 implies that there exists a primitive recursive function subst such that if $\phi(x)$ is a τ_{Arithm}-formula and $n \in \mathbb{N}$, then $\mathrm{subst}(\ulcorner \phi \urcorner, n) = \ulcorner \phi(\underline{\mathrm{n}}) \urcorner$. By the representability theorem (Theorem 7.3.9), there exists a Σ_1-formula $G(z, x, y)$ representing subst. For a given formula $\phi(x)$, define

$$\begin{aligned} H_\phi(x) &:= \exists z \left(G(z, x, x) \wedge \phi(z) \right) \\ n_\phi &:= \ulcorner \neg H_\phi \urcorner \\ \Delta_\phi &:= \neg H_\phi(\underline{n_\phi}). \end{aligned}$$

Proposition 7.4.1 (The diagonal argument). *Let $\phi(x)$ be a τ_{Arithm}-formula. Then*

$$\mathrm{PA}_0 \vdash \left(\phi(\underline{\ulcorner \Delta_\phi \urcorner}) \Leftrightarrow \neg \Delta_\phi \right).$$

Proof. Note that

$$\mathrm{subst}(n_\phi, n_\phi) = \mathrm{subst}(\ulcorner \neg H_\phi \urcorner, n_\phi) = \ulcorner \Delta_\phi \urcorner. \tag{10}$$

Let $\underline{A}$ be a model of PA_0. Then

$$\begin{aligned} \underline{A} \models \phi(\underline{\ulcorner \Delta_\phi \urcorner}) &\Leftrightarrow \underline{A} \models \phi(\underline{\mathrm{subst}(n_\phi, n_\phi)}) && \text{(by (10))} \\ &\Leftrightarrow \underline{A} \models \exists z \big(G(z, \underline{n_\phi}, \underline{n_\phi}) \wedge \phi(z) \big) && \text{(by the definition of } G\text{)} \\ &\Leftrightarrow \underline{A} \models H_\phi(\underline{n_\phi}) && \text{(by the definition of } H_\phi\text{)} \\ &\Leftrightarrow \underline{A} \models \neg \Delta_\phi && \text{(by the definition of } \Delta_\phi\text{).} \quad \square \end{aligned}$$

Corollary 7.4.2 (Fixed point theorem). *For any τ_{Arithm}-formula $\phi(x)$ there exists a τ_{Arithm}-sentence ψ such that*

$$\mathrm{PA}_0 \vdash \left(\phi(\underline{\ulcorner \psi \urcorner}) \Leftrightarrow \psi \right).$$

Proof. This follows from Theorem 7.4.1 by taking $\psi := \Delta_{\neg\phi}$:

$$\mathrm{PA}_0 \vdash \left(\neg\phi(\underline{\ulcorner \Delta_{\neg\phi} \urcorner}) \Leftrightarrow \neg \Delta_{\neg\phi} \right) \quad \square$$

Theorem 7.4.3 (Tarski's theorem on the Non-definability of Truth). *Let $\underline{A} \models \mathrm{PA}_0$. Then there is no τ_{Arithm}-formula $\phi(x)$ such that for every τ_{Arithm}-sentence ψ one has $\underline{A} \models \psi$ if and only if $\underline{A} \models \phi(\underline{\ulcorner \psi \urcorner})$.*

Proof. Suppose for contradiction that $\phi(x)$ is a τ_{Arithm}-sentence as in the statement of the theorem. By Proposition 7.4.1 we have $\mathrm{PA}_0 \vdash \phi(\underline{\ulcorner \Delta_\phi \urcorner}) \Leftrightarrow \neg \Delta_\phi$. Hence, $\underline{A} \models \phi(\underline{\ulcorner \Delta_\phi \urcorner})$ if and only if $\underline{A} \models \neg \Delta_\phi$. However, by assumption $\underline{A} \models \phi(\underline{\ulcorner \Delta_\phi \urcorner})$ if and only if $\underline{A} \models \Delta_\phi$, a contradiction. $\square$

The following is much stronger than the statement from Exercise 88.

Theorem 7.4.4 (Church's theorem). *Let T be a consistent τ_{Arithm}-theory that contains PA_0. Then T is undecidable.*

Proof. Suppose for contradiction that $\#\,\mathrm{Thm}(T)$ would be recursive. Then by the representability theorem (Theorem 7.3.9) there exists a Σ_1-formula $\phi(x)$ representing $\#\,\mathrm{Thm}(T)$. Applying Proposition 7.4.1 we would get

$$\begin{aligned} T \vdash \Delta_\phi &\Leftrightarrow \#\Delta_\phi \in \#\,\mathrm{Thm}(T) && \text{(by definition of \# and Thm)} \\ &\Leftrightarrow \mathrm{PA}_0 \vdash \phi(\underline{\ulcorner \Delta_\phi \urcorner}) && \text{(by assumption on } \phi\text{)} \\ &\Leftrightarrow \mathrm{PA}_0 \vdash \neg \Delta_\phi && \text{(by Proposition 7.4.1)} \end{aligned}$$

which is a contradiction to the assumption that T is a consistent theory that contains PA_0. □

COROLLARY 7.4.5 (Undecidability of first-order logic). *There exists a finite signature τ such that*

$$\{\ulcorner\phi\urcorner \mid \phi \text{ is valid } \tau\text{-sentence}\}$$

is undecidable.

7.5. Gödel's First Incompleteness Theorem

The following is Gödel's first incompleteness theorem [**8**] in an elegant strengthening that was found by Rosser [**22**].

THEOREM 7.5.1 (Gödel-Rosser). *Let T be a consistent and recursive τ_{Arithm}-theory that contains PA_0. Then T is incomplete.*

PROOF. If T were complete, then T would be decidable by Theorem 7.2.3, contradicting Church's theorem (Theorem 7.4.4). □

Note that every model of ZF contains in particular $\mathbb{N}$ as an element, and that the relations 0, s, $+$, $\cdot$, $<$ on $\mathbb{N}$ are first-order definable in ZF, so the following should not come as a surprise.

COROLLARY 7.5.2. *Let T be a recursive and consistent $\{\in\}$-theory that contains ZF. Then T is incomplete.*

PROOF SKETCH. Construct a consistent and recursive $\{\in\} \cup \tau_{\mathrm{Arithm}}$-theory T' that contains T and PA_0, and apply Theorem 7.5.1. □

Exercises.

(93) Find a Turing machine M so that the question whether M halts, formalised in ZF, is independent from ZF.
(94) Can you modify your solution to Exercise 90 to prove Theorem 7.5.1?

7.6. Σ_1-Definability of Truth of Σ_1-Sentences

The formula True_{Σ_1} introduced in Proposition 7.6.1 below plays an important role in the formulation of Gödel's second incompleteness theorem. Starting from here, everything that follows is bonus material for the course, included for the interested reader.

PROPOSITION 7.6.1. *There exists a Σ_1-formula $\mathrm{True}_{\Sigma_1}(x)$ such that for every Σ_1-sentence ϕ*

$$\mathrm{PA} \vdash \big(\phi \Leftrightarrow \mathrm{True}_{\Sigma_1}(\underline{\ulcorner\phi\urcorner})\big).$$

We mention that here it is essential that we use Peano arithmetic PA rather than weak Peano arithmetic PA_0; also note that the assumption that ϕ is a Σ_1-sentence is necessary, because for general first-order formulas the statement would be false by Tarski's theorem on the non-definability of truth (Theorem 7.4.3).

7.7. Gödel's Second Incompleteness Theorem

Informally, Gödel's second incompleteness theorem states that the consistency of PA cannot be shown in PA (and similarly, that the consistency of ZF cannot be shown within ZF). In order to phrase this statement formally, we must clarify what we mean by 'showing the consistency of PA within PA'. Similarly as in [**10**], we follow the presentation of Ziegler [**29**].

Throughout this section, let T be a recursive τ_{Arithm}-theory that contains PA. To formulate the second incompleteness theorem of Gödel we have to construct a specific Σ_1-formula that expresses that a sentence is provable in T. We use the following sentences in the construction.

- $\mathrm{Ax}_T(x)$ for the Σ_1-formula that represents the set of all Gödel numbers of logical axioms and of sentences in T,
- $\mathrm{MP}(x, y, z)$ for the Σ_1-formula that represents the set of triples $(\ulcorner\phi\urcorner, \ulcorner\psi\urcorner, \ulcorner\delta\urcorner)$ with ϕ, ψ, δ such that δ is obtained from ϕ and ψ by modus ponens.

It is clear that such formulas exist by the representability theorem (Theorem 7.3.9). In our construction, we will also use the primitive recursive component function $(x)_i$ from Lemma 6.1.15: the representability theorem implies that there is a Σ_1-formula $\phi(x_1, x_2, x_3)$ that represents this function, and hence we may freely use it in terms of Σ_1-formulas.

Definition 7.7.1. Let $\mathrm{Proof}_T(n)$ be the formula

$$\exists a\, \forall i \leq n \big(\mathrm{Ax}_T((a)_i) \vee \exists j, k < i.\, \mathrm{MP}((a)_j, (a)_k, (a)_i) \tag{11}$$

$$\vee\, \mathrm{True}_{\Sigma_1}((a)_i)\big). \tag{12}$$

We write $\Box_T\phi$ for $\mathrm{Proof}_T(\underline{\ulcorner\phi\urcorner})$.

If we dropped in the formula $\mathrm{Proof}_T(n)$ the disjunct in (12), then the correspondingly modified sentence $\Box_T\phi$ would express exactly that ϕ has a formal proof in T. The addition of the disjunct in (12) is motivated by the following proposition, which fails otherwise (if PA is consistent; see Exercise 96).

Proposition 7.7.2. *Let ϕ be a Σ_1-sentence. Then*

$$\mathrm{PA} \vdash (\phi \Rightarrow \Box_T\phi).$$

Proof. If $\mathrm{PA} \vdash \phi$ then $\mathrm{PA} \vdash \mathrm{True}_{\Sigma_1}(\underline{\ulcorner\phi\urcorner})$ and hence $\mathrm{PA} \vdash \Box_T\phi$. □

The following properties (L1)-(L3) are called *Loeb's axioms*, or sometimes the *Hilbert-Bernays provability conditions.*

Theorem 7.7.3. *Let ϕ and ψ be first-order τ_{Arithm}-sentences. Then $\Box_T$ satisfies the following properties:*

$(L1)$	*If $T \vdash \phi$ then $T \vdash \Box_T\phi$*	*(necessitation)*
$(L2)$	$T \vdash (\Box_T\phi \wedge \Box_T(\phi \to \psi)) \Rightarrow \Box_T\psi$	*(internal modus ponens)*
$(L3)$	$T \vdash \Box_T\phi \Rightarrow \Box_T\Box_T\phi$	*(internal necessitation)*

Proof. (L1). If $T \vdash \phi$, then $\underline{N}_{\mathrm{st}} \models \Box_T\phi$ (also see Exercise 96 (1)). Since $\Box_T\phi$ is a Σ_1-sentence, it holds in any model of PA_0 by Theorem 7.3.8, so in particular in every model of T.

(L2): Let $\underline{A}$ be a model of $T \cup \{\Box_T\phi, \Box_T(\phi \Rightarrow \psi)\}$. Then there exists an element in A that represents a formal proof of ϕ, and an element that represents a formal proof of $\phi \Rightarrow \psi$, and since $\underline{A} \models \mathrm{PA}$ we may find an element that represents a formal proof of ψ by combining the formal proofs above and applying modus ponens. So $\underline{A} \models \Box_T\psi$; hence, the statement follows from the completeness theorem.

(L3): This is a special case of Proposition 7.7.2 since $\Box_T\phi$ is a Σ_1-sentence. □

COROLLARY 7.7.4. *Let ϕ and ψ be τ_{Arithm}-sentences.*

$$\textit{If } T \vdash \phi \Rightarrow \psi, \textit{ then } T \vdash \Box_T\phi \Rightarrow \Box_T\psi$$

PROOF. Assume $T \vdash \phi \Rightarrow \psi$. Then $T \vdash \Box_T(\phi \Rightarrow \psi)$ by (L1), and hence (L2) implies that $T \vdash \Box_T\phi \Rightarrow \Box_T\psi$. □

REMARK 7.7.5. The operator $\Box_T$ provides an abstract view on provability so that we may forget about the details of the definition of $\Box_T\phi$ as a τ_{Arithm}-sentence; this idea goes back to Solovay [**26**]. A good example that illustrates this is Corollary 7.7.4, where we have only used Loebs axioms. This perspective is formalised in *provability logic*, which in our case leads to the *modal logic K*; see e.g. [**2**] for a full treatment. For example, (L1) is known in modal logic under the name **N** and (L3) is known under the name **4**. Provability logic can be used to analyse other logics as well. In our case, the only fact that we still need to establish before we can fully work in the framework of provability logic is Theorem 7.7.6.

THEOREM 7.7.6 (Modal fixed points). *Let ϕ be a τ_{Arithm}-sentence. Then there exists a τ_{Arithm}-sentence ψ such that*

$$T \vdash \psi \Leftrightarrow (\Box_T\psi \Rightarrow \phi).$$

PROOF. There exists a τ_{Arithm}-formula $\theta(x)$ such that for any τ_{Arithm}-sentence η

$$T \vdash \theta(\underline{\ulcorner \eta \urcorner}) \Leftrightarrow (\Box_T\eta \Rightarrow \phi).$$

This follows from the observation that decoding Gödel numbers is primitive recursive and Theorem 7.3.9. We apply Corollary 7.4.2 to the formula θ and obtain

$$T \vdash \psi \Leftrightarrow \theta(\underline{\ulcorner \psi \urcorner})$$

which implies the statement. □

From now on, it suffices to work with Loeb's axioms and the modal fixed point theorem. This is why we drop the reference to T and the signature τ_{Arithm} from now on until the statement of Gödel's second incompleteness theorem (they do not play any role in the arguments that follow).

THEOREM 7.7.7 (Loeb's theorem). *Let ϕ be a first-order sentence. Then*

$$\vdash \Box(\Box\phi \Rightarrow \phi) \Rightarrow \Box\phi.$$

PROOF. By Theorem 7.7.6, there exists a first-order sentence ψ such that

$$\vdash \psi \Leftrightarrow (\Box\psi \Rightarrow \phi). \tag{13}$$

Hence,

$\vdash \Box\psi \Rightarrow \Box(\Box\psi \Rightarrow \phi)$	(by (13) and Corollary 7.7.4 (1))	(14)
$\vdash \Box\psi \Rightarrow \big(\Box\Box\psi \Rightarrow \Box\phi)\big)$	(by (14) and box distributivity)	(15)
$\vdash \Box\psi \Rightarrow \Box\phi$	(by (15) and internal necessitation)	(16)
$\vdash (\Box\phi \Rightarrow \phi) \Rightarrow (\Box\psi \Rightarrow \phi)$	(by (16))	(17)
$\vdash (\Box\phi \Rightarrow \phi) \Rightarrow \psi$	(by (17) and (13))	(18)
$\vdash \Box(\Box\phi \Rightarrow \phi) \Rightarrow \Box\psi$	(by (18) and (Corollary 13))	(19)
$\vdash \Box(\Box\phi \Rightarrow \phi) \Rightarrow \Box\phi$	(by (19) and (16))	(20)

which concludes the proof. □

REMARK 7.7.8. If we interpret $\Box\phi$ as

'I believe ϕ'

then Loeb's theorem corresponds to *modesty*[2]: you do not believe that your belief in ϕ would imply that ϕ is true, without first believing that ϕ is true.

COROLLARY 7.7.9. *Let ϕ be a first-order sentence. If $\vdash \Box\phi \Rightarrow \phi$, then $\vdash \Box\phi$.*

PROOF. If $\vdash \Box\phi \Rightarrow \phi$ then $\vdash \Box(\Box\phi \Rightarrow \phi)$ by (L1), so the statement follows from Theorem 7.7.7. □

COROLLARY 7.7.10. *Let ϕ be a first-order sentence. If $\vdash \neg\Box\phi$, then $\vdash \Box\phi$.*

Let $\mathrm{Con}_T := \neg\Box_T(0 = 1)$, expressing the consistency of the τ_{Arithm}-theory T.

THEOREM 7.7.11 (Gödel's Second Incompleteness Theorem). *Let T be a recursive τ_{Arithm}-theory that contains* PA. *If T is consistent, then $T \nvdash \mathrm{Con}_T$.*

PROOF. Suppose $T \vdash \mathrm{Con}_T$. Then Corollary 7.7.10 implies that $T \vdash \Box_T(0 = 1)$, so T is inconsistent. □

Again, we obtain the corresponding statement for ZF as a consequence.

COROLLARY 7.7.12. *Let T be a recursive and consistent $\{\in\}$-theory that contains ZF. Then $T \nvdash \mathrm{Con}(T)$.*

Exercises.

(95) Prove *box distributivity*[3]:

$$\vdash \Box(\phi \Rightarrow \psi) \Rightarrow (\Box\phi \Rightarrow \Box\psi).$$

(96) Let $\mathrm{Proof}'_T(n)$ be the formula $\mathrm{Proof}_T(n)$ but without the disjunct in (12), and let $\Box'_T\phi$ be defined with $\mathrm{Proof}'_T(n)$ instead of $\mathrm{Proof}_T(n)$. Prove that
   (a) for every τ_{Arithm}-sentence ϕ

$$T \vdash \phi \text{ if and only if } \underline{N}_{\mathrm{st}} \models \Box'_T\phi.$$

   (b) if ϕ is a Σ_1-sentence, then

$$\underline{N}_{\mathrm{st}} \models \phi \Rightarrow \underline{N}_{\mathrm{st}} \models \Box'_T\phi.$$

   (c) Show that $\Box'_T$ satisfies (L1) and (L2).
   (d) (*) Assuming that PA is consistent, show that Proposition 7.7.2 fails for $\Box'_T(n)$ instead of $\Box_T(n)$. In other words, there is a recursive τ_{Arithm}-theory T that contains PA and a Σ_1-sentence ϕ such that

$$\mathrm{PA} \nvdash (\phi \Rightarrow \Box'_T\phi).$$

[2]This interpretation of the operator $\Box$ gives rise to *doxastic logic.*

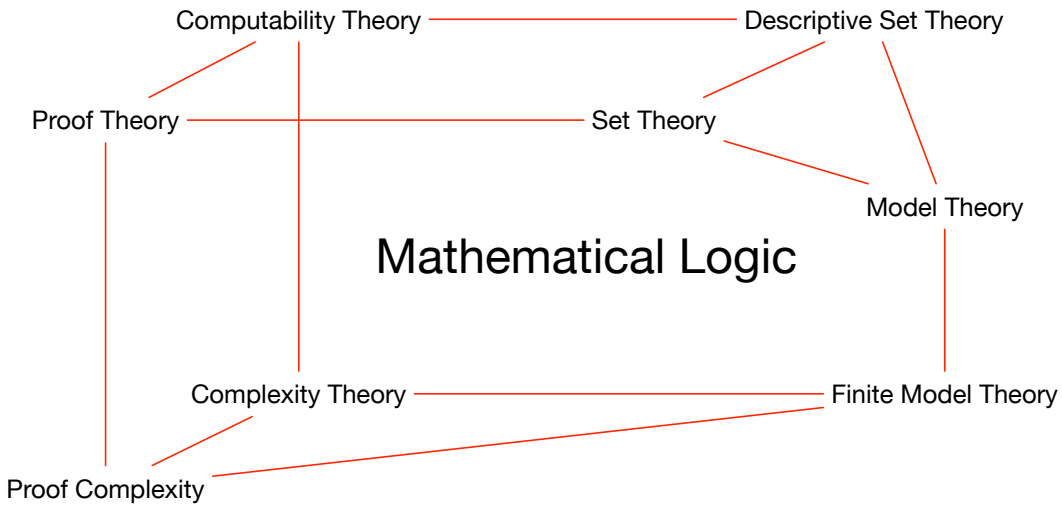
[3]In modal logic, box distributivity has the name **K**, not to be confused with K, which is $\{\mathbf{K}, \mathbf{N}\}$.

CHAPTER 8

Further Reading

There are several avenues to continue from here. I would like to mention *model theory*, *proof theory*, *recursion theory* (also known as *computability theory*), *complexity theory*, *proof complexity*, *finite model theory*, *set theory*, and *descriptive set theory*.

- *Model theory* is concerned with *tame* first-order theories, that is, theories that are well-behaved so that we can at least partially understand how the models of the theory look like. Model theory can then be applied in many areas of mathematics. I recommend [**12**] and its long version [**11**] and [**27**]. The compactness theorem plays a central role in model theory, while the completeness theorem does not. Model theory can help a lot if we want to prove that certain first-order theories are decidable.
- *Complexity theory* is concerned with the power of computation if we impose resource restrictions. Computation is still formalised with Turing machines; typical resources are then *space* and *time*: we may impose bounds on the running time or the space consumption of the Turing machine, depending on the size of the input. A more subtle interesting resource is *alternation*; the concept of *non-deterministic computation* being a special case. Complexity theory has plenty of open problems. Many of these open problems belong to the most difficult problems in all of mathematics, the most famous open problem being the P=NP problem. Here, P is the class of all problems that can be solved by a Turing machine in polynomial time, and NP is the class of all problems that can be solved by a non-deterministic Turing machine in polynomial time. We recommend the text books [**1**, **20**].



- *Proof complexity* is an area where the questions are motivated by complexity theory, but they are treated from the perspective of (propositional) proof theory. In this course we have seen an important propositional proof system, namely resolution. We have also hinted at a propositional proof system based on Modus Ponens, called *Frege's propositional calculus*. Many of the central questions in proof complexity are concerned with the size of shortest proofs in proof systems such as resolution and Frege. Similarly as complexity theory, this area is full of extremely difficult longstanding open problems. One of the most famous open problems is whether there are polynomial-size Frege proofs for all propositional tautologies, which corresponds to the question whether NP = coNP in complexity theory. We recommend Krajicek's book which is freely available as a pdf document [**16**].
- One of the starting points of *finite model theory* is the observation that the class NP, unlike the class P, has a simple logical characterisation which does not involve Turing machines, namely *Fagin's theorem*: it states that a class $\mathcal{C}$ of finite structures which is closed under isomorphism is in NP if and only if there exists a *second-order sentence* which holds on a finite structure if and only if the structure is in $\mathcal{C}$. Many other complexity classes can be characterised by a logic, in particular if we assume that the structure is ordered (such an order is implicitly given if we are interested in classes of words over some given finite alphabet). The questions in finite model theory are often motivated by complexity theory, but the methods often come from model theory; we recommend the text-books [**4**, **13**, **17**].
- *Proof theory:* in this first course we have seen a Hilbert-style proof system that allowed for a relatively short proof of the completeness theorem; however, we have also seen (mostly in the exercises) that already for very simple mathematical arguments it can be very difficult to translate them into our formalism. This motivates the search for proof systems where it is easier to come up with formal proofs. Such proof systems can be combined with automatic theorem provers so that tedious verification steps can be done by a computer. The correctness of these systems themselves can be formally verified as well! Such systems already exist, e.g., there is the system *Isabelle* developed by the university of Cambridge and TU München, or the system *Coq* developed by INRIA, CNRS, Ecole Polytechnique, Université Paris Diderot and others. The long-term vision is that mathematicians work interactively with such systems to eventually transform their proofs into formal proofs that can be checked efficiently by a program. A classic text book in the area is [**23**].
- *Set theory* is a deep area. One of the topics are *large cardinal properties*. Whether large cardinals exist cannot be proved in ZFC, and the existence of such cardinals can be viewed as a way to extend ZFC. One of the goals is to obtain a systematic understanding of the resulting extensions of ZFC. There are important connections with model theory and proof theory; I refer to the text-books [**14**, **19**].
- *Recursion theory (computability theory)*: what can be computed by a Turing machine if the machine has access to an oracle that solves the halting problem, or the complement of the halting problem? Of course, there are again function from $\mathbb{N} \to \mathbb{N}$ that cannot be computed in this way, simply by a counting argument. In fact, there is a complicated landscape of larger and larger sets of functions over the natural numbers; the beauty of the subject is that, unlike in complexity theory, we have proofs showing that these

classes are indeed different. Recursion theory is quite old [**9**]; recent developments include for instance *continuous computability theory*, *Kolmogorov complexity*, and *reverse mathematics*.

- *Descriptive set theory* is the study of certain classes of well-behaved subsets of $\mathbb{R}$, the set of real numbers. Exercises 47 and 48 in this course are basic facts from descriptive set theory. The subject exists in a classical version [**15**] and an effective version [**6**] which creates a link with recursion theory. Descriptive set theory has applications e.g. in functional analysis and the study of group actions.

Bibliography

[1] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 594 pages.

[2] G. Boolos. *The Logic of Provability*. Cambridge University Press, 1993.

[3] G. Cantor. Über unendliche, lineare Punktmannigfaltigkeiten. *Mathematische Annalen*, 23:453–488, 1884.

[4] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer, Berlin, Heidelberg, New York, 1999. Second edition.

[5] A. Fraenkel. Zu den Grundlagen der Cantor-Zermeloschen Mengenlehre (wesentlich Wiedergabe eines Vortrags a. d. deutschen Mathematikertag, Jena 921, vgl. Jahresb. d. D. Math.-Ver., 30 [1921], 97 f.). *Math. Annalen*, 86:230–237, 1922.

[6] S. Gao. *Invariant Descriptive Set Theory*. Pure and applied mathematics. Taylor and Francis, 2008.

[7] K. Gödel. Die Vollständigkeit der Axiome des logischen Funktionenkalküls. *Monatshefte für Mathematik (in German)*, 37(1):349–360, 1930.

[8] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I. *Monatshefte für Mathematik und Physik*, 38(1):173–198, 1931.

[9] J. Hartley Rogers. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill Book Co, 1967.

[10] M. Hils and F. Loeser. *A first Journey through Logic*, volume 89 of *Student Mathematical Library*. American Mathematical Society, Providence, RI, September 2019.

[11] W. Hodges. *Model theory*. Cambridge University Press, 1993.

[12] W. Hodges. *A shorter model theory*. Cambridge University Press, Cambridge, 1997.

[13] N. Immerman. *Descriptive Complexity*. Graduate Texts in Computer Science, Springer, 1998.

[14] T. Jech. *Set theory*. Springer Monographs in Mathematics. Springer-Verlag, Berlin, 2003. The third millennium edition, revised and expanded.

[15] A. Kechris. *Classical descriptive set theory*, volume 156 of *Graduate Texts in Mathematics*. Springer, 1995.

[16] J. Krajicek. *Proof complexity*. Cambridge University Press, 2019. Encyclopedia of Mathematics and Its Appplications, Vol.170.

[17] L. Libkin. *Elements of Finite Model Theory*. Springer, 2004.

[18] Y. V. Matiyasevich. *Hilbert's Tenth Problem*. MIT Press, Cambridge, Massachusetts, 1993.

[19] Y. N. Moshovakis. *Notes on Set Theory*. Springer Science and Business Media, 1994.

[20] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[21] C. René and D. Lascar. *Logique mathématique : cours et exercices; préface de J.-L. Krivine, 1. Calcul propositionnel, algèbres de Boole, calcul des prédicats*. Axiomes collection de logique mathématique. Masson, Paris, 1993.

[22] J. B. Rosser. Extensions of some theorems of Gödel and Church. *Journal of Symbolic Logic*, 1:87–91, 1936.

[23] K. Schütte. *Proof theory*. Springer-Verlag Berlin Heidelberg, 1977.

[24] S. Shelah and E. Milner. Graphs with no unfriendly partitions. *Special issue 'a tribute to Paul Erdös', Hungarian Academy of Sciences*, 1990.

[25] T. Skolem. Einige Bemerkungen zur axiomatischen Begründung der Mengenlehre. *Matematikerkongressen i Helsingfors*, pages 217–232, 1922.

[26] R. M. Solovay. Provability interpretations of modal logic. *Israel Journal of Mathematics*, 25:287–304, 1976.

[27] K. Tent and M. Ziegler. *A course in model theory*. Lecture Notes in Logic. Cambridge University Press, 2012.

[28] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42:230–265, 1937.

[29] M. Ziegler. *Mathematische Logik, 2. Auflage*. Birkhäuser, 2017.