Introduction to Mathematical Logic

J. Adler, J. Schmid

May 2, 2007

Contents

Chapter 1

Introduction

Almost everybody in the Western world is taught a certain portion of mathematics in her/his school career. For reasons that are not clear, this exposure to mathematics is in the majority of the cases a source of uneasiness rather than joy. Accordingly, people whose experience was negative do not mind if most of what they have learned quickly sinks into oblivion after their leaving school. Still, there are a few particulars that do not share this unworthy destiny. The name Pythagoras and his famous theorem, the numbers $\sqrt{2}$ and π are particulars that find a permanent place in every mind, no matter how adverse to mathematics it may be. This also holds for the notion of proof.

Mathematics and proofs are intimately related. There is no mathematics without proofs. Insistence on proofs is what distinguishes mathematics from any other academic discipline. A proof is an argument that establishes the truth of a claim. It starts from assumptions and proceeds to a conclusion by applying certain rules to the assumptions. Neither this procedure, nor the requirement that people have to agree on what is accepted as a proof are unique to the field of mathematics. The difference between mathematics and any other discipline lies in the rigor that is imposed on a mathematical argument in order for it to be accepted as a proof.

Although mathematicians write and read proofs without having encountered an exact definition of this notion, there is in most cases a consensus on whether an argument is acceptable as a proof, and if a proof is not acceptable, what gaps need to be filled.

Mathematics in the spirit of being the "science of proofs" started in the Greek period about 600 BC[1] and was pursued successfully until the mid nineteenth century. In this period people like René Descartes, Leonard Euler, Carl Friedrich Gauss and Bernhard Riemann, to name some of the most important figures, accomplished their ground-breaking contributions to mathematics.

A radical change in the world of mathematics occurred in the eighteen-seventies when Georg Cantor created set theory to lay a foundation for analysis, a subject which had gone through an impetuous development for the preceding hundred years. As we shall see in the following, the importance of Cantor's theory is due firstly to new results obtained using set theory and secondly to inconsistencies that appeared in it. These inconsistencies threatened the whole of mathematics because all of mathematics can be expressed with set theory.

Cantor obtained one of his first results in this new area in 1873, when he demonstrated that the rational numbers can be placed in a one-to-one relationship with the natural numbers, i.e. that the rational numbers are countable.

Using the same notions of set theory Cantor also proved that in a certain sense almost all real numbers are transcendental[2]. He showed that (1) the set of real numbers is uncountable and that (2) there are only countably many real numbers that are not transcendental. Therefore, the large remainder consists of transcendental numbers. Thus, Cantor's finding comprehends a result of Joseph Liouville's who established in 1851 the existence of transcendental numbers. Liouville showed that $\Sigma_{n=1}^{\infty}(\frac{1}{10})^{n!}$ is transcendental.

¿From 1879 to 1884 Cantor published five fundamental papers on set theory. Amongst other things, he developed the theory of transfinite numbers[3].

[1] Keith Devlin writes in *The Math Gene* "Thales of Miletus introduced the idea that the precisely stated assertions of mathematics could be logically proved by formal argument. For the Greeks, this approach culminated in the publication, around 350 BC, of Euclid's mammoth thirteen-volume text *Elements*, reputedly the second most widely circulated book of all time after the Bible."

[2] Transcendental numbers are real numbers that are not the root of a polynomial with integer coefficients.

[3] Infinite sets on which a specific order is imposed. The set of natural numbers is the smallest transfinite number.

Although a substantial part of the mathematical community was dismissive of Cantor's new theory, which produced many results that were not intuitive, set theory was nevertheless recognized as a branch beneficial to the advancement of mathematics, because it offered altogether new methods of proof.

Cantor was not restrictive as to the principles he admitted to define sets. In his theory, the set of all sets was permitted to exist. This is the set from which Bertrand Russell derived his famous antinomy, the contradictory set that consists of all sets that do not contain themselves. This discovery caused a great stir and called for the laying of a consistent foundation for mathematics.

One reaction was *intuitionism*. Intuitionists, led by the Dutch mathematician L. E. J. Brouwer, disapproved of set theoretic principles, because they are strongly *non-constructive*. In a constructivist argument, the assertion of the existence of objects with certain properties is admitted only if a concrete object with these properties can be constructed. The position of the constructivist implied that infinitary methods, and with them most of the mathematics existing at that time, had to be discarded until they could be proved using *finitary*[4] methods.

David Hilbert, one of the leading mathematicians at the end of the nineteenth century, agreed with Brouwer that there was a need for a proof that classical mathematics was correct. However, he did not want to be "expelled from the paradise Cantor had created", that is, to abstain from using the powerful new principles. His plan, called *Hilbert's programme*, was to prove, using finitary methods, that classical mathematics can be reduced to the manipulation of strings of symbols according to certain rules. The assertion of the existence of an object would be a string of symbols. Its proof would start from the finite strings formalizing the axioms and would arrive at the string representing the assertion after a finite number of applications of the specified rules. Consistency could be expressed by prohibiting certain combinations of strings from occurring. Finitary methods could be used to show that the rules to manipulate the strings of symbols would not produce forbidden combinations. Intuitionists would also have to accept this proof due to its realization by finitary character.

[4] We refrain from going into the particulars of the term "finitary".

More concretely, Hilbert's programme consists of the following four steps:

Hilbert's programme

(S1) Specification of a formal language in which classical mathematics can be described;

(S2) Specification of a complete deductive system (a set of logical axioms and a set of inference rules) to manipulate the strings of symbols specified in (S1);

(S3) Specification of a complete system of mathematical axioms;

(S4) Proof of the consistency of the formal system in (S1)-(S3).

All notions occurring in this programme are required to be finite or recursively enumerable[5]. This means:

(S1): The alphabet and the notions of the formal language are recursively enumerable.

(S2): The set of logical axioms and the set of rules and hence the set of conclusions from a recursive set of formulae is recursively enumerable.

(S3): The set of mathematical axioms is recursively enumerable.

(S4): The proof has to be performed by finitistic means.

Completeness in (S2) signifies that adding further logical axioms or inference rules[6] does not enable us to infer more from a given set of mathematical axioms without allowing contradictions to occur. Completeness in (S3) implies that every sentence φ or its negation $\neg\varphi$ in the formal language specified in (S1) is deducible from a complete deductive system. Otherwise, the set of axioms could be properly extended by adding either φ or $\neg\varphi$.

[5] A set, for instance a set of strings over an alphabet, is recursively enumerable if there is an algorithm (or a computer program) that yields all elements of the set. The set of prime numbers is recursively enumerable as there is an algorithm that computes all primes. This notion will be explained in more detail in Section 5.2.

[6] There are also so-called logical axioms that belong to a deductive system, cf. Chapter 4.

The aim of Hilbert's programme was the development of an unassailable foundation for the whole of mathematics. Hilbert's intention was to realize his program, and then be free to return to advancing mathematical knowledge. In the 1920's many mathematicians became actively involved in Hilbert's project and parts of it were indeed accomplished.

Hilbert's programme foundered when, in 1931, Kurt Gödel published his famous *incompleteness theorems*.

Gödel's incompleteness theorems (Gödel 1931)

1. For every consistent recursive axiom system[a] that contains elementary arithmetic there is an arithmetic sentence that is true but not provable in the system.

2. The consistency of the axiom system is such a sentence.

[a]That is, there is an algorithm that for a given sentence computes whether the sentence belongs to the axiom system.

These theorems have the implication that for intrinsic theoretical reasons (S3) and (S4) can never be accomplished!

Let us consider what the situation looked like after the publication of Gödel's incompleteness theorems.

(S1) of Hilbert's programme can be realized relatively easily.

(S2) was shown by Gödel. This result is called the "Completeness Theorem of first-order logic".

As shown by Gödel (S3) already fails for relatively modest parts of mathematics, let alone an axiom system for the whole of mathematics. As soon as a recursive system of axioms is strong enough to describe the basic properties of the natural numbers, there is a true statement that cannot be proved from it.

(S4) can be regarded as a special case of (S3), the sentence being the consistency of the system.

The impossibility of fulfilling (S3) and (S4) means that we cannot prove, and will never be able to prove, that mathematics is consistent. However, this does not prevent researchers from pursuing mathematics, enjoying doing so and proving results that have important applications. If a contradiction occurs somewhere, then it has to be resolved by changing the underlying axioms. Afterwards the pursuit of theorems can be continued although, the appearance of new contradictions cannot be excluded.

In Chapters 2 and 3 we will show that the common mathematical systems can be formalized, that is we will carry out (S1). This includes set theory, in which ordinary mathematics can be formalized. Cf. Example 2.1.4.

Chapter 6 is devoted to Leon Henkin's proof of the Completeness Theorem of first-order logic. That is, (S2) is carried out. Chapters 4 and 5, in which our deductive system is introduced and its first properties are shown, prepare the proof of the Completeness Theorem.

A rigorous analysis of (S3) and (S4) is beyond the scope of this course. The interested reader is advised to consult the literature.

We end this introduction by a fictitious discussion between an "ideal mathematician" (I.M.) and a student quoted from the excellent book "The Mathematical Experience"[7].

Student:	Sir, what is a mathematical proof?
I.M. :	You don't know *that*? What year are you in?
Student:	Third-year graduate.
I.M. :	Incredible! A proof is what you've been watching me do at the board three times a week for three years! That's what a proof is.
Student:	Sorry, sir, I should have explained. I'm in philosophy, not math. I've never taken your course.
I.M. :	Oh! Well, in that case - you have taken *some* math, haven't you? You know the proof of the fundamental theorem of calculus - or the fundamental theorem of algebra?
Student:	I've seen arguments in geometry and algebra and calculus that were called proofs. What I'm asking you for isn't *examples* of proof, it's a definition of proof. Otherwise, how can I tell what examples are correct?

[7]by Philip J. Davis and Reuben Hersh

I.M. : Well, this whole thing was cleared up by the logician Tarski, I guess, and some others, maybe Russell and Peano. Anyhow, what you do is, you write down the axioms of your theory in a formal language with a given list of symbols or alphabet. Then you write down the hypothesis of your theorem in the same symbolism. Then you show that you can transform the hypothesis step by step, using the rules of logic, till you get the conclusion. That's a proof.

Student: Really? That's amazing! I've taken elementary and advanced calculus, basic algebra, and topology, and I've never seen that done.

I.M. : Oh, of course no one ever really *does* it. It would take forever! You just show that you could do it, that's sufficient.

Student: But even that doesn't sound like what was done in my courses and textbooks. So mathematicians don't really do proofs, after all.

I.M. : Of course we do! If a theorem isn't proved, it's nothing.

Student: Then what is a proof? If it's this thing with a formal language and transforming formulae, nobody ever proves anything. Do you have to know all about formal languages and formal logic before you can do a mathematical proof?

I.M. : Of course not! The less you know, the better. That stuff is all abstract nonsense anyway.

Student: Then really what *is* a proof?

I.M. : Well, it's an argument that convinces someone who knows the subject.

Student: Someone who knows the subject? Then the definition of proof is subjective, it depends on particular persons. Before I can decide if something is a proof, I have to decide who the experts are. What does that have to do with proving things?

I.M. : No, no. There's nothing subjective about it! Everybody knows what a proof is. Just read some books, take courses from a competent mathematician, and you'll catch on.

Student: Are you sure?

I.M. : Well - it is possible that you won't, if you don't have any aptitude for it. That can happen, too.

Student: Then *you* decide what a proof is, and if I don't learn to decide in the same way, you decide I don't have any aptitude.

I.M. : If not me, then who?

To test your knowledge, solve the multiple choice test of this chapter.

Chapter 2

Formal languages

In order to consider proofs as mathematical objects we have to introduce formal languages. As is the case with natural languages like English and German, formal languages consist of an alphabet to form strings that are candidates for meaningful entities, and rules (corresponding to the grammar of a natural language) that determine which of the candidates qualify as such. These constructs are called *terms* and *formulae* and correspond to the words and sentences of natural languages.

Having formulae at our disposal we can consider finite sequences of them. Proofs will be finite sequences of formulae satisfying certain conditions. We will therefore be able to define proofs formally within the framework we introduce in this chapter.

We will talk about the mathematical objects expressed in a formal language in the same way we formulate statements in geometry and calculus, using "everyday English" together with expressions such as "it follows," "if and only if," "implies." This language is called the *meta language*.

The formal languages we treat in this course are so-called *first-order languages with equality*. As the term "first-order" suggests there are also second and higher order languages. The alphabet of a first-order language contains only a single type of variable whereas for instance a second-order language contains two type of variables. A second-order variable then stands for an object that is a collection of objects that stand for first-order variables.

2.1 Alphabets

An alphabet defines what symbols can be used to form terms, formulae and finally also deductions. Given the specific rules to form them the mathematical objects we are going to investigate are determined by the alphabet. The entirety of these objects pertaining to an alphabet will be denoted by the term *formal language*.

Definition 2.1.1. The *alphabet* of a formal language $\mathcal{L}$ consists of the following symbols:

Connectives	$\neg, \wedge$	
Quantifier	$\forall$	*logical symbols*
The equality symbol	$\doteq$	
Variable symbols	$v_n, n \in \mathbb{N}$	
Relation symbols	$R_i \quad (i \in I)$	
Function symbols	$f_j \quad (j \in J)$	*non-logical symbols*
Constant symbols	$c_k \quad (k \in K)$	
Auxiliary symbols	parentheses, comma	

Remark 2.1.2. The index sets I, J, K just have to be of the right size to name the relation, function and constant symbols needed. This is the only condition they must satisfy. Such an index set can be finite (also empty), countably infinite or even uncountable.

Every alphabet of a first-order language is infinite since there are infinitely many variable symbols $v_n, n \in \mathbb{N}$. This can be avoided by considering variable symbols that consist only of the symbols v and $'$: $v, v', v'', \ldots$

Definition 2.1.1 warrants further remarks. The auxiliary symbols parentheses and commas will be used to build expressions in the same way they were used to write down functions in high-school. Consider the function $f(a, b) = \sqrt{a^2 + b^2}$ calculating the absolute value of the complex number $a + bi$. It would be possible to write fab instead of $f(a, b)$ to denote this function, but avoiding auxiliary symbols would be at the cost of legibility.

The connectives will be used to build new formulae from given ones, with $\neg$ having the intended meaning "not," $\wedge$ the intended meaning "and." The same holds for the quantifier $\forall$, having the intended meaning "for all."

Not surprisingly the intended meaning of $\doteq$ is "equal." Thus $\doteq$ will be used to equate strings that are supposed to have the same meaning. The ordinary symbol for equality $=$ will be used to denote equality in the meta language.

The symbols discussed so far are part of the alphabet of every formal language we will consider. What distinguishes different formal languages are the constant, relation and function symbols. These types of symbols are denoted as *nonlogical symbols*, whereas the others (except the auxiliary symbols) are referred to as *logical symbols*.

That $\doteq$ is included among the logical symbols, and is not treated as an ordinary relation symbol, will be motivated by the special role equality will play among the relation symbols (cf. Definition 3.2.6 and Remark 3.2.7).

2.1.1 The relationship between alphabets and formal languages

In order to know how to build a term with a function symbol f_j, or a formula with a relation symbol R_i, the arity[1] of the symbol must be known. Therefore, in order to determine a formal language two functions $\lambda : I \to \mathbb{N} \setminus \{0\}$ and $\mu : J \to \mathbb{N} \setminus \{0\}$ fixing the arity of each function and relation symbol must be specified.

Definition 2.1.1 and the preceding remarks give rise to the following definition:

Definition 2.1.3. A *formal (first-order) language (with equality)* $\mathcal{L}$ is characterized by

- the language's alphabet,
- three index sets I, J, K,
- two arity functions $\lambda : I \to \mathbb{N} \setminus \{0\}$ and $\mu : J \to \mathbb{N} \setminus \{0\}$, where $\lambda(i)$ is the arity of the i-th relation symbol R_i and $\mu(j)$ the arity of the j-th function symbol f_j.

[1] The arity of a function or relation in an ordinary mathematical structure denotes the number of arguments. As function and relation symbols refer to functions and relations respectively, the arity of each such symbol must be specified. It fixes the number of terms needed to form a term with a function symbol or a formula with a relation symbol.

Notation: $\mathcal{L} = (\lambda, \mu, K)$

If $\mathcal{L}' = (\lambda', \mu', K')$ is another formal language such that $I \subseteq I'$, $J \subseteq J'$ and $K \subseteq K'$ and $\lambda = \lambda'|I$, $\mu = \mu'|J$ then we write $\mathcal{L} \subseteq \mathcal{L}'$ and say that $\mathcal{L}'$ is an extension of $\mathcal{L}$.

A formal language $\mathcal{L}$ is completely determined by the triple (λ, μ, K): I and J are the domains of the functions λ and μ and can therefore be retrieved from them. The auxiliary and logical symbols as well as the variables are the same for all formal first-order languages with equality.

With such a definition at hand it is possible to formulate theorems that relate properties of the triple (λ, μ, K) and the corresponding formal language $\mathcal{L}$.

Example 2.1.4. Let $\mathcal{L} = (\lambda, \mu, K)$ with $I = \{0\}$, $J = \emptyset$, $K = \emptyset$ and $\lambda(0) = 2$. That is, $\mathcal{L}$ is a formal language with only one nonlogical symbol R_0, a 2-ary relation symbol. With regard to a possible interpretation of the nonlogical symbol we write suggestively $\in$ instead of R_0.

The suggestive symbol $\in$ was chosen since this formal language is the language of set theory, in which most of contemporary mathematics can be formalized.

Example 2.1.5 (Standard language). Let $\mathcal{L} = (\lambda, \mu, \mathcal{K})$ with $I = \{0\}$, $J = \{0, 1, 2\}$, $K = \{0, 1\}$ and $\lambda(0) = 2$, $\mu(0) = 1$, $\mu(1) = \mu(2) = 2$.

The non-logical symbols consist of the 2-ary relation symbol R_0, the unary function symbol f_0, two binary function symbols f_1 and f_2, and finally the constant symbols c_0 und c_1. With regard to a possible interpretation of the nonlogical symbols we write suggestively:

$$\begin{array}{lll} \leq & \text{instead of} & R_0 \\ - & \text{instead of} & f_0 \\ + & \text{instead of} & f_1 \\ \cdot & \text{instead of} & f_2 \\ 0 & \text{instead of} & c_0 \\ 1 & \text{instead of} & c_1 \end{array}$$

Note that so far the symbols do not have any meaning.

This formal language is going to be used in many examples of this course. We call it therefore the *standard language*.

Remark 2.1.6. When talking about variable symbols we shall use the shorter term "variable," likewise for constant, function and relation symbols.

2.2 Terms and formulae

2.2.1 Terms

In this section we determine which strings qualify as terms and formulae respectively.

We introduced formal languages in order to develop a tool to analyze statements about mathematical structures. Such a statement could be the claim that in a given structure a certain equation has a specific solution or that it cannot be proved that a property α_1 implies a property α_2. As such statements talk about objects and properties of that structure it must be possible to refer in the corresponding formal language to these objects and properties.

The strings of symbols of a formal language $\mathcal{L}$ that refer to these objects are called *$\mathcal{L}$-terms*.

Definition 2.2.1 (Terms). Let $\mathcal{L} = (\lambda, \mu, \mathcal{K})$ be a formal language. Then a finite string t of symbols from $\mathcal{L}$ is an *$\mathcal{L}$-term* if either

1. t is a variable v_n, $n \in \mathbb{N}$.
2. t is a constant c_k, $k \in K$.
3. t is $f_j(t_1, t_2, \ldots, t_{\mu(j)})$ for $\mathcal{L}$-terms $t_1, t_2, \ldots, t_{\mu(j)}, j \in J$.

If the formal language is clear from the context we simply speak of a term instead of an $\mathcal{L}$-term.

Remark 2.2.2. Definition 2.2.1 is an instance of a *definition by induction.* The set of terms is constructed "bottom-up" by starting with variables and constants. Then this set is extended by adding the strings that are obtained by concatenating function symbols with elements that are already in the set, that is, with terms consisting of fewer symbols. This process is iterated. The number of symbols in a string does not include commas and parentheses.

In other words, the set of $\mathcal{L}$-terms of a formal language is the smallest set of strings of symbols of $\mathcal{L}$ containing on the one hand all variable and constant symbols and on the other hand with $t_1, \ldots, t_{\mu(j)}$ also $f_j(t_1, \ldots, t_{\mu(j)})$, $j \in J$.

Exercise 2.2.3. Explain the connection between a definition by induction and the principle of complete induction.

We use our "standard language" introduced in Example 2.1.5 to illustrate the definition of term (Definition 2.2.1) in the next example.

Example 2.2.4. Let $\mathcal{L}$ be the language introduced in Example 2.1.5. We use the infix-notation to stick to the usual mathematical notation. This means that we write

$$\begin{array}{lll} t_1 + t_2 & \text{instead of} & +(t_1, t_2) \\ t_1 \cdot t_2 & \text{instead of} & \cdot\,(t_1, t_2) \\ t_1 \leq t_2 & \text{instead of} & \leq (t_1, t_2) \end{array}$$

Then the following is clear:

$$\begin{array}{ll} 0,\ -1,\ 1+1,\ v_0+1,\ v_0 \cdot (v_0+1) & \text{are } \mathcal{L}\text{-terms} \\ 2,\ +1,\ v_0+,\ v_0 - v_1 & \text{are not } \mathcal{L}\text{-terms} \end{array}$$

2.2.2 Formulae

At the beginning of this section we said that by means of an appropriate formal language properties of mathematical structures would be analyzed. The strings of symbols of a formal language $\mathcal{L}$ that refer to these properties are called $\mathcal{L}$-*formulae*.

Definition 2.2.5 (Formulae). Let $\mathcal{L} = (\lambda, \mu, \mathcal{K})$ be a formal language. Then a finite string φ of symbols from $\mathcal{L}$ is an $\mathcal{L}$-*formula* if either

1. φ is $t_1 \doteq t_2$ for $\mathcal{L}$-terms t_1, t_2,
2. φ is $R_i(t_1, t_2, \ldots, t_{\lambda(i)})$ for $\mathcal{L}$-terms $t_1, t_2, \ldots, t_{\lambda(i)}, i \in I$,
3. φ is $(\neg\alpha)$ for an $\mathcal{L}$-formula α,
4. φ is $(\alpha \wedge \beta)$ for $\mathcal{L}$-formulae α and β,

5. φ is $(\forall v_n \alpha)$ for an $\mathcal{L}$-formula α and any variable v_n, $n \in \mathbb{N}$.

If the formal language is clear from the context we simply speak of a formula instead of an $\mathcal{L}$-formula.

Formulae that satisfy one of the first two clauses are called *atomic formulae.*

In Definition 2.1.1 the symbol $\forall$ was denoted as a quantifier. In a formula $\forall v_n \alpha$ we use "quantifier" to denote both $\forall$ and $\forall v_n$.

Atomic formulae consist of terms and relation symbols. They refer to elementary relationships between the objects of mathematical structures.

Parentheses are necessary to avoid ambiguities when clauses 3-5 are applied more than once.

In the following example we use our standard language again (cf. Example 2.1.5).

Example 2.2.6. Let $\mathcal{L}$ be the language introduced in Example 2.1.5. Then

$v_0 \doteq 1,\ 0 \leq 1,\ 1 + 1 \leq 0$	are atomic $\mathcal{L}$-formulae
$v_0 \doteq 1,\ \forall v_0\,(v_0 \doteq 1),\ \forall v_0\,\neg(v_0 + v_0 \leq 0)$	$\mathcal{L}$-formulae
$v_0 + v_1,\ v_0 \doteq,\ \forall v_0 \wedge \doteq 0$	are not $\mathcal{L}$-formulae

Since terms and formulae are defined by induction, properties of terms and formulae can be proved by so-called *induction on the complexity* of a term and a formula respectively.

For terms this means that one first has to show that a property holds for variables and constants and secondly, for all $j \in J$, that if it holds for terms $t_1, \ldots, t_{\mu(j)}$ then also for $f_j(t_1, \ldots, t_{\mu(j)})$.

E.g., the simple fact that every formula contains at least the symbol $\doteq$ or some relation symbol R_i may be shown as follows: This is obviously true for atomic formulae, that is, by the first two clauses of the definition of formulae (cf. Definition 2.2.5). Assume now that the claim is true for formulae α and β. It follows immediately that this is also the case for $\neg\alpha$, $\alpha \wedge \beta$ and $\forall v_n \alpha$. Hence, the property under consideration is preserved also under the application of clauses 3-5 of Definition 2.2.5).

For a further example, consider Exercise 8.1.3 in Section 8.1

2.2.3 Abbreviations and conventions

In order to increase the legibility of formulae we introduce "derived" connectives $\vee$, $\rightarrow$, $\leftrightarrow$ and quantifier $\exists$ respectively into the alphabets of the first-order languages. The expressions in parentheses on the right signify the intended meaning of these new symbols; this will be made precise in Chapter 3.

$(\varphi \vee \psi)$	for	$(\neg(\neg\varphi \wedge \neg\psi))$	(read as "or")
$(\varphi \rightarrow \psi)$	for	$((\neg\varphi) \vee \psi)$	(read as "implies")
$(\varphi \leftrightarrow \psi)$	for	$((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi))$	(read as "equivalent")
$(\exists v_n \, \varphi)$	for	$(\neg(\forall v_n(\neg\varphi)))$	(read as "exists")

That the derived connectives indeed behave as they are read will be shown in Subsection 3.3.2. To find the corresponding argument for the derived quantifier $\exists$ is the task of Exercise 3.2.8.

The following conventions are introduced to simplify the notation of formulae and to render it more readable.

C1 $\neg$ binds more strongly than $\wedge$ and $\vee$; $\wedge$ and $\vee$ bind more strongly than $\rightarrow$ and $\leftrightarrow$.

C2 We often write $(\neg(\alpha))$ instead of $(\neg\alpha)$ to improve readability, e.g.
$\neg(f_1((t_2, t_4) + f_2(t_1, t_2) \doteq f_3(t_4, t_6))$
instead of
$\neg f_1((t_2, t_4) + f_2(t_1, t_2) \doteq f_3(t_4, t_6)$

C3 $t_1 \not\doteq t_2$ instead of $\neg(t_1 \doteq t_2)$.

C4 $t_1 R_i t_2$ and $t_1 f_j t_2$ instead of $R_i(t_1, t_2)$ and $f_j(t_1, t_2)$ respectively for $i \in I$, $j \in J$ and $\lambda(i) = \mu(j) = 2$ (infix-notation).

C5 $(\forall v_1, \ldots, v_n \varphi)$ instead of $(\forall v_1 \ldots (\forall v_n \varphi) \ldots)$
and
$(\exists v_1, \ldots, v_n)$ instead of $(\exists v_1 \ldots (\exists v_n \varphi) \ldots)$.

C6 Left association of parentheses for $\wedge$ and $\vee$, e.g. $(\alpha \wedge \beta \wedge \gamma)$ and $(\alpha \vee \beta \vee \gamma)$ instead of $((\alpha \wedge \beta) \wedge \gamma)$ and $((\alpha \vee \beta) \vee \gamma)$, respectively.

C7 Parentheses are dropped whenever they can be restored unambiguously.

Example 2.2.7. Applying these rules to $((((\neg(t_1 \doteq t_2)) \wedge R_i(t_1, t_2)) \wedge \beta) \rightarrow (\forall u \forall v \, \delta))$ we obtain the much more legible $(t_1 \not\doteq t_2 \wedge t_1 R_i t_2 \wedge \beta) \rightarrow \forall u, v \, \delta$. And we write $\neg\psi \vee \forall v \neg\varphi$ instead of $((\neg\psi) \vee (\forall v(\neg\varphi)))$.

We end this subsection by the following conventions which concern the meta language. These conventions facilitate the discussion of formal languages and the proving of results about them.

$t_1, t_2, ...$	$\mathcal{L}$-terms
$\varphi, \psi, \sigma, \alpha, \beta, ...$	$\mathcal{L}$-formulae
$\Sigma, \Theta, \ldots$	sets of $\mathcal{L}$-formulae
$u, v, x, y, z, ...$	variables
$\mathrm{Tm}(\mathcal{L})$	the set of all $\mathcal{L}$-terms
$\mathrm{Fml}(\mathcal{L})$	the set of all $\mathcal{L}$-formulae
$\mathrm{Vbl}(\mathcal{L})$	the set of all $\mathcal{L}$-variables (which is the same for all formal languages $\mathcal{L}$)

Solve the exercises in Section 8.1!

2.3 Sentences

As already stated in Section 2.2, formulae refer to statements about objects in mathematical structures. However, so far we have not defined what we mean by a mathematical structure. Without such a definition it is not possible to give a formal definition of how terms and formulae refer to mathematical structures.

However, outside this course we have already encountered many different mathematical structures, such as the number systems of the natural, rational, real and complex numbers, and their pertaining laws. Furthermore, the intended meaning of the logical symbols has been indicated and the inductive definitions of terms and formulae are very suggestive with regard to how terms and formulae will refer to objects and properties of mathematical structures. Therefore, the intuitive idea of the meaning of terms and formulae suffices to motivate the definition of a subset of $\mathrm{Fml}(\mathcal{L})$ whose members are called *sentences*.

As formulae refer to statements about mathematical structures, and statements about mathematical structures are either true or false, the idea to assign one of the truth values “true” or “false” to a formula with regard to a mathematical structure is not far-fetched. An assignment of a truth value to a formula may depend on the objects the variables in the formula refer to.

Sentences will be certain formulae whose truth value does not depend on the objects its variables refer to when interpreted in a mathematical structure.

Let us look at the following formulae in our standard language:

$$
\begin{aligned}
\varphi_1 &:= \forall x \exists y \, x \dot{\neq} y \\
\varphi_2 &:= \forall x, y \exists z ((x \leq y \wedge x \dot{\neq} y) \rightarrow (x \leq z \wedge z \leq y \wedge z \dot{\neq} x \wedge z \dot{\neq} y)) \\
\varphi_3 &:= \forall y \, x \leq y
\end{aligned}
$$

The truth of the first two formulae depends only on the structure in which we interpret them. ("Truth of a formula" and "interpretation of a formula in structure" will be defined rigourously later.) φ_1 claims that for every x there is a y different from it. This is true in every structure with two or more elements. φ_2 asserts that if x is smaller than y then there is an element in between. This statement is true in the case of the rational and the real numbers, whereas it is wrong for the natural numbers.

The truth of φ_3 cannot be decided. It claims that x is equal to or less than every y. However, since we do not know what object x refers to we cannot decide whether φ_3 is true. If we interpret φ_3 in the natural numbers and assign to x the value 0, then we obtain a true statement.

In order to discuss the difference between the three formulae let us introduce two notions concerning formulae. In a formula $\forall v \alpha$ the formula α is called the *scope* of $\forall v$. An occurrence of a variable v in a formula φ is called *bound* if $\forall v$ occurs in φ and v is in its scope. Otherwise, an occurrence is called *free*.

The difference between φ_1 and φ_2 on the one hand and φ_3 on the other is that in the first two formulae all variables occurring are bound by quantifiers and cannot be interpreted freely, whereas in the third formula the variable x is not in the scope of a quantifier, that is, its occurrence is free.

We now define "free occurrence of a variable" inductively:

Definition 2.3.1. Let v be a variable and φ a formula. Then an occurrence of v in φ is said to be *free* if one of the following holds:

1. φ is atomic,
2. φ is $\neg\alpha$ and the occurrence of v in α is free,
3. φ is $\alpha \wedge \beta$ and the occurrence of v in α or in β is free,

4. φ is $\forall u\alpha$, v is different from u and the occurrence of v in α is free.

Put $\mathrm{Fr}(\varphi) := \{v \mid v \text{ occurs freely in } \varphi\}$

For a formula of the type, e.g., $\varphi \vee \psi$ free occurrences are determined by replacing $\varphi \vee \psi$ by its definition $\neg(\neg\varphi \wedge \neg\beta)$, and analogously for the other "derived" connectives and for $\exists$.

Example 2.3.2. Let $\mathcal{L}$ be the standard language and consider

$$\varphi = \forall v_0\ (v_0 \leq 0 \rightarrow v_0 \leq v_1) \wedge \exists v_2\ (0 \leq v_2 \wedge v_2 \leq v_0 \wedge v_0 \not\doteq 0 \wedge v_2 \not\doteq v_0).$$

In φ the occurrence of v_2 is only bound, the occurrence of v_1 is free and the occurrence of v_0 both free and bound. Thus, the occurrence of a variable in a formula can be free and bound at the same time!

The notion of free and bound variables can be related to calculus when we look at $\int_0^t x^2\,dx$. The value of this integral depends on t which can therefore be considered a free variable. The variable x can be considered bound as it does not make sense to look at a specific value for x.

We are now in the position to define sentences, a type of formula to which a truth value can be assigned when interpreted in a mathematical structure.

Definition 2.3.3. A formula is called a *sentence* if it does not contain any free variables.

The set $\mathrm{Sen}(\mathcal{L}) := \{\varphi \in \mathrm{Fml}(\mathcal{L}) | \, \varphi \text{ is a sentence}\}$ consists of all formulae of a formal language $\mathcal{L}$ that are sentences.

Exercise 2.3.4.

1. Consider Definition 2.3.1 with regard to whether it captures the preceding informal definition of "free."

2. Define the set $\mathrm{Bnd}(\varphi) := \{v \mid v \text{ occurs bound in } \varphi\}$ by induction on the complexity of φ.

3. Let t be an $\mathcal{L}$-term. A string s of symbols of $\mathcal{L}$ is called a *subterm* of t if s is a term that occurs as a substring of t. Give an inductive definition of the set $\{s \in \mathrm{Tm}(\mathcal{L}) \mid s \text{ is a subterm of } t\}$.

4. Let φ be an $\mathcal{L}$-formula. A string α of symbols of $\mathcal{L}$ is called a *subformula* of φ if α is a formula and occurs as a substring of φ. Give an inductive definition of the set

$$\mathrm{Subfml}(\varphi) := \{\alpha \in \mathrm{Fml}(\mathcal{L}) \mid \alpha \text{ is a subformula of } \varphi\}.$$

We end this section with a definition in which we introduce the universal closure $\forall\varphi$ of a formula φ. $\forall\varphi$ is a sentence obtained from φ by prefixing to φ a quantifier $\forall v$ for each $v \in \mathrm{Fr}(\varphi)$, thus binding all free variables of φ.

Definition 2.3.5 (Universal closure). Let φ be a formula with $\mathrm{Fr}(\varphi) = \{v_1, \ldots, v_n\}$. Then the formula $\forall v_1 \ldots \forall v_n \varphi$, denoted by $\forall\varphi$, is called the universal closure of φ.

Obtaining $\forall\varphi$ from φ is so far merely a syntactical operation. In the chapters to follow we will see that φ and $\forall\varphi$ are very closely related.

To test your knowledge, solve the multiple choice test of this chapter.

Chapter 3

Semantics

In the preceding chapter formal languages were motivated as tools to study the process of mathematical reasoning. Mathematical reasoning in the ordinary sense is the study of the properties of a mathematical structure. In this chapter we want to define the notion of mathematical structure pertaining to a formal language. That is, we define the notion of $\mathcal{L}$-*structure* for any given first-order language $\mathcal{L}$.

An $\mathcal{L}$-structure consists of a set, called *universe* of the structure, and a specific function, a specific relation and a specific constant for each function, relation and constant symbol, respectively. This permits us to define to which object a term refers, and to which statement about objects of the structure a formula refers. To complete the semantics of $\mathcal{L}$ the validity of a formula in a structure must be fixed.

3.1 Structures

Definition 3.1.1. Let $\mathcal{L} = (\lambda, \mu, K)$ be a formal language. An $\mathcal{L}$-*structure* $\mathcal{A}$ consists of

(i) a non-empty set $A = |\mathcal{A}|$, the universe of $\mathcal{A}$,

(ii) for every $i \in I$ a $\lambda(i)$-ary relation $R_i^{\mathcal{A}}$ on A, that is, a subset $R_i^{\mathcal{A}} \subseteq A^{\lambda(i)}$,

(iii) for every $j \in J$ a $\mu(j)$-ary function $f_j^{\mathcal{A}}$ on A, that is, a map

$f_j^{\mathcal{A}} : A^{\mu(j)} \to A,$

(iv) for every $k \in K$ a fixed element (constant) $c_k^{\mathcal{A}} \in A$.

$R_i^{\mathcal{A}}$, $f_j^{\mathcal{A}}$ and $c_k^{\mathcal{A}}$ are called the *interpretations* of R_i, f_j and respectively c_k, in $\mathcal{A}$.

Example 3.1.2. Again we consider the standard language and define

$$
\begin{aligned}
|\mathcal{A}| &:= \mathbb{C} = \text{ the set of complex numbers} \\
+^{\mathcal{A}} &:= \text{addition in } \mathbb{C} \\
\cdot^{\mathcal{A}} &:= \text{multiplication in } \mathbb{C} \\
-^{\mathcal{A}} &:= \text{subtraction in } \mathbb{C} \\
\leq^{\mathcal{A}} &:= \{(x,x) \mid x \in \mathbb{C}\} \\
0^{\mathcal{A}} &:= 0 \text{ of } \mathbb{C} \\
1^{\mathcal{A}} &:= 1 \text{ of } \mathbb{C}.
\end{aligned}
$$

$\mathcal{A}$ is an $\mathcal{L}$-structure.

This example shows that for any given $\mathcal{L}$-language there is a wide range as to the choice of the universe and the interpretation of the relation, function and constant symbols. In the above example $\leq$ could be interpreted as any 2-ary relation on $\mathbb{C}$. We chose the identity since there is no ordering of the complex numbers that is compatible with addition and multiplication. The standard language thus is not the ideal choice for describing the complex numbers. Had the complex numbers been the intended interpretation the symbol $\leq$ would probably have been dropped from $\mathcal{L}$.

In the following example we define for an arbitrary formal language $\mathcal{L}$ an $\mathcal{L}$-structure called *Henkin Structure* for $\mathcal{L}$. The construction seems vacuous since it is very closely linked to the syntax of the formal language and does not convey any insight. However, this construction, as it is available for any formal language, will be the basis of the construction of a model of a consistent theory in our proof of the completeness theorem.

Example 3.1.3 (Henkin Structures). Let $\mathcal{L}$ be an arbitrary formal language. Then let $\mathcal{A}$ be the $\mathcal{L}$-structure with

(i) $A = |\mathcal{A}| := \{t \in \mathrm{Tm}(\mathcal{L}) : t \text{ does not contain any variables}\}$, i.e. A consists of terms built from constants and function symbols,

(ii) for every $i \in I$ let $R_i^{\mathcal{A}}$ be an arbitrary subset of $A^{\lambda(i)}$,

(iii) for every $j \in J$ and terms $t_1, \dots, t_{\mu(j)} \in A$ set

$$f_j^{\mathcal{A}}(t_1, \dots, t_{\mu(j)}) = f_j(t_1, \dots, t_{\mu(j)}),$$

which clearly defines a map from $A^{\mu(j)}$ to A,

(iv) for every $k \in K$ set $c_k^{\mathcal{A}} = c_k$.

$\mathcal{A}$ is called a Henkin Structure for $\mathcal{L}$.

These structures are named after *Leon Henkin* who introduced them in his proof of the completeness theorem. An $\mathcal{L}$-structure demands an interpretation of every relation symbol of $\mathcal{L}$, and, as for the relation symbol $\leq$ in Example 3.1.2, $R_i^{\mathcal{A}}$ can be any $\lambda(i)$-ary relation on the universe. As we will see in the proof of the completeness theorem, given a set $\Sigma \subset \mathrm{Fml}(\mathcal{L})$ of axioms, there is an interpretation of the relation symbols R_i which depends on Σ in a *canonical* way.

3.2 Variable assignments and validity

In Chapter 2 terms were first introduced to refer to objects in mathematical structures, then formulae to refer to statements about these structures. This was motivated intuitively without a formal definition of an $\mathcal{L}$-structure.

Now, having the formal definition of an $\mathcal{L}$-structure at our disposal we can straighten out what we mean when we say that a term refers to an object in a mathematical structure. And, having done that, we can do the same with "truth of a formula in a mathematical structure."

When we defined the notion of sentence in Section 2.3 we observed that the truth of a formula depended on how its free variables were interpreted. However, also the object a term refers to depends on the interpretation of the variables occurring in this term. Depending on the interpretation of the free variables a formula relates different objects.

The next two definitions formalize the dependence between the object a term refers to and the interpretation of the variables occurring in this term.

Definition 3.2.1. Let $\mathcal{L}$ be a formal language and $\mathcal{A}$ an $\mathcal{L}$-structure. A function $h : \mathrm{Vbl}(\mathcal{L}) \to |\mathcal{A}|$ is called a *variable assignment function into* $\mathcal{A}$.

A variable assignment function $h : \mathrm{Vbl}(\mathcal{L}) \to |\mathcal{A}|$ is not subject to any conditions, that is, it need neither be injective nor surjective. It only has to assign to every variable v a value $h(v)$ in the pertaining universe $|\mathcal{A}|$.

As we will see e.g. in Definition 3.2.6 it is sometimes useful to change the value of an assignment function in a single argument. That is, given an assignment function $h : \mathrm{Vbl}(\mathcal{L}) \to |\mathcal{A}|$ and an element $a \in |\mathcal{A}|$ we define the valuation $h\binom{v}{a}$ that agrees with h for all variables u different from v and assigns to v the value a.

Definition 3.2.2. Let $\mathcal{A}$ be an $\mathcal{L}$-structure, h an assignment function, v a variable and $a \in |\mathcal{A}|$. Then the assignment function

$$h\binom{v}{a}(u) = \begin{cases} h(u) & u \neq v \\ a & u = v \end{cases}$$

is called a *v-modification of the assignment function h*.

We are now in a position to define formally what is meant by the interpretation of a term t and the truth of a formula in an $\mathcal{L}$-structure.

Definition 3.2.3. Let $\mathcal{L}$ be a formal language, $\mathcal{A}$ an $\mathcal{L}$-structure and h a variable assignment function into $\mathcal{A}$. Then the function $\bar{h} : \mathrm{Tm}(\mathcal{L}) \to |\mathcal{A}|$, called the *term assignment function generated by* h is defined inductively by assigning to a term t

1. $h(v)$ if t is the variable v,
2. $c_k^{\mathcal{A}}$ if t is the constant c_k,
3. $f_j^{\mathcal{A}}(\bar{h}(t_1), \ldots, \bar{h}(t_{\mu(j)}))$ if t is $f_j(t_1, \ldots, t_{\mu(j)})$.

The following remark is obvious but nonetheless useful.

Remark 3.2.4. Let t be an $\mathcal{L}$-term and h, h' variable assignments such that $h(v) = h'(v)$ for all variables v that occur in t. Then $\bar{h}(t) = \bar{h}'(t)$.

In particular, if a term t does not contain any variables its value $\bar{h}(t)$ does not depend on h.

Exercise 3.2.5. Prove Remark 3.2.4.

The truth of a formula has been an issue several times already, although we have not yet defined this notion rigourously. We have seen that if a formula contains free variables, then the objects they refer to have to be determined in order to assign a truth value to the formula. In our new terminology this means that a variable assignment function has to be specified in order to assign a truth value to a formula.

Definition 3.2.6. Let $\mathcal{L}$ be a formal language, φ an $\mathcal{L}$-formula, $\mathcal{A}$ an $\mathcal{L}$-structure and h a variable assignment into $\mathcal{A}$. Then we say $\mathcal{A}$ *satisfies* φ *under the assignment* h, denoted by $\mathcal{A} \models \varphi[h]$, if and only if

1. φ is $t_1 \doteq t_2$ for $\mathcal{L}$-terms t_1, t_2 and $\bar{h}(t_1) = \bar{h}(t_2)$,
2. φ is $R_i(t_1, \ldots, t_{\lambda(i)})$ for $\mathcal{L}$-terms $t_1, \ldots, t_{\lambda(i)}$, and $(\bar{h}(t_1), \ldots, \bar{h}(t_{\lambda(i)})) \in R_i^{\mathcal{A}}$,
3. φ is $\neg\alpha$ for an $\mathcal{L}$-formula α, and $\mathcal{A} \not\models \alpha[h]$ ("not $\mathcal{A} \models \alpha[h]$"),
4. φ is $\alpha \wedge \beta$ for $\mathcal{L}$-formulae α and β, and both $\mathcal{A} \models \alpha[h]$ and $\mathcal{A} \models \beta[h]$,
5. φ is $\forall v \alpha$ for an $\mathcal{L}$-formula α and a variable v, and $\mathcal{A} \models \alpha[h\binom{v}{a}]$ for all $a \in |\mathcal{A}|$.

If Σ is a set of $\mathcal{L}$-formulae, we say $\mathcal{A}$ satisfies Σ with assignment h if $\mathcal{A} \models \sigma[h]$ for all $\sigma \in \Sigma$. Notation: $\mathcal{A} \models \Sigma[h]$

φ is said to be *satisfiable* if there is an $\mathcal{L}$-structure $\mathcal{A}$ and an assignment h such that $\mathcal{A} \models \varphi[h]$.

This definition warrants a few remarks.

Remark 3.2.7. "Satisfaction" can be viewed as ternary relation between $\mathcal{L}$-structures, $\mathcal{L}$-formulae and variable assignments. Definition 3.2.6 intends to reflect the intuitive meaning (in the meta-language) of "equal", "not", "and" and "for all" by the way satisfaction is defined for $\doteq$, $\neg$, $\wedge$ and $\forall$.

This is straightforward for the first three logical symbols, however a few words about $\forall$ may be in order:

For $\mathcal{A} \models \forall v\varphi[h]$ to hold the relation $\mathcal{A} \models \varphi[h\binom{v}{a}]$ has to be true for all $a \in |\mathcal{A}|$. Whatever assignment function h is considered *all possible* values for the variable v are to be tested.

A similar argument shows that the "derived" quantifier $\exists$ perfectly mirrors the metalinguistic "there exists" with regard to satisfaction, cf. the next exercise.

Exercise 3.2.8. Write a clause for $\exists v\varphi$ extending Definition 3.2.6.

The next lemma asserts that in this relation only the value of the assignment function h on the free variables of φ in the relation $\mathcal{A} \models \varphi[h]$ matters.

Lemma 3.2.9. *Let h_1, h_2 be variable assignments into an $\mathcal{L}$-structure $\mathcal{A}$. Show that*

1. *if $h_1(v) = h_2(v)$ for all v occurring in t, then $t^{\mathcal{A}}[h_1] = t^{\mathcal{A}}[h_2]$*
2. *if $h_1(v) = h_2(v)$ for all $v \in Fr(\varphi)$, then $\mathcal{A} \models \varphi[h_1]$ if and only if $\mathcal{A} \models \varphi[h_2]$.*
3. *if $\varphi \in Sen(\mathcal{L})$, then $\mathcal{A} \models \varphi[h_1]$ if and only if $\mathcal{A} \models \varphi[h_2]$ for all variable assignments h_1, h_2.*

Proof: Cf. Exercise 3.2.10. ■

Exercise 3.2.10. Prove Lemma 3.2.9.

Finally, the next definition states formally what we mean when we say that a formula is true in a structure. In this case the structure is called a *model* of the formula. As the notion "true" refers to the structure as a whole, the truth of a formula in a structure is independent of a particular assignment of values to the variables in the formula.

Definition 3.2.11 (Validity). Let φ be an $\mathcal{L}$-formula and $\mathcal{A}$ an $\mathcal{L}$-structure. Then

1. φ is said to be *valid* or *true in* $\mathcal{A}$, denoted by $\mathcal{A} \models \varphi$, if $\mathcal{A} \models \varphi[h]$ holds for all variable assignments $h : \mathrm{Vbl} \to |\mathcal{A}|$, in which case $\mathcal{A}$ is called a *model of* φ.

2. If $\mathcal{B} \models \varphi$ for every $\mathcal{L}$-structure $\mathcal{B}$ then φ is said to be *valid*.

3. If $\Sigma \subset \mathrm{Fml}(\mathcal{L})$ then $\mathcal{A}$ is said to be a *model* of Σ, denoted by $\mathcal{A} \models \Sigma$, if $\mathcal{A} \models \varphi$ holds for all $\varphi \in \Sigma$.

Remark 3.2.12. From this definition and Lemma 3.2.9 it follows that for an $\mathcal{L}$-sentence φ and an $\mathcal{L}$-structure $\mathcal{A}$ either φ or its negation $\neg\varphi$ is valid in $\mathcal{A}$ since variable assignments do not matter for a sentence. A sentence φ is called *false in* $\mathcal{A}$ if its negation $\neg\varphi$ is valid in $\mathcal{A}$

It will prove useful to have at our disposal, in every formal language $\mathcal{L}$, a sentence that is not valid in any $\mathcal{L}$-structure.

Definition 3.2.13. Let $\mathcal{L}$ be an arbitrary formal language. Then we set

$$\bot \quad := \quad \exists v_0(v_0 \neq v_0),$$

$\bot$ being called *falsum*. As every element of an $\mathcal{L}$-structure is equal to itself, $\bot$ is obviously false in every $\mathcal{L}$-structure.

In the comment preceding Definition 2.3.5 it was claimed that φ and its universal closure $\forall\varphi$ are closely linked. The following lemma and corollary state that this is the case with regard to validity.

Lemma 3.2.14. *Let $\mathcal{A}$ be an $\mathcal{L}$-structure and $\varphi \in Fml(\mathcal{L})$. Then the following holds:*

$$\mathcal{A} \models \varphi \qquad \textit{iff} \qquad \mathcal{A} \models \forall\varphi.$$

Proof: It suffices to show $\mathcal{A} \models \varphi$ iff $\mathcal{A} \models \forall v\,\varphi$, $v \in \mathrm{Vbl}(\mathcal{L})$ arbitrary, since iteration over the free variables of φ then yields the result.

$$\begin{array}{lll} \mathcal{A} \models \varphi & \text{iff} & \mathcal{A} \models \varphi[h] \text{ for every assignment } h. \\ & \text{iff} & \mathcal{A} \models \varphi[h\binom{v}{a}] \text{ for every assignment } h \text{ and every } a \in |\mathcal{A}| \\ & \text{iff} & \mathcal{A} \models \forall v\,\varphi[h] \text{ for every assignment } h. \\ & \text{iff} & \mathcal{A} \models \forall v\,\varphi \end{array}$$

■

Now we come to a definition whose importance will become clear in later chapters. It captures the notion of implication with regard to structures which we call *logical implication*.

Definition 3.2.15 (Logical implication). Let $\mathcal{L}$ be a formal language, Σ a set of $\mathcal{L}$-formulae and φ an $\mathcal{L}$-formula.

If for all $\mathcal{L}$-structures $\mathcal{A}$ we have that $\mathcal{A} \models \Sigma$ implies $\mathcal{A} \models \varphi$, then we say that Σ *logically implies* φ and write

$$\Sigma \Vdash \varphi.$$

The following corollary is a generalization of the preceding lemma and follows from it immediately.

Corollary 3.2.16. *Let $\Sigma \subset Fml(\mathcal{L})$ and $\varphi \in Fml(\mathcal{L})$. Then the following holds:*

$$\Sigma \Vdash \varphi \qquad \textit{iff} \qquad \Sigma \Vdash \forall\varphi.$$

Proof: The proof follows immediately from Lemma 3.2.14. ■

Solve the exercises of Section 8.2!

3.3 Propositional logic and validity

The connectives $\neg$ and $\wedge$ as well as the derived connectives $\vee$, $\rightarrow$ and $\leftrightarrow$ serve the purpose of building more complex formulae from basic ones. In the definition of $\mathcal{A} \models \varphi[h]$ (Definition 3.2.1) the intended meaning of $\neg$ ("not") and $\wedge$ ("and") is formalized. We will see later in this section that this is also true for $\vee$, $\rightarrow$ and $\leftrightarrow$.

Let now $\mathcal{L}$ be an arbitrary formal language, α an $\mathcal{L}$-formula and $\mathcal{A}$ an $\mathcal{L}$-structure. Defining $\varphi = \alpha \vee \neg\alpha$ we obtain $\mathcal{A} \models \varphi[h]$ for every assignment function h into $\mathcal{A}$:
From the definition of a variable assignment (Definition 3.2.6) and the definition of the derived connective $\vee$ in Subsection 2.2.3 it follows that $\mathcal{A} \models \varphi[h]$ holds if one of $\mathcal{A} \models \alpha[h]$ or $\mathcal{A} \models \neg\alpha[h]$ holds. If $\mathcal{A} \models \alpha[h]$ is not true then it follows from Definition 3.2.6 that $\mathcal{A} \models \neg\alpha[h]$ is true. Since $\mathcal{A} \models \varphi[h]$ for every assignment h, we have $\mathcal{A} \models \varphi$.

As the formal language $\mathcal{L}$, the $\mathcal{L}$-structure $\mathcal{A}$ and the $\mathcal{L}$-formula α were arbitrary, it is the way φ is built from α that accounts for $\mathcal{A} \models \varphi$, and not the formula α itself. φ can be obtained by substituting α for P in the expression $P \vee \neg P$. P is denoted an *atomic proposition*.

In the following we want to identify simultaneously, for all formal languages, the formulae that are valid due to the way their structure is determined by connectives. To achieve this goal we consider the expressions ϕ composed of atomic propositions $P_0, P_1, \ldots$ and the connectives $\neg$ and $\wedge$ ($\vee$, $\rightarrow$ and $\leftrightarrow$ will be introduced as abbreviations) such that the following holds: If $\mathcal{L}$ is an arbitrary formal language and φ an $\mathcal{L}$-formula that is obtained from ϕ by substituting the atomic propositions by $\mathcal{L}$-formulae then $\mathcal{A} \models \varphi$ holds for every $\mathcal{L}$-structure $\mathcal{A}$. A given atomic proposition thereby has to be substituted always by the same formula. We call such expressions Φ *propositional tautologies*.

For a systematic treatment we introduce *a language for propositional logic* in the following subsection.

3.3.1 A language for propositional logic

As for first-order languages we have to first fix an alphabet and then the strings that qualify as propositional formulae. The propositional tautologies will be a subset of the propositional formulae. Unlike for first-order languages there is only one language for propositional logic.

Definition 3.3.1. The *alphabet* of the language $\mathcal{L}_{PL}$ of propositional logic consists of

- atomic propositions: $P_0, P_1, \ldots$
- connectives: $\neg$, $\wedge$
- auxiliary symbols: parentheses

Definition 3.3.2. Let ϕ be a finite string of symbols from $\mathcal{L}_{PL}$. Then ϕ is a *propositional formula* if either

1. ϕ is an atomic proposition P_i, $i \in \mathbb{N}$,
2. ϕ is $(\neg\psi)$ for a propositional formula ψ or
3. ϕ is $(\psi_1 \wedge \psi_2)$ for propositional formualae ψ_1, ψ_2.

$\mathrm{AtP}(\phi)$ is the set of all atomic propositions that occur in ϕ.

Remark 3.3.3. We employ the conventions for formulae of first-order languages listed in Subsection 2.2.3 that can be applied to $\mathcal{L}_{PL}$. This means in particular that $\vee$, $\rightarrow$ and $\leftrightarrow$ are introduced as abbreviations.

3.3.2 Propositional valuations

The next step is to introduce a semantics for $\mathcal{L}_{PL}$ that reflects how $\mathcal{A} \models \varphi[h]$ depends on how the φ is built up from simpler formulae by connectives. By means of this semantics we will single out the propositional formulae that are propositional tautologies.

We proceed in a way similar to first-order languages. As there are no function, relation and constant symbols to interpret, only the universe has to be fixed. We content ourselves with the set consisting of the truth values *true* and *false*.

The universe consists of the set $\{\mathsf{t}, \mathsf{f}\}$, where t stands for "true" and f for "false". Propositional logic serves as a very simple model of reasoning in which dependencies of statements are investigated. For this purpose the two truth values suffice. Below we will set up a relationship between propositional formulae and first-order formulae. In this relationship the truth value "true" of a propositional formulae corresponds to "is valid" of a corresponding first-order formula.

Definition 3.3.4. A *propositional valuation* s is a function that assigns to every atomic proposition P_i, $i \in \mathbb{N}$, one of the values t or f, that is $s : \{P_i : i \in \mathbb{N}\} \rightarrow \{\mathsf{t}, \mathsf{f}\}$.

In Definition 3.2.6 we defined $\mathcal{A} \models \varphi[h]$ for a given assignment function h by induction on the complexity of φ. We do the same for a given propositional formula ϕ when a propositional valuation s is given.

Definition 3.3.5. Let the propositional valuation $s : \{P_i : i \in \mathbb{N}\} \rightarrow \{\mathsf{t}, \mathsf{f}\}$ be given. Then we obtain a function $\bar{s} : \mathrm{Fml}(\mathcal{L}_{PL}) \rightarrow \{\mathsf{t}, \mathsf{f}\}$ by the following inductive definition:

1. $\bar{s}(P_i) = \mathsf{t}$ iff $s(P_i) = \mathsf{t}$ for the variables P_i, $i = 0, 1, \ldots$
2. $\bar{s}(\neg\psi) = \mathsf{t}$ iff $\bar{s}(\psi) = \mathsf{f}$
3. $\bar{s}(\psi_1 \wedge \psi_2) = \mathsf{t}$ iff $\bar{s}(\psi_1) = \mathsf{t}$ and $\bar{s}(\psi_2) = \mathsf{t}$

If $\bar{s}(\phi) = \mathsf{t}$ we say that the valuation s *satisfies* ϕ.

As in Definition 3.2.6 the connectives $\neg$ and $\wedge$ are interpreted as "not" and "and" respectively. This is the place to justify the definitions of the "derived" connectives $\vee$, $\rightarrow$ and $\leftrightarrow$ in Subsection 2.2.3.

The case of $\vee$ — intended to model the metalinguistic "or" — is easy. We certainly want "ψ_1 or ψ_2" to be true iff at least one of ψ_1 and ψ_2 is true. Now one verifies by inspection that the propositional formula $\Phi = \neg(\neg\psi_1 \wedge \neg\psi_2)$ has this property - $\bar{s}(\Phi) = \mathsf{t}$ iff at least one of $\bar{s}(\psi_1) = \mathsf{t}$ and $\bar{s}(\psi_2) = \mathsf{t}$ holds, for any propositional valuation s. Hence, we *define* $\psi_1 \vee \psi_2$ to be an abbreviation for $\neg(\neg\psi_1 \wedge \neg\psi_2)$.

$\rightarrow$ is more difficult. How should the truth of "ψ_1 implies ψ_2" depend on the truth of ψ_1 and ψ_2, respectively?
As there are four combinations of true/false which the pair (ψ_1, ψ_2) may take, there are $2^4 = 16$ possibilities for a function assigning either "true" or "false" to each of the four combinations. We have to choose one such function, reflecting our intuitive understanding of the metalinguistic "implies". We admit that there is no completely canonical way to pick this function; however, our choice - that of the so-called "classical" propositional calculus - is widely accepted and justified by the success of this calculus in modelling (parts of) ordinary mathematics.

We proceed as follows: We assign ψ_1 and ψ_2 each one of the four possible combinations of t and f, and consider what might be desirable as truth value for each of the propositional formulae $\psi_1 \rightarrow \psi_2$, $\psi_2 \rightarrow \psi_1$ and $\neg\psi_2 \rightarrow \neg\psi_1$.

ψ_1	ψ_2	$\psi_1 \rightarrow \psi_2$	$\psi_2 \rightarrow \psi_1$	$\neg\psi_2 \rightarrow \neg\psi_1$
t	t	t 3)	t 3)	t 5)
t	f	f 1)	$x = \mathsf{t}$ 4)	f 5)
f	t	$x = \mathsf{t}$ 4)	f 2)	t 5)
f	f	$y = \mathsf{t}$ 6)	$y = \mathsf{t}$ 6)	$? = \mathsf{t}$ 6)

If ψ_1 is true, but ψ_2 is false, we certainly do not want $\psi_1 \rightarrow \psi_2$ to be true. This gives the entry f in step 1). With the roles of ψ_1 and ψ_2 reversed, we obtain the entry f in step 2). Being liberal we accept $\psi_1 \rightarrow \psi_2$ to be true if both ψ_1 and ψ_2 are true, giving two entries t in step 3). How should x and y in the first column be chosen? Reversing roles again, we see that obviously the same values must be entered in the appropriate places of the second column. Now if x were f, then the "truth-table" for $\psi_1 \to \psi_2$ and $\psi_2 \to \psi_1$ would be identical (independent from the particular choice for y!). However, this does clearly *not* reflect the intuitive meaning of implies - so we settle for 4) $x = \mathsf{t}$. It remains to determine y. We can calculate already the first three entries of the last column in step 5) based on our choices in steps 1)-4): if we want the "contraposition" to hold - $\psi_1 \to \psi_2$ has the same "truth-behaviour" as $\neg\psi_2 \to \neg\psi_1$ - then the first and third columns must be identical, and the first three entries $\psi_1 \to \psi_2$ are already determined. The fourth entry ? corresponds to $\mathsf{t} \to \mathsf{t}$ and is therefore the same as the first entry of $\psi_1 \to \psi_2$. y, the fourth entry of $\psi_1 \to \psi_2$, is thus determined too.

So we arrive at "$\psi_1 \to \psi_2$ is false exactly if ψ_1 is true and ψ_2 is false". By inspection again, we see that the propositional formula $\neg\psi_1 \vee \psi_2$ (that is, $\neg(\neg\neg\psi_1 \wedge \neg\psi_2)$) has this property:
$\overline{s}(\neg(\neg\neg\psi_1 \wedge \neg\psi_2)) = \mathsf{f}$ iff $\overline{s}(\psi_1) = \mathsf{t}$ and $\overline{s}(\psi_2) = \mathsf{f}$. Hence, we *define* $\psi_1 \to \psi_2$ to be simply an abbreviation for $\neg(\neg\neg\psi_1 \wedge \neg\psi_2)$ respectively $\neg\psi_1 \vee \psi_2$.

We leave it as an exercise to establish what the truth-behavior of $\psi_1 \leftrightarrow \psi_2$ should be and that it is indeed identical with that of $(\psi_1 \to \psi_2) \wedge (\psi_2 \to \psi_1)$.

Exercise 3.3.6. Establish the truth-behavior for the derived connective $\leftrightarrow$ such that it reflects the metalingual "equivalent".

Having done all that, we may indeed claim that $\vee$, $\to$ and $\leftrightarrow$ model "or", "implies" and "equivalent" in the metalanguage.

3.3.3 Tautologies

The reason for introducing $\mathcal{L}_{PL}$ and propositional formulae was to find the propositional tautologies. The propositional formulae that are described in the next definition are precisely these expressions.

Definition 3.3.7. A propositional formula ϕ is called a *propositional tautology* if and only if $\overline{s}(\phi) = \mathsf{t}$ for every propositional valuation $s : \{P_i : i \in \mathbb{N}\} \to \{\mathsf{t}, \mathsf{f}\}$.

Example 3.3.8. The propositional formulae $P_0 \vee \neg P_0$, $(P_0 \wedge \neg P_0) \to P_1$, $(P_0 \wedge P_1) \to P_0$ and $(P_0 \to P_1) \to ((P_1 \to P_2) \to (P_0 \to P_2))$ are examples of propositional tautologies.

Before defining the crucial notion of *first-order tautology* we have to state formally what it means to obtain a first-order formula by substituting first-order formulae for atomic propositions in a propositional formula.

Definition 3.3.9. Let $\mathcal{L}$ be a first-order language, $\alpha_1, \ldots, \alpha_n$ $\mathcal{L}$-formulae and ϕ a propositional formula with $\mathrm{AtP}(\phi) = \{P_{i_1}, \ldots, P_{i_n}\}$ for pairwise different natural numbers $i_j, j = 1, \ldots, n$. Then

$$\phi[P_{i_1}, \ldots, P_{i_n}/\alpha_1, \ldots, \alpha_n]$$

denotes the $\mathcal{L}$-formula that is obtained by replacing in ϕ every occurrence of P_{i_j} by α_j, $j = 1, \ldots, n$.

Remark 3.3.10. To show that $\phi[P_{i_1}, \ldots, P_{i_n}/\alpha_1, \ldots, \alpha_n]$ in Definition 3.3.9 is always an $\mathcal{L}$-formula is an easy induction on the complexity of ϕ.

The following lemma states that for an $\mathcal{L}$-formula $\varphi = \phi[P_{i_1}, \ldots, P_{i_n}/\alpha_1, \ldots, \alpha_n]$, an $\mathcal{L}$-structure $\mathcal{A}$ and a variable assignment h into $\mathcal{A}$ there is a propositional valuation $s_{\mathcal{A},h,\varphi}$ such that $\mathcal{A} \models \varphi[h]$ is equivalent to $\bar{s}_{\mathcal{A},h,\varphi}(\phi) = \mathsf{t}$. This means that satisfaction of such a first-order formula can be reduced to satisfaction of a propositional formula.

Lemma 3.3.11. *Let $\mathcal{L}$ be a first-order language and φ an $\mathcal{L}$-formula with $\varphi = \phi[P_{i_1}, \ldots, P_{i_n}/\alpha_1, \ldots, \alpha_n]$ for $\mathcal{L}$-formulae $\alpha_1, \ldots, \alpha_n$ and a propositional formula ϕ. Further let $\mathcal{A}$ be an $\mathcal{L}$-structure and h a variable assignment into $\mathcal{A}$.*

If we define the propositional valuation $s_{\mathcal{A},h,\varphi}$ depending on $\mathcal{A}$, h and $\varphi = \phi[P_{i_1}, \ldots, P_{i_n}/\alpha_1, \ldots, \alpha_n]$ by

$$s_{\mathcal{A},h,\varphi}(P) = \begin{cases} \mathsf{t} & P = P_{i_j} \text{ and } \mathcal{A} \models \alpha_j[h] \text{ for a } j \in \{1, \ldots, n\} \\ \mathsf{f} & \text{otherwise} \end{cases}$$

we obtain

$$\bar{s}_{\mathcal{A},h,\varphi}(\phi) = \mathsf{t} \quad \text{iff} \quad \mathcal{A} \models \varphi[h]$$

Proof: We prove the claim by induction on the complexity of ϕ. The claim is obvious if ϕ is an atomic proposition. If $\phi = \neg\psi$ or $\phi = \psi_1 \wedge \psi_2$ the claim follows from the induction hypothesis and the definition of $\mathcal{A} \models \varphi[h]$ and $\bar{s}(\phi) = \mathsf{t}$ (Definitions 3.2.6 and 3.3.5), respectively. ∎

Definition 3.3.12. Let $\mathcal{L}$ be a first-order language. An $\mathcal{L}$-formula φ is called a *first-order tautology* if there is a propositional tautology ϕ with $\mathrm{AtP}(\phi) = \{P_{i_1}, \ldots, P_{i_n}\}$ and $\mathcal{L}$-formulae $\alpha_1, \ldots, \alpha_n$ such that

$$\varphi = \phi[P_{i_1}, \ldots, P_{i_n}/\alpha_1, \ldots, \alpha_n].$$

Now we can state the main result of this section:

Theorem 3.3.13. *Let $\mathcal{L}$ be a first-order language and φ an $\mathcal{L}$-formula that is a first-order tautology. Then we have the relation*

$$\mathcal{A} \models \varphi[h]$$

for every $\mathcal{L}$-structure $\mathcal{A}$ and any assignment function h into $\mathcal{A}$. That is, the first-order tautology φ is valid in every $\mathcal{L}$-structure.

Proof: Let $\varphi = \phi[P_{i_1}, \ldots, P_{i_n}/\alpha_1, \ldots, \alpha_n]$ for a suitable propositional tautology ϕ with $\mathrm{AtP}(\phi) = \{P_{i_1}, \ldots, P_{i_n}\}$ and suitable $\mathcal{L}$-formulae $\alpha_1, \ldots, \alpha_n$. Furthermore, let $\mathcal{A}$ and $h : \mathrm{AtP}(\mathcal{L}) \to \mathcal{A}$ be an arbitrary $\mathcal{L}$-structure and a variable assignment function, respectively. We want to show that

$$\mathcal{A} \models \varphi[h]. \tag{3.1}$$

Now we define the propositional valuation $s_{\mathcal{A},h,\varphi}$ as in Lemma 3.3.11 and obtain

$$\bar{s}_{\mathcal{A},h,\varphi}(\phi) = \mathtt{t} \quad \text{iff} \quad \mathcal{A} \models \varphi[h].$$

However, since ϕ is a propositional tautology we have $\bar{s}_{\mathcal{A},h,\varphi}(\phi) = \mathtt{t}$ which implies (3.1). ■

Remark 3.3.14. There are valid first-order formulae that are not first-order tautologies. Let $\mathcal{L}$ be any language. Then the formula $\forall v\varphi \to \exists v\varphi$ is valid but not a propositional tautology.

We end this section with a remark on how one can find out if a propositional formula is a propositional tautology.

Remark 3.3.15. Let ϕ be a propositional formula, $\mathrm{AtP}(\phi) = \{P_{i_1}, \ldots, P_{i_n}\}$. Then it can be determined if ϕ is a propositional tautology by proceeding as follows: The value $\bar{s}(\phi)$ only depends on the values $s(P_{i_1}), \ldots, s(P_{i_n})$. There are 2^n valuations $s_{|\{P_{i_1},\ldots,P_{i_n}\}}$ restricted on $\{P_{i_1}, \ldots, P_{i_n}\}$. If for all of these restricted valuations $\bar{s}_{|\{P_{i_1},\ldots,P_{i_n}\}}(\phi) = \mathtt{t}$ holds, then Φ is a propositional tautology.

To test your knowledge, solve the multiple choice test of this chapter.

Solve the exercises of Section 8.3!

Chapter 4

Deductions

As was described in the introductory chapter, a proof in ordinary mathematics is an argument that convinces the mathematical community of the truth of a statement about a mathematical structure.

We intend to treat proofs themselves as mathematical objects and to investigate their properties, as is done with any mathematical object. This is done by abstracting and formalizing features common to the proofs we have encountered in ordinary mathematics. The key notions in the above description of a proof are “statement”, “mathematical structure” and “argument”. The concept of “truth” is used to single out a subset of the set of all statements about a mathematical structure.

Therefore, if we want to obtain a formalization of the notion of proof the above concepts have to be formalized. By defining formal languages and the corresponding mathematical structures we have already achieved this for statements and mathematical structures. Furthermore, the definition of the validity of a formula in a structure (Definition 3.2.11) captures “truth” in the sense described above. Therefore only the notion of “argument” remains to be formalized.

Let us consider an argument that qualifies as a mathematical proof. When proving a property of a mathematical structure one starts from “axioms” characterizing the structure. Eg., in the case of groups, these axioms would be the well-known “axioms of group theory”. One then proceeds by deriving new results from these axioms, thereby applying inference rules to previously established results or self-evident statements. Eg., having inferred from the axioms of group theory that the order of a subgroup of a finite group divides the order of the group, this result can be used to show that the order of an element of a finite group divides the order of the group. A self-evident statement about groups would for instance be that the order of any element is either finite or infinite, since this is a statement of the type $\alpha \vee \neg\alpha$. This process goes on until one arrives, after a finite number of steps, at the statement that was to be proven.

This description indicates how to formalize the notion of proof relative to a formal language $\mathcal{L}$. A formal proof in $\mathcal{L}$, henceforth called *deduction* in order to distinguish it from proofs in everyday mathematics, is a finite sequence of $\mathcal{L}$-formulae such that each member is either

1. an element of Σ, a designated set of $\mathcal{L}$-formulae describing the $\mathcal{L}$-structures about which the deduction intends to establish a statement.

2. a *logical axiom*, an $\mathcal{L}$-formula which is the formal counterpart of a self-evident statement,

or

3. obtained by applying an *inference rule* to previous elements of the sequence.

One of the crucial properties of a deduction, called *correctness*, is that any statement it establishes from a set of formulae Σ is valid in any structure provided this is the case for the members of Σ. The logical axioms and inference rules have to take this into account.

Before giving a formal definition of the notion of deduction in Section 4.4 we therefore have to specify the logical axioms for a given formal language $\mathcal{L}$. This boils down to finding collections of $\mathcal{L}$-formulae that are valid in all $\mathcal{L}$-structures which is the contents of Section 4.2.

Furthermore, we have to fix a set of rules that allow us to obtain, given a formal language $\mathcal{L}$, an $\mathcal{L}$-formula as the "conclusion" from a set of $\mathcal{L}$-formulae ("the premises"), if these $\mathcal{L}$-formulae satisfy certain syntactical conditions. These rules will be the formal counterpart to the rules applied in "normal" proofs. The conditions are such that the conclusion is true in an $\mathcal{L}$-structure if this is the case for the $\mathcal{L}$-formulae in the premise. This means that the inference rules applied in a proof are *correct*. The set of inference rules will be presented in Section 4.3.

In order to make this precise we have to define the syntactical operation of substituting a term for a variable in a formula. This is done in Section 4.1.

4.1 Substitution

If a formula $\forall v\, \alpha$, v occurring free in α, is true in a structure, then α is true in this structure independent on how v is interpreted. The formula $\varphi_1 = \forall x \exists y\, (x \not\doteq y)$ is, as we have seen in the preceding chapter, true in every structure with at least two elements. Whatever value is assigned to x a different value can be assigned to y making the formula $\exists y\, x \not\doteq y$ true.

The preceding paragraph suggests that the following inference rule is correct: The premise $\forall v\, \alpha$, v occurring free in α, admits all formulae as conclusions that are obtained from $\forall v\, \alpha$ by dropping $\forall v$ and substituting an arbitrary term for v in α. As we have seen above, any value can be assigned to v in α.

This rule also reflects the intended meaning "for all" of the quantifier $\forall$: any value in the universe of the mathematical structure can be assigned to the quantified variable. Therefore, the variable can be substituted by any term.

If there are no free occurrences of v in α, the quantifier $\forall v$ in $\forall v\, \alpha$ can be dropped and the substitution of v in α need not be considered. This is reflected by Clause 5 in Definition 4.1.2). This definition is preceded by another definition, namely the substitution of a term *in a term*.

Definition 4.1.1. Let s and t be $\mathcal{L}$-terms and v be a variable. Then $s(v/t)$ ("s with v replaced by t") is inductively defined as follows:

1. If s is v then $s(v/t)$ is t.
2. If s is a variable not equal to v then $s(v/t)$ is s.
3. If s is a constant c_k for a $k \in K$ then $s(v/t)$ is s.
4. If s is $f_j(t_1, \ldots, t_{\mu(j)})$ for $\mathcal{L}$-terms $t_1, \ldots, t_{\mu(j)}$ then $s(v/t)$ is $f_j(t_1(v/t), \ldots, t_{\mu(j)}(v/t))$.

Thus s with v replaced by t is obtained by replacing every occurrence of v in s by t.

Definition 4.1.2. Let φ be an $\mathcal{L}$-formula, t an $\mathcal{L}$-term and v a variable. The $\mathcal{L}$-formula $\varphi(v/t)$ ("φ with v replaced by t") is defined inductively as follows:

1. If φ is $t_1 \doteq t_2$ for $\mathcal{L}$-terms t_1, t_2 then $\varphi(v/t)$ is $t_1(v/t) \doteq t_2(v/t)$.
2. If φ is $R_i(t_1, \ldots, t_{\lambda(i)})$ is for $\mathcal{L}$-terms $t_1, \ldots, t_{\lambda(i)}$ then $\varphi(v/t)$ is $R_i(t_1(v/t), \ldots, t_{\lambda(i)}(v/t))$.
3. If φ is $\neg\alpha$ for an $\mathcal{L}$-formula α then $\varphi(v/t)$ is $\neg\alpha(v/t)$.
4. If φ is $\alpha \wedge \beta$ for $\mathcal{L}$-formulae α and β then $\varphi(v/t)$ is $\alpha(v/t) \wedge \beta(v/t)$.
5. If φ is $\forall u\alpha$ for an $\mathcal{L}$-formula α and a variable u then

$$\varphi(v/t) = \begin{cases} \forall u\alpha(v/t) & v \neq u \\ \varphi & v = u. \end{cases}$$

Example 4.1.3. Let $\mathcal{L}$ be the standard language and

$$\varphi = \forall u\ (u \leq v + 1)$$

Then we obtain

$$\begin{array}{rcll} \varphi(v/0) & = & \forall u\ (u \leq 0 + 1) & (v \text{ substituted by } 0) \\ \varphi(v/v+1) & = & \forall u\ (u \leq (v+1) + 1) & (v \text{ substituted by } v+1) \\ \varphi(v/v) & = & \varphi & \\ \varphi(u/v) & = & \varphi & (\text{Clause 5 in 4.1.2}) \\ \varphi(v/u) & = & \forall u\ (u \leq u + 1) & (v \text{ substituted by } u) \end{array}$$

The following exercise asks to show that substituting variables which are not free has no effect.

Exercise 4.1.4. Prove by induction on the complexity of φ that $\varphi(v/t) = \varphi$ if $v \notin \mathrm{Fr}(\varphi)$.

Written in the notation of Definition 4.1.2 the rule discussed in the introduction of this section would permit to derive $\varphi(v/t)$ from $\forall v\varphi$ for any term

t. That some caution with regard to the choice of the term t is necessary shall be shown now.

Let us consider the formula $\varphi = \exists y(x \neq y)$. Then the proposed rule would imply $\varphi(x/t)$ from $\forall x\varphi$ for any term t. Let first $t := u$, u a variable different from y. $\varphi(x/t)$, the formula we obtain when x in φ is substituted by t, is $\exists y(u \neq y)$. Whenever a structure has at least two elements, whatever value is assigned to the free variable u, there is always a different value which can be assigned to y making $\exists y(u \neq y)$ true. Thus, whenever φ is true, so is $\exists y(u \neq y)$.

However, if we now substitute the term $t = y$ for the variable x in φ, we obtain $\varphi(x/t) = \exists y(y \neq y)$, which is a formula that is false in every structure, independent of the number of elements of this structure.

What went wrong? The variable y substituted for x in φ came in the scope of the $\exists y$ within φ. This situation is denoted by "y is not free for x in φ".

More generally, *a term t is free for a variable v in a formula φ* if there is no variable u in t and no quantifier $\forall u$ in φ such that a free occurrence of v is in the scope of $\forall u$.
We now give a formal definition of this important notion:

Definition 4.1.5. Let φ be an $\mathcal{L}$-formula, t a term and v a variable. Then *t is free for v in φ* if

1. φ is atomic, e.g. satisfies clause 1 or 2 of Definition 2.2.5.
2. φ is $\neg\alpha$ and t is free for v in α.
3. φ is $\alpha \wedge \beta$ and t is free for v in both α and β.
4. φ is $\forall x\alpha$ and either
   (a) v is not free in α, or
   (b) x does not occur in t and t is free for v in α.

Example 4.1.6. Let $\mathcal{L}$ be the standard language. Then

- u is not free for v in $\exists u(u \neq v)$
- u is not free for v in $\varphi = \forall u(u \leq v + 1)$, but $t = v + 1$ is free for v in φ.

Solve the exercises of Section 8.4!

4.2 Logical axioms

As stated in the introduction of this chapter, the necessary condition for a formula to be a logical axiom is validity. This means that an $\mathcal{L}$-formula φ must be valid in all $\mathcal{L}$-structures if it is to be a logical axiom. A deduction thus can use a logical axiom in the premise of any inference rule without running the risk of producing an invalid conclusion provided the inference rule itself is correct.

As a logical axiom is valid in every $\mathcal{L}$-structure, the interpretations of the nonlogical symbols in a particular $\mathcal{L}$-structure do not matter. The validity of a logical axiom in any structure results exclusively from the way the axiom is built up using $\neg$, $\wedge$, $\forall$ and $\doteq$.

There are three kinds of logical axioms: *first-order tautologies*, *identity axioms* and *quantifier axioms*. In the case of first-order tautologies the logical symbols that account for the validity are the connectives, whereas in the case of the identity and the quantifier axioms it is the equality symbol $\doteq$ and $\forall$, respectively.

4.2.1 First-order tautologies

First-order tautologies were defined in Subsection 3.3.3. According to Theorem 3.3.13 first-order tautologies are valid.

Because we have already treated first-order tautologies in Subsection 3.3.3, here we content ourselves with considering an example.

Example 4.2.1. Let $\mathcal{L}$ be the standard language and φ the formula

$$(((0 < x \wedge 0 < y) \rightarrow 0 < x + y) \wedge (0 < x + y \rightarrow (0 < x \vee 0 < y)))$$
$$\rightarrow ((0 < x \wedge 0 < y) \rightarrow (0 < x \vee 0 < y))$$

$$\begin{aligned}
\alpha_0 &= 0 < x \wedge 0 < y,\\
\alpha_1 &= 0 < x + y,\\
\alpha_2 &= 0 < x \vee 0 < y,\\
\phi &= (P_0 \to P_1) \wedge (P_1 \to P_2) \to (P_0 \to P_2),
\end{aligned}$$

we obtain

$$\varphi = \phi[P_0, P_1, P_2/\alpha_0, \alpha_1, \alpha_2].$$

As ϕ is a propositional tautology in $\mathcal{L}_{PL}$, φ is a first-order tautology.

4.2.2 Identity axioms

As stated in Remark 3.2.7 it follows from Definition 3.2.6 that the symbol $\doteq$ is interpreted as equality of the objects the terms refer to. This implies that the axioms introduced below are valid.

Let $\mathcal{L}$ be an arbitrary formal language. The identity axioms are then the following collections of formulae:

(I1) For all variables v

$$v \doteq v$$

(I2) For all variables v_1, v_2, v_3

$$v_1 \doteq v_2 \to (v_1 \doteq v_3 \to v_2 \doteq v_3)$$

(I3) For all variables $v_1, \dots, v_{\lambda(i)}$, $w_1, \dots, w_{\lambda(i)}$ and all $i \in I$

$$(v_1 \doteq w_1 \wedge \ldots \wedge v_{\lambda(i)} \doteq w_{\lambda(i)}) \to (R_i(v_1, \dots, v_{\lambda(i)}) \to R_i(w_1, \dots, w_{\lambda(i)}))$$

(I4) For all variables $v_1, \dots, v_{\mu(j)}$, $w_1, \dots, w_{\mu(j)}$ and all $j \in J$

$$(v_1 \doteq w_1 \wedge \ldots \wedge v_{\mu(j)} \doteq w_{\mu(j)}) \to f_j(v_1, \dots, v_{\mu(j)}) \doteq f_j(w_1, \dots, w_{\mu(j)})$$

The identity axioms (I1) and (I2) are the syntactic counterpart to the equality relation. (I3) and (I4) express that variables referring to the same object can be substituted for each other in atomic formulae.

Each I_i is actually a collection of formulae because the variables and function relation symbols are left unspecified. Often we say axiom (I3) when we really mean one element (an "instance") of the collection (I3). This is also our practice for the other axioms, where "axiom" can mean either a collection of formulae or a specific instance of this collection.

To see that (I4) is valid let us assume that

$$\mathcal{A} \models v_1 \doteq w_1 \wedge \ldots \wedge v_{\mu(j)} \doteq w_{\mu(j)} \ [h],$$

$\mathcal{A}$ an arbitrary $\mathcal{L}$-structure, h an arbitrary variable assignment into $\mathcal{A}$. Then $h(v_1) = h(w_1), \ldots, h(v_{\mu(j)}) = h(w_{\mu(j)})$ which implies

$$\bar{h}(f_j(v_1, \ldots, v_{\mu(j)})) = \bar{h}(f_j(w_1, \ldots, w_{\mu(j)})),$$

thus

$$\mathcal{A} \models f_j(v_1, \ldots, v_{\mu(j)}) = f_j(w_1, \ldots, w_{\mu(j)}) \ [h],$$

which proves the claim.

Exercise 4.2.2. Show that the axioms (I1), (I2) and (I3) are valid.

4.2.3 Quantifier axioms

Let $\mathcal{L}$ be a formal language. Then the quantifier axioms are the $\mathcal{L}$-formulae of the following type:

(Q) All formulae φ such that

$$\varphi = \forall v\, \alpha \to \alpha(v/t)$$

for any formula α and any term t that is free for v in α.

To see that t has to be free for v in α for φ to be valid, consider

$$\alpha = \exists u (v \neq u).$$

The term $t = u$ is not free for v in α. The formula $\forall v\, \alpha \to \alpha(v/t)$ is then the formula

$$\forall v\, \exists u (v \neq u) \to \exists u (u \neq u),$$

which is not valid. The proof that all formulae in (Q) are valid is more complicated than the proof of the validity of the identity axioms. It is based on two technical lemmata and will be presented in Section 5.1.

4.3 Inference rules

Again, we fix a formal language $\mathcal{L}$ and assume that all formulae and structures refer to this language. An inference rule enables the derivation of a new formula φ from a finite set Γ of previously established formulae. The elements of Γ are called the *premises* of the rule (the whole set Γ is called the *premise*), φ is called the *conclusion* of the rule. This situation is denoted by $\langle\Gamma, \varphi\rangle$ or $\frac{\Gamma}{\varphi}$.

As stated in the introduction the inference rules have to be correct. This means that Γ and φ must satisfy certain syntactical conditions guaranteeing that $\mathcal{A} \models \Gamma$ implies $\mathcal{A} \models \varphi$.

As e.g. (I_1) is a collection of formulae an inference rule is a collection of pairs $\langle\Gamma, \varphi\rangle$. The inference rule applies to a given pair $\langle\Gamma, \varphi\rangle$ if it is an element of the inference rule. An element of an inference rule is called an instance of this rule.

There are two inference rules in our deductive system. The first is called *modus ponens* and is denoted by (MP). The premise of an instance of (MP) consists of two formulae. Therefore, this inference rule can be considered a subset of $(\mathrm{Fml}(\mathcal{L}))^2 \times \mathrm{Fml}(\mathcal{L})$. The second inference rule is called *generalization* and is denoted by $(\forall)$. The premise of its instances consists of a single formula, which means $(\forall) \subset \mathrm{Fml}(\mathcal{L}) \times \mathrm{Fml}(\mathcal{L})$.

1. **Modus Ponens**

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

Applying (MP) in a deduction means that if the deduction has already produced φ and $\varphi \to \psi$ then the deduction can be extended by ψ. Since $\varphi \to \psi$ stands for "φ implies ψ" it is natural to infer from φ, which is the premise of $\varphi \to \psi$, and $\varphi \to \psi$ the conclusion ψ of $\varphi \to \psi$. That the interpretation of the symbol $\to$ in an $\mathcal{L}$-structure corresponds to "implies" yields the correctness of (MP):

Let $\mathcal{A}$ be an $\mathcal{L}$-structure and h a variable assignment into $\mathcal{A}$ such that the premise $\{\varphi, \varphi \to \psi\}$ of an instance of (MP) is valid in $\mathcal{A}$ under h. That is

$$\mathcal{A} \models \varphi[h] \tag{4.1}$$

and

$$\mathcal{A} \models (\varphi \to \psi)[h]. \tag{4.2}$$

(4.2) signifies, cf. Subsection 3.3.2, that $\mathcal{A} \models \varphi[h]$ implies $\mathcal{A} \models \psi[h]$ from which with (4.1) we obtain $\mathcal{A} \models \psi[h]$.

This shows that the inference rule (MP) is correct.

2. **Generalization**

$$\frac{\varphi \to \psi}{\varphi \to \forall v \psi} \qquad \text{if } v \notin \mathrm{Fr}(\varphi) \tag{$\forall$}$$

The application of ($\forall$) in a deduction means that a deduction resulting in $\varphi \to \psi$ may be extended by $\varphi \to \forall v \psi$ provided v is not free in φ.

According to Lemma 3.2.14 $\forall \psi$ is valid if and only if ψ is valid. Thus, a natural rule would be to extend a deduction of ψ by the formula $\forall v \psi$, thereby preserving validity. We will see in Section 5.3, by appropriately choosing φ, that the corresponding inference rule

$$\frac{\psi}{\forall v \psi} \tag{res$\forall$}$$

follows from ($\forall$). Thus, although there is a restriction in the choice of φ in ($\forall$), the inference rule ($\forall$) is more general than (res$\forall$).

To see that the condition $v \notin \mathrm{Fr}(\varphi)$ is necessary let $\mathcal{L}$ be the standard language and consider the $\mathcal{L}$-formulae $\varphi = 0 < v$ and $\psi = 0 < v + v$. Then the formula $\varphi \to \psi$ is valid in the real numbers but not $\varphi \to \forall v \psi$.

We end this section by showing that ($\forall$) is correct.

Lemma 4.3.1. *The inference rule ($\forall$) is correct.*

Proof: We have to show, for an arbitrary structure $\mathcal{A}$, that $\mathcal{A} \models \varphi \to \psi$ implies $\mathcal{A} \models \varphi \to \forall v \psi$ for $v \notin \mathrm{Fr}(\varphi)$.

So let us assume $\mathcal{A} \models \varphi \to \psi$ and pick an arbitrary valuation h into $|\mathcal{A}|$. We must therefore show $\mathcal{A} \models (\varphi \to \forall v \psi)[h]$, that is,

$$\mathcal{A} \models \varphi[h] \qquad \text{implies} \qquad \mathcal{A} \models \forall v \psi[h]. \tag{4.3}$$

So assume

$$\mathcal{A} \models \varphi[h]. \tag{4.4}$$

Pick $a \in |\mathcal{A}|$. Since $\mathcal{A} \models \varphi \to \psi$ we have $\mathcal{A} \models \varphi \to \psi[h\binom{v}{a}]$, that is,

$$\mathcal{A} \models \varphi[h\binom{v}{a}] \qquad \text{implies} \qquad \mathcal{A} \models \psi[h\binom{v}{a}]. \tag{4.5}$$

Since $v \notin \mathrm{Fr}(\varphi)$ we have the equivalence $\mathcal{A} \models \varphi[h] \Leftrightarrow \mathcal{A} \models \varphi[h\binom{v}{a}]$, from which we obtain with the assumption (4.4)

$$\mathcal{A} \models \varphi[h\binom{v}{a}]. \tag{4.6}$$

From (4.5) and (4.6) we obtain

$$\mathcal{A} \models \psi[h\binom{v}{a}].$$

This holds for any $a \in |\mathcal{A}|$, therefore

$$\mathcal{A} \models \forall v \psi[h].$$

■

4.4 Definition of the notion of deduction

Having formalized the ingredients of a deduction, we can now give its exact definition.

Definition 4.4.1. Let $\mathcal{L}$ be a formal language, $\Sigma \subseteq \mathrm{Fml}(\mathcal{L})$ a collection of $\mathcal{L}$-formulae and D a finite sequence $(\varphi_1, \varphi_2, \ldots, \varphi_n)$ of $\mathcal{L}$-formulae. D is said to be a *deduction from* Σ if for each i, $1 \leq i \leq n$, at least one of the following clauses holds.

1. φ_i is a logical axiom,
2. $\varphi_i \in \Sigma$
3. There is an inference rule $\langle \Gamma, \tau \rangle$ such that $\Gamma \subseteq \{\varphi_1, \ldots, \varphi_{i-1}\}$ and $\varphi_i = \tau$.

If for an $\mathcal{L}$-formula φ there is a deduction $D = (\varphi_1, \varphi_2, \ldots, \varphi_n)$ from Σ with $\varphi = \varphi_n$ then D is called a *deduction of* φ *from* Σ which is denoted by $\Sigma \vdash \varphi$. We write $\Sigma \nvdash \varphi$ if there is no deduction of φ from Σ.

At this point it seems appropriate to call attention to a few issues concerning formal languages and the meta language.

With the introduction of the notion of deduction we have extended the mathematical objects we study in this course. This is an essential extension as treatment of the notion of proof as a mathematical object necessitated the concept of a formal language.

In the following we will have to differentiate between "proof" and "deduction." The first denotes a proof in the customary sense. It is an argument on the meta level concerning the mathematical objects of this course, among which we find deductions as introduced in Definition 4.4.1. In the Deduction Theorem (cf. Theorem 5.5.1) the statement on the meta level concerns a property of deductions. The proof is an ordinary argument in the semiformal language on the meta level.

The following exercise is useful for becoming familiar with the definition of a formal deduction. The statement which we are about to prove will often be used in the following, mostly without explicit mention.

Exercise 4.4.2 (Concatenated deductions). Let $D_1 = (\psi_1, \ldots, \psi_m)$ and $D_2 = (\varphi_1, \ldots, \varphi_n)$ be deductions from a set Σ. Show that

$$(D_1, D_2) := (\psi_1, \ldots, \psi_m, \varphi_1, \ldots, \varphi_n)$$

is also a deduction from Σ. (Observe that in (D_1, D_2) the order of D_1 and D_2 is preserved.). This means that two deductions can be concatenated.

As stated above, one of the crucial properties of a deduction is *correctness*. For the inference rules and the logical axioms except (Q) this has either shown or put as an exercise. We will complete the proof of the correctness of our deductive system in Section 5.1.

Remark 4.4.3. The notation introduced in Definition 3.2.15 allows us to express neatly the correctness of a deductive system:

$$\Sigma \vdash \varphi \qquad \text{implies} \qquad \Sigma \Vdash \varphi$$

(see Theorem 5.1.4).

Another central property is *completeness* which will be considered extensively in Chapter 6. There we will show that the chosen deductive system fulfils the requirement of completeness.

A third essential property of a deductive system is *decidability* which will be discussed in Section 5.2.

There are more than just one unique deductive system with these three properties. What system one choses is mainly a matter of taste. As has already been mentioned, formal languages and deductive systems are not introduced to formalize concrete mathematical structures, but to show general properties like correctness, completeness and decidability of formalizations and the principal limitations of deductive systems. At the same time, deductive systems should be intuitive and easy to handle. In the following exercise you are asked to show the equivalence of the chosen system to an alternative one.

Exercise 4.4.4. Show that we obtain an equivalent deductive system if we replace the inference rule ($\forall$) in our deductive system by (res$\forall$) and add as a further logical axiom the following collection of formulae:

(Q2) All formulae φ such that

$$\varphi = \forall v\,(\alpha \to \beta) \to (\alpha \to \forall v \beta)$$

for formulae α and β with v not free in α.

The set of $\mathcal{L}$-formulae Σ that occurs in the relation $\Sigma \vdash \varphi$ is intended to characterize a class of $\mathcal{L}$-structures, as for example the axioms of groups characterize the class of groups. In an $\mathcal{L}$-structure there is no formula φ such that both φ and its negation $\neg\varphi$ are valid in it. If Σ is to characterize a nonempty class of $\mathcal{L}$-structures and the deductive system is correct, then there should be no formula φ such that simultaneously $\Sigma \vdash \varphi$ and $\Sigma \vdash \neg\varphi$ hold. A set Σ of formulae with this property is called *consistent*.

Definition 4.4.5 (Consistency). A set of $\mathcal{L}$-formulae is *consistent* if there is no deduction of an $\mathcal{L}$-formula φ and its negation $\neg\varphi$ from Σ at the same time. Otherwise, Σ is called *inconsistent*.

The following lemma states that from an inconsistent set of formulae every formula can be deduced. Its proof is a first example of working in a deductive system.

Lemma 4.4.6. *Let Σ be an inconsistent set of $\mathcal{L}$-formulae. Then $\Sigma \vdash \varphi$ holds for every $\mathcal{L}$-formula φ.*

Proof: Let φ be an arbitrary formula. Since Σ is inconsistent there is a formula α such that there are deductions $D_1 = (\psi_1, \ldots, \psi_m)$ and $D_2 = (\varphi_1, \ldots, \varphi_n)$ with $\psi_m = \alpha$ and $\varphi_n = \neg\alpha$. Then the sequence

$$(D_1, D_2, \alpha \to (\neg\alpha \to \varphi), \neg\alpha \to \varphi, \varphi)$$

is a deduction of φ from Σ:

From Exercise 4.4.2 it follows that (D_1, D_2) is a deduction from Σ. This deduction can be extended first by $\alpha \to (\neg\alpha \to \varphi)$ since this is a first-order tautology. The deduction $(D_1, D_2, \alpha \to (\neg\alpha \to \varphi))$ can then be extended by $\neg\alpha \to \varphi$ and φ as the result of applying the instances $\langle\{\alpha, \alpha \to (\neg\alpha \to \varphi)\}, \neg\alpha \to \varphi\rangle$ and $\langle\{\neg\alpha, \neg\alpha \to \varphi\}, \varphi\rangle$ of (MP) since α and $\neg\alpha$ occur in $(D_1, D_2, \alpha \to (\neg\alpha \to \varphi))$.

The claim is proved as φ is the last formula of $(D_1, D_2, \alpha \to (\neg\alpha \to \varphi), \neg\alpha \to \varphi, \varphi)$. ■

Remark 4.4.7. Since a deduction is a finite sequence of formulae it follows from $\Sigma \vdash \varphi$ that there is a finite subset Σ' of Σ with $\Sigma' \vdash \varphi$. If $(\varphi_1, \ldots, \varphi_n)$ is a deduction of φ from Σ consider $\Sigma' := \Sigma \cap \{\varphi_1, \ldots, \varphi_n\}$, the elements of Σ that actually occur in the deduction.

Although this observation seems innocent, its consequences are far-reaching. A first consequence is the next lemma.

Lemma 4.4.8. *A set of $\mathcal{L}$-formulae Σ is consistent if and only if every finite subset of Σ is consistent.*

Proof: If Σ is consistent, then this is obviously the case for every finite subset of Σ.

To show the other direction, we assume that Σ is inconsistent. Then we can find a formula φ and deductions $(\psi_1, \ldots, \psi_m)$ and $(\varphi_1, \ldots, \varphi_n)$ from Σ of φ and $\neg\varphi$ respectively. These deductions are from the finite set $\Sigma' := \Sigma \cap \{\psi_1, \ldots, \psi_m, \varphi_1, \ldots, \varphi_n\}$. This means that the finite set Σ' is inconsistent. ■

In Definition 3.2.13 we introduced the formula $\perp$ which is not valid in any $\mathcal{L}$-structure for a given formal language $\mathcal{L}$. Thus, if a set Σ is to characterize a nonempty class of $\mathcal{L}$-structures, then $\Sigma \nvdash \perp$ if the deductive system is correct. This observation gives rise to the following exercise.

Exercise 4.4.9. Show that a set of $\mathcal{L}$-formulae Σ is consistent if and only if $\Sigma \nvdash \perp$.

To test your knowledge, solve the multiple choice test of this chapter.

Solve the exercises of Section 8.5!

Chapter 5

Properties of our deductive system

In Section 4.4 of the preceding chapter the notion of deduction has been defined and some initial properties of it have been shown.

As already mentioned, deductions are the formal counterparts of the proofs in ordinary mathematics. Our goal is not to formalize all of existing mathematics with the pertaining proofs, but to execute as much of Hilbert's program as possible, as described in Chapter 1.

The concept of a formal language and its pertaining structures is suitable for the formalization of traditional mathematics since traditional mathematics can be described using set theory. Therefore, a formalization of set theory, which can be performed with the formal language defined in Example 2.1.4, satisfies (S1) of Hilbert's programme, described in Chapter 1.

For a deductive system to fulfil (S2) of Hilbert's program it has to possess the three properties mentioned in Section 4.4, namely correctness, completeness and decidability. Once we have established them we know that the way we have defined formal languages and deductions is suitable for the study of proofs as mathematical objects.

As mentioned in Chapter 1, Gödel's endeavour to accomplish Hilbert's program led to his famous incompleteness theorems. They state that (S3) and (S4) cannot be realized.

This is due to the fundamental limitations of the set $\{\varphi \in \mathrm{Fml} | \Sigma \vdash$

$\varphi\}$ given that Σ is decidable and has some natural properties that follow from being the axiomatization of mathematical systems. Decidability is the subject matter of Section 5.2.

That correctness is a mandatory requirement for deductions has been discussed extensively in Chapter 4. Parts of correctness (for a concise description of this notion cf. Remark 4.4.3) have been verified with the introduction of the logical axioms and inference rules. Completing this task is the topic of Section 5.1.

Correctness states that everything that can be deduced from a set of formulae that are valid in a given structure is also valid in this structure. Thus, our notion of deduction is *not too strong* as formulae that should not be deduced are not deducible.

Completeness is in a certain sense the complement of correctness. That means that our notion of deduction is not *too weak*. We would like that everything that should be deducible can be deduced. $\Sigma \Vdash \varphi$ means that φ is valid in every model of Σ. In this case one would expect that there is a proof of φ from Σ, which in our formal system amounts to the existence of a deduction of φ from Σ. That is $\Sigma \Vdash \varphi$ implies $\Sigma \vdash \varphi$ which together with Remark 4.4.3 means that $\Vdash$ and $\vdash$ are equivalent. Since the proof of completeness is intricate all of Chapter 6 is devoted to it.

As has been mentioned it is not the goal of the introduction of a formal system to formalize all of existing mathematics. What can be realized of Hilbert's program shows that in principle every ordinary proof *could be* formalized. In Section 5.6 we will interrupt our general investigation of the properties of our deductive system and carry out the formalization of an ordinary proof in detail. This is a cumbersome process even when using the derived inference rules of Section 5.3 which abbreviate frequently occurring short deductions as new inference rules.

Not only Section 5.3 but also 5.4 and 5.5 consist of derived results which will be used mainly to establish completeness.

5.1 Correctness

In this section we fill the remaining gaps of the proof of the correctness of deductions. The main endeavor consists of proving two technical lemmata.

The first lemma states that the value assigned by a variable assignment h to a term s in which a variable v was substituted by a term t can be obtained by evaluating the unchanged term s under a variable assignment h' that is the result of appropriately modifying the value of v under h.

Lemma 5.1.1. *Let $\mathcal{L}$ be a formal language, s and t $\mathcal{L}$-terms. Furthermore, let v be a variable, h a variable assignment into an $\mathcal{L}$-structure $\mathcal{A}$. Then*

$$\bar{h}(s(v/t)) \quad = \quad \bar{h}\binom{v}{\bar{h}(t)}(s)$$

Proof: To simplify notation we write h' for $h\binom{v}{\bar{h}(t)}$. We proceed by induction on the complexity of the term s.

If s is the variable v then

$$\begin{aligned} \bar{h}(s(v/t)) &= \bar{h}(v(v/t)) \\ &= \bar{h}(t) \\ &= \bar{h}'(v) \\ &= \bar{h}'(s) \end{aligned}$$

If s is a variable w different from v then

$$\begin{aligned} \bar{h}(s(v/t)) &= \bar{h}(w(v/t)) \\ &= \bar{h}(w) \\ &= h(w) \\ &= h'(w) \\ &= \bar{h}'(s). \end{aligned}$$

If s is a constant symbol c, then

$$\bar{h}(s(v/t)) = \bar{h}(c(v/t)) = \bar{h}(c) = c^{\mathcal{A}} = \bar{h}'(c) = \bar{h}'(s).$$

If s is $f_j(u_1, \dots, u_{\mu(j)})$ for a function symbol f_j and the terms $u_1, \dots, u_{\mu(j)}$ then

$$\begin{aligned}
\bar{h}(s(v/t)) &= \bar{h}(f_j(u_1, \dots, u_{\mu(j)})(v/t)) \\
&= \bar{h}(f_j(u_1(v/t), \dots, u_{\mu(j)}(v/t))) && \text{(Definition 4.1.1)} \\
&= f_j^{\mathcal{A}}(\bar{h}(u_1(v/t)), \dots, \bar{h}(u_{\mu(j)}(v/t))) && \text{(Definition 3.2.3)} \\
&= f_j^{\mathcal{A}}(\bar{h'}(u_1), \dots, \bar{h'}(u_{\mu(j)}) && \text{(inductive hypothesis)} \\
&= \bar{h'}(f_j(u_1, \dots, u_{\mu(j)})) && \text{(Definition 3.2.3)} \\
&= \bar{h'}(s).
\end{aligned}$$

■

The second lemma is the corresponding statement for formulae: a formula φ in which a term t is substituted for a variable v is valid under a variable assignment h if and only if φ is valid under the variable assignment h' that is obtained by appropriately modifying the value of v under h. This holds under the condition that t is free for v in φ.

Lemma 5.1.2. *Let $\mathcal{L}$ be a formal language, φ an $\mathcal{L}$-formula and v a variable. Furthermore, suppose that t is an $\mathcal{L}$-term that is free for v in φ and h a variable assignment into an $\mathcal{L}$-structure $\mathcal{A}$. Then*

$$\mathcal{A} \models \varphi(v/t)[h] \iff \mathcal{A} \models \varphi[h\binom{v}{\bar{h}(t)}]$$

Proof: To simplify notation we introduce the variable assignment $h' = h\binom{v}{\bar{h}(t)}$. We proceed by induction on the complexity of φ.

The inductive argument starts with the atomic formulae. First, φ be $u_1 \doteq u_2$ for terms u_1, u_2. Then we obtain the following sequences of equivalences:

$$\begin{aligned}
\mathcal{A} \models \varphi(v/t)[h] &\iff \mathcal{A} \models u_1(v/t) \doteq u_2(v/t)[h] \\
&\iff \bar{h}(u_1(v/t)) = \bar{h}(u_2(v/t)) && \text{(Definition 3.2.6)} \\
&\iff \bar{h'}(u_1) = \bar{h'}(u_2) && \text{(Lemma 5.1.1)} \\
&\iff \mathcal{A} \models \varphi[h']
\end{aligned}$$

The case where φ is $R_i(u_1, \dots, u_{\lambda(i)})$ is shown similarly.

The inductive cases involving the connectives $\neg$ and $\wedge$ follow immediately from the induction hypothesis.

Therefore, the case where φ is $\forall x\alpha$ remains. We have to distinguish the subcases $x = v$ and $x \neq v$.

1. $x = v$. This implies $\varphi(v/t) = \varphi$ according to Definition 4.1.2. From this it follows that $\mathcal{A} \models \varphi(v/t)[h]$ is equivalent to $\mathcal{A} \models \varphi[h]$. However, the latter is, according to Lemma 3.2.9, equivalent to $\mathcal{A} \models \varphi[h']$ since h and h' agree on the free variables of φ (v is not free in φ.)

2. $x \neq v$. In this subcase we distinguish two sub–subcases, (a) $v \notin \mathrm{Fr}(\alpha)$ and (b) $v \in \mathrm{Fr}(\alpha)$.

   (a) $v \notin \mathrm{Fr}(\alpha)$. From Exercise 4.1.4 we know that $\alpha(v/t) = \alpha$, and thus $\varphi(v/t) = \varphi$. As in the first subcase above, we obtain from this the equivalence of $\mathcal{A} \models \varphi(v/t)[h]$ and $\mathcal{A} \models \varphi[h']$, since again h and h' agree on the free variables of φ.

   (b) $v \in \mathrm{Fr}(\alpha)$. Because t is free for v in φ, we obtain from Definitions 2.3.1 and 4.1.5 that

   (1) x does not occur in t, and

   (2) t is free for v in α.

   From (1) and $x \neq v$ it follows that for arbitrary $a \in |\mathcal{A}|$

$$h'\binom{x}{a} = h\binom{x}{a}\binom{v}{\bar{h}(t)} \tag{5.1}$$

   (2) implies that the induction hypothesis can be applied to $\alpha(v/t)$, which together with (5.1) yields

$$\mathcal{A} \models \alpha(v/t)[h\binom{x}{a}] \iff \mathcal{A} \models \alpha[h'\binom{x}{a}] \tag{5.2}$$

   for every $a \in |\mathcal{A}|$.

   Thus, we have the following sequence of equivalences:

$$\begin{aligned}
\mathcal{A} \models \varphi(v/t)[h] &\iff \mathcal{A} \models \forall x\alpha(v/t)[h] \\
&\iff \mathcal{A} \models \alpha(v/t)[h\binom{x}{a}] \quad \text{(for all } a \in |\mathcal{A}|) \\
&\overset{(5.2)}{\iff} \mathcal{A} \models \alpha[h'\binom{x}{a}] \quad \text{(for all } a \in |\mathcal{A}|) \\
&\iff \mathcal{A} \models \forall x\alpha[h'] \\
&\iff \mathcal{A} \models \varphi[h']
\end{aligned}$$

   This finishes the proof.

■

Exercise 5.1.3. Find an example that shows that the conclusion of the preceding lemma does not hold if t is not free for v in α.

Theorem 5.1.4 (Correctness). *Deductions are correct, which means, according to Remark 4.4.3, that*

$$\Sigma \vdash \varphi \qquad \textit{implies} \qquad \Sigma \Vdash \varphi$$

Proof: Let $\mathcal{L}$ be a formal language, Σ a set of $\mathcal{L}$-formulae and $\mathcal{A}$ an $\mathcal{L}$-structure such that $\mathcal{A} \models \Sigma$. Furthermore, let $(\varphi_1, \ldots, \varphi_n)$ be a deduction from Σ. We show by induction on n that $\mathcal{A} \models \varphi_i$ for $i = 1, \ldots, i = n$.

1. $n = 1$. Then either
   (a) φ_1 is an element of Σ, from which $\mathcal{A} \models \varphi_1$ immediately follows, or
   (b) φ_1 is a logical axiom. Three cases are possible:
      i. φ_1 is a first-order tautology. According to Theorem 3.3.13 φ_1 is valid, therefore $\mathcal{A} \models \varphi_1$.
      ii. φ_1 is an identity axiom. The validity of (I4) has been shown in Subsection 4.2.2, whereas the proof of the validity of (I1)-(I3) is the contents of Exercise 4.2.2. The validity of φ_1 implies $\mathcal{A} \models \varphi_1$.
      iii. φ_1 is a quantifier axiom. This means

$$\varphi = \forall v\, \alpha \to \alpha(v/t)$$

for a formula α and a term t that is free for v in α. Let h be an arbitrary variable assignment such that

$$\mathcal{A} \models \forall v \alpha[h]. \tag{5.3}$$

In order for $\mathcal{A} \models \varphi[h]$ to hold we have to show that (5.3) implies

$$\mathcal{A} \models \alpha(v/t)[h], \tag{5.4}$$

according to Definition 3.2.6. But (5.3) means that

$$\mathcal{A} \models \alpha[h\binom{v}{a}] \tag{5.5}$$

for all $a \in |\mathcal{A}|$. Setting $a = \bar{h}(t)$ in (5.5) yields

$$\mathcal{A} \models \alpha[h\binom{v}{\bar{h}(t)}]. \tag{5.6}$$

Since t is free for v in α, we can apply Lemma 5.1.2 to (5.6) yielding (5.4), which is what we had to show.

2. $n \to n+1$. Let us assume that for an arbitrary deduction $(\tau_1, \ldots, \tau_n)$ from Σ of length n we have $\mathcal{A} \models \tau_i$ for $i = 1, \ldots, n$, and consider a deduction $(\varphi_1, \ldots, \varphi_{n+1})$ of length $n+1$. We have to show that $\mathcal{A} \models \varphi_i$ for $i = 1, \ldots, n+1$ holds.

   Since $(\varphi_1, \ldots, \varphi_n)$ is a deduction of length n, the induction hypothesis yields $\mathcal{A} \models \varphi_i$ for $i = 1, \ldots, n$. Therefore, $\mathcal{A} \models \varphi_{n+1}$ remains to be shown.

   If φ_{n+1} is an element of Σ or a logical axiom, the claim follows as in the proof of the induction basis $n = 1$. Thus, let us assume that φ_{n+1} is the conclusion of an inference rule.

   (a) There are $i_1, i_2 \in \{1, ..., n\}$ such that $\langle\{\varphi_{i_1}, \varphi_{i_2}\}, \varphi_{n+1}\rangle$ is an instance of modus ponens (MP). Since $i_1, i_2 < n+1$ we have $\mathcal{A} \models \varphi_{i_1}$, $\mathcal{A} \models \varphi_{i_2}$. The correctness of (MP), which has been shown in 4.3, yields $\mathcal{A} \models \varphi_{n+1}$.

   (b) φ_{n+1} is the conclusion of an instance of the inference rule $(\forall)$. Because this rule is correct (Lemma 4.3.1), the proof is the same as for the preceding case.

■

5.2 The undecidability of first-order logic

In this section we touch upon the notion of decidability. Since this notion is treated extensively in module N2, we abstain from giving an exact definition, but content ourselves with an informal description that enables the reader to grasp this notion in the context of first-order logic. Our proof of the undecidability of first-order logic will be based on the insolvability of Post's correspondence problem, which is dealt with in module N2.

A *decision problem* is a question that usually possesses several unspecified parameters and is answerable either by "yes" or "no". An instance of the problem is obtained by specifying the given parameters.

One of the most famous decision problems is *The Travelling Salesperson Problem*. Given a list of cities $c_1, \ldots, c_n$ and distances d_{ij}, $i, j \in \{1, \ldots, n\}$, $i \neq j$, between the cities as well as a bound b, is there a tour starting and ending in c_1 that visits every other city exactly once such that its length is less than b?

The decision problem that interests us here is the following: If we have a formal language $\mathcal{L}$, a set of $\mathcal{L}$-formulae Σ and an $\mathcal{L}$-formula φ, does the relation $\Sigma \vdash \varphi$ hold? The parameters of this decision problem are the alphabet of $\mathcal{L}$, Σ and φ. We speak of the *decidability of first-order* logic. We will show that even for $\Sigma = \emptyset$ there is no decision procedure (see below) for this question if $\mathcal{L}$ contains at least a binary relation symbol, two unary function symbols and a constant symbol.

A *decision procedure* for a problem is an algorithm or computer program that answers with "yes" if the input is an instance that is solvable, and "no" otherwise.

The existence of a procedure deciding first-order logic would be an algorithm that on input $\mathcal{L}$, Σ and φ has output "yes" if the relation $\Sigma \vdash \varphi$ holds, and "no" otherwise.

The existence of a decision procedure for the Travelling Salesperson Problem is clear. Just check for a given instance all tours that start and end in the first city and visit all the other cities exactly once, then compute the length of each tour and compare it with the bound b and finally answer "yes" if there is a tour whose length is smaller than b. Here the question of interest is whether there is an efficient procedure.

In order to formulate *Post's correspondence problem* we have to introduce the notions of *word over a set* and the *concatenation of words*. Let S be an arbitrary set. A word over S is a finite sequence $a_1 a_2 \ldots a_n$ of elements $a_i \in S$, $i = 1, \ldots, n$, $n \in \mathbb{N}$ arbitrary. The number n is called the *length* of the word. The symbol ϵ denotes the word of length 0, which is also called the empty word. Given two words $x = a_1 \ldots a_m$ and $y = b_1 \ldots b_n$ the concatenation xy of x and y is the sequence $a_1 \ldots a_m b_1 \ldots b_n$ of length

$m+n$. The inductive definition of the concatenation of any finite number of words is straightforward. With S^*, finally, we denote the *set of all words over* S.

Now we are in a position to formulate Post's correspondence problem.

Definition 5.2.1 (Post's correspondence problem). Let $(x_1, y_1), \ldots,$ (x_k, y_k) be a finite number of pairs of words $x_i, y_i \in \{0,1\}^*$.

Problem: Is there a *finite* sequence of indices $i_1, \ldots, i_n \in \{1, \ldots, k\}$, with repetitions permitted, such that

$$x_{i_1} x_{i_2} \ldots x_{i_n} = y_{i_1} y_{i_2} \ldots y_{i_n},$$

that is, the concatenation of the first elements of the sequence of pairs is the same as the concatenation of the second elements?

Example 5.2.2. Let the sequence $(1, 101), (10, 00), (011, 11)$ of pairs be given. Choosing $i_1 = 1, i_2 = 3, i_3 = 2, i_4 = 3$ yields

$$x_{i_1} x_{i_2} x_{i_3} x_{i_4} = 1|011|10|011 = 101|11|00|11 = y_{i_1} y_{i_2} y_{i_3} y_{i_4},$$

with the auxiliary symbol $|$ separating the occurrences of the x_i's and y_i's. The chosen sequence therefore solves the given problem.

Theorem 5.2.3. *There is no algorithm giving on input* $K = \{(x_1, y_1), \ldots, (x_k, y_k)\}$, $k \in \mathbb{N}$, $x_i, y_i \in \{0,1\}^*$ *arbitrary, the answer "yes" if there is a solution for the correspondence problem* K *and "no" if there is none.*

Proof: Cf. module N2 or consult the literature, for instance [hermes78]. ■

Now we turn to the decision problem that is the focus of this section. By relating it to Post's correspondence problem we show that there is no algorithm for deciding whether, for an arbitrary $\mathcal{L}$-formula φ, $\mathcal{L}$ a first-order language containing a binary relation symbol, two unary function symbols and a constant symbol, $\vdash \varphi$ holds, or equivalently $\mathcal{A} \models \varphi$ for all $\mathcal{L}$-structures $\mathcal{A}$ (cf. Completeness Theorem 6.5.1 and Corollary 6.5.2). That is, there is no algorithm language for deciding the validity of an $\mathcal{L}$-formula φ for such a formal language.

Theorem 5.2.4. *Let $\mathcal{L}$ be a first-order language containing a binary relation symbol P, two unary function symbols f_0, f_1 and a constant symbol a. Then there is no algorithm yielding for input $\varphi \in Fml(\mathcal{L})$ the answer "yes" if φ is valid, and "no" otherwise.*

Proof: We proceed by showing that from the existence of an algorithm deciding the validity of a formula, an algorithm solving Post's correspondence problem could be inferred.

Let $K = \{(x_1, y_1), \ldots, (x_k, y_k)\}$, $k \in \mathbb{N}$, $x_i, y_i \in \{0,1\}^*$ be an instance of the correspondence problem. We define an $\mathcal{L}$-formula φ_K such that an algorithm deciding the validity of φ_K yields a procedure deciding K. This is achieved if for φ_K the following holds:

$$\varphi_K \text{ is valid} \qquad \Longleftrightarrow \qquad K \text{ is solvable} \tag{5.7}$$

First, we assign to every word $j_1 j_2 \ldots j_s \in \{0,1\}^*$ and $\mathcal{L}$-term t an $\mathcal{L}$-term $f_{j_1 j_2 \ldots j_s}(t)$ by defining

$$f_{j_1 j_2 \ldots j_s}(t) := f_{j_s}(f_{j_{s-1}} \ldots (f_{j_2}(f_{j_1}(t))) \ldots). \tag{5.8}$$

φ_K is then defined as the formula $\rho \wedge \sigma \to \tau$, where

$$\begin{aligned}
\rho &:= P(f_{x_1}(a), f_{y_1}(a)) \wedge \ldots \wedge P(f_{x_k}(a), f_{y_k}(a)) \\
\sigma &:= \forall u \forall v (P(u,v) \to (P(f_{x_1}(u), f_{y_1}(v)) \wedge \ldots \wedge P(f_{x_k}(u), f_{y_k}(v)))) \\
\tau &:= \exists z P(z,z)
\end{aligned}$$

Now we show equivalence (5.7):

1. *φ_K is valid $\Longrightarrow$ K is solvable.* We construct an appropriate $\mathcal{L}$-structure $\mathcal{A}$ as follows:

$$\begin{aligned}
|\mathcal{A}| &:= \{0,1\}^* \\
a^{\mathcal{A}} &:= \epsilon \text{ (empty word)} \\
f_0^{\mathcal{A}}(x) &:= x0 \\
f_1^{\mathcal{A}}(x) &:= x1
\end{aligned}$$

$$P^{\mathcal{A}} := \{(x,y) \in \{0,1\}^* \times \{0,1\}^* \mid \exists i_1, i_2, \ldots, i_s \in \{1, \ldots, k\} \text{ such that } x = x_{i_1} \ldots x_{i_s} \text{ and } y = y_{i_1} \ldots y_{i_s}\}$$

From the definition of $P^{\mathcal{A}}$ follows that this part of the proof is finished if we can show $\mathcal{A} \models \tau$.

From the definition of $f_0^{\mathcal{A}}$ and $f_1^{\mathcal{A}}$ and (5.8) follows

$$f_x^{\mathcal{A}}(\gamma) = \gamma x \text{ for } x, \gamma \in \{0,1\}^*. \tag{5.9}$$

Since $a^{\mathcal{A}} = \epsilon$ we obtain from (5.9) $f_{x_i}^{\mathcal{A}}(a) = \epsilon x_i = x_i$ and $f_{y_i}^{\mathcal{A}}(a) = \epsilon y_i = y_i$ for $i = 1, \dots, k$, thus $\mathcal{A} \models \rho$ since $P^{\mathcal{A}}(x_i, y_i)$ according to the definition of $P^{\mathcal{A}}$.

Furthermore, it follows from (5.9) that

$$P^{\mathcal{A}}(x, y) \text{ implies } P^{\mathcal{A}}(f_{x_i}^{\mathcal{A}}(x), f_{y_i}^{\mathcal{A}}(y))$$

for all $i \in \{1, \dots, k\}$:

$P^{\mathcal{A}}(x, y)$ signifies the existence of a sequence $i_1, \dots, i_s$ of elements from $\{1, \dots, k\}$ such that $x = x_{i_1} \dots x_{i_s}$ and $y = y_{i_1} \dots y_{i_s}$. This sequence can be extended by $i_{s+1} := i$ to yield a sequence with $f_{x_i}^{\mathcal{A}}(x) = x_{i_1} \dots x_{i_s} x_i$ and $f_{y_i}^{\mathcal{A}}(y) = y_{i_1} \dots y_{i_s} y_i$ from which $P^{\mathcal{A}}(f_{x_i}^{\mathcal{A}}(x), f_{y_i}^{\mathcal{A}}(y))$ follows. Thus $\mathcal{A} \models \sigma$.

The last two paragraphs yield $\mathcal{A} \models \rho \wedge \sigma$, which, together with the validity of $\rho \wedge \sigma \to \tau$, implies $\mathcal{A} \models \tau$.

2. *K is solvable $\Longrightarrow$ φ_K is valid.*

   Let $i_1, \dots, i_s$ be a solution, that is, $x_{i_1} \dots x_{i_s} = y_{i_1} \dots y_{i_s}$. We have to show that $\mathcal{A} \models \rho \wedge \sigma \to \tau$ for an arbitrary $\mathcal{L}$-structure $\mathcal{A}$.

   We can assume $\mathcal{A} \models \rho \wedge \sigma$ since otherwise $\mathcal{A} \models \rho \wedge \sigma \to \tau$ immediately follows from Subsection 3.3.2 and the fact that ρ, σ and τ are sentences (The assertions in Subsection 3.3.2 include valuations, but since sentences are concerned here, the valuations can be dropped.)

   Let us define $\mu : \{0,1\}^* \longrightarrow |\mathcal{A}|$ inductively as follows:

$$\begin{aligned} \mu(\epsilon) &:= a^{\mathcal{A}} \\ \mu(x0) &:= f_0^{\mathcal{A}}(\mu(x)) \\ \mu(x1) &:= f_1^{\mathcal{A}}(\mu(x)) \end{aligned}$$

   From the definition of μ and (5.8) we obtain (as in (5.9)) for $i \in \{1, \dots, k\}$

$$\mu(x_i) = (f_{x_i}(a))^{\mathcal{A}} \qquad \text{and} \qquad \mu(y_i) = (f_{y_i}(a))^{\mathcal{A}} \tag{5.10}$$

as well as

$$\mu(\gamma\delta) = f_\delta^{\mathcal{A}}(\mu(\gamma)) = f_\delta^{\mathcal{A}}(f_\gamma^{\mathcal{A}}(a^{\mathcal{A}})) \tag{5.11}$$

The assumption $\mathcal{A} \models \rho$ and (5.10) imply

$$(\mu(x_i), \mu(y_i)) \in P^{\mathcal{A}} \qquad (i \in \{1, \ldots, k\}). \tag{5.12}$$

The assumption $\mathcal{A} \models \sigma$ states that if $b_1, b_2 \in |\mathcal{A}|$ and $i \in \{1, \ldots, k\}$, then

$$(b_1, b_2) \in P^{\mathcal{A}} \quad \Longrightarrow \quad (f_{x_i}^{\mathcal{A}}(b_1), f_{y_i}^{\mathcal{A}}(b_2)) \in P^{\mathcal{A}}. \tag{5.13}$$

Choosing i_1 in (5.12) and i_2 in (5.13) yields

$$(f_{x_{i_2}}^{\mathcal{A}}(\mu(x_{i_1})), f_{y_{i_2}}^{\mathcal{A}}(\mu(y_{i_1}))) \in P^{\mathcal{A}},$$

which according to (5.11) is equivalent to

$$(\mu(x_{i_1}x_{i_2}), \mu(y_{i_1}y_{i_2})) \in P^{\mathcal{A}},$$

from which with (5.13), choosing successively $i_3, \ldots, i_s$ we finally obtain

$$(\mu(x_{i_1} \ldots x_{i_s}), \mu(y_{i_1} \ldots y_{i_s})) \in P^{\mathcal{A}}. \tag{5.14}$$

Since we have assumed $x_{i_1} \ldots x_{i_s} = y_{i_1} \ldots y_{i_s}$, we have $(d, d) \in P^{\mathcal{A}}$ with $d := \mu(x_{i_1} \ldots x_{i_s}) = \mu(y_{i_1} \ldots y_{i_s}) \in |\mathcal{A}|$.

However, this means $\mathcal{A} \models \tau$, and therefore $\mathcal{A} \models \rho \wedge \sigma \to \tau = \varphi_K$, and the proof is finished.

■

From the preceding theorem we obtain the corollary that the situation remains the same if we test the *satisfiability* of a formula.

Corollary 5.2.5. *Let $\mathcal{L}$ be a first-order language containing a binary relation symbol P, two unary function symbols f_0, f_1 and a constant symbol a. Then there is no algorithm yielding for input $\varphi \in Fml(\mathcal{L})$ the answer "yes" if φ is satisfiable, and "no" otherwise.*

Proof: An $\mathcal{L}$-formula φ is satisfiable if and only if $\neg\varphi$ is not valid. A decision procedure for satisfiability would thus yield at the same time a decision procedure for validity contradicting Theorem 5.2.4. ■

5.3 Derived inference rules

The proof of Lemma 4.4.6 was our first example of a formal proof. In order to use the assumption that α and $\neg\alpha$ are already deduced, the deduction had to be extended with a first-order tautology and then by applying the inference rule (MP). This cumbersome procedure could be abbreviated by the rule $\langle \alpha \wedge \neg\alpha, \varphi \rangle$.

Let us consider another example. In ordinary mathematics one would immediately use the formula $\varphi \to \rho$, after having established $\varphi \to \psi$ and $\psi \to \rho$. In a deduction one would have to combine deductions $D_1 = (\ldots, \varphi \to \psi)$ and $D_2 = (\ldots, \psi \to \rho)$ to a deduction $D = (\ldots, \varphi \to \psi, \ldots, \psi \to \rho)$ and then extend D by the sequence $((\varphi \to \psi) \to ((\psi \to \rho) \to (\varphi \to \rho)), (\psi \to \rho) \to (\varphi \to \rho), \varphi \to \rho)$.

The preceding paragraph shows that it is justified to do without the deduction carried out there and use the inference rule

$$\langle \{\varphi \to \psi, \varphi \to \rho\}, \varphi \to \rho \rangle,$$

with premises $\varphi \to \psi$, $\psi \to \rho$ and conclusion $\varphi \to \rho$. It allows to extend the deduction D directly by $\varphi \to \rho$.

It is called a *derived inference rule* since it can be proved within our formal system. The inference rule derived above is called the *chain rule* and is the rule (CR) of Lemma 5.3.1.

Although we will not systematically formalize proofs, we will in this section introduce derived inference rules to shorten deductions. This is necessary to handle the proofs of the properties of our formal system, especially completeness. The following lemmata are a collection of such rules.

Lemma 5.3.1. *Let $\mathcal{L}$ be a formal language and φ, ψ and ρ be arbitrary $\mathcal{L}$-formulae and Σ an arbitrary set of $\mathcal{L}$-formulae. If there are deductions from Σ of the premises of any of the rules below then there is a deduction from Σ of the conclusion of the rule.*

1. *(a)*

$$\frac{\varphi \wedge \psi}{\varphi} \qquad (\wedge B_1)$$

(b)

$$\frac{\varphi \wedge \psi}{\psi} \qquad (\wedge B_2)$$

2. (a)

$$\frac{\varphi}{\varphi \vee \psi} \qquad (\vee B_1)$$

(b)

$$\frac{\psi}{\varphi \vee \psi} \qquad (\vee B_2)$$

3. (a)

$$\frac{\varphi \to \psi, \psi \to \rho}{\varphi \to \rho} \qquad (\mathrm{CR})$$

(b)

$$\frac{\varphi \to \psi}{\neg\psi \to \neg\varphi} \qquad (\mathrm{CP})$$

4. (a)

$$\frac{\varphi \leftrightarrow \psi}{\varphi \to \psi} \qquad (\leftrightarrow_1)$$

(b)

$$\frac{\varphi \to \psi, \psi \to \varphi}{\varphi \leftrightarrow \psi} \qquad (\leftrightarrow_2)$$

5. (a)

$$\frac{\varphi, \psi}{\varphi \wedge \psi} \qquad (\wedge)$$

(b)

$$\frac{\varphi \to \rho, \psi \to \rho}{\varphi \vee \psi \to \rho} \qquad (\vee)$$

(CR) stands for chain rule and (CP) for contraposition.

Proof: We show the assertion for (CR), which was shown less formally in the introduction of this section. The other derived inference rules, whose proof is left as Exercise 8.5.6, are also shown by drawing on a first-order tautology and then applying the rule (MP).

The deductions of the premises can be concatenated to yield a deduction

$$D = (\ldots, \varphi \to \psi, \ldots, \psi \to \rho).$$

D can be extended by the formulae

$$(\varphi \to \psi) \to ((\psi \to \rho) \to (\varphi \to \rho)) \qquad \text{(first-order tautology)}$$
$$(\psi \to \rho) \to (\varphi \to \rho) \qquad \text{(MP)}$$
$$\varphi \to \rho \qquad \text{(MP)}$$

since the first is a first-order tautology and the second and third can then be obtained by applying the rule (MP). The extension of D is a deduction of the conclusion of the rule. ■

Lemma 5.3.2. *Let $\mathcal{L}$ be a formal language, φ be an arbitrary $\mathcal{L}$-formula and Σ an arbitrary set of $\mathcal{L}$-formulae. If there is a deduction from Σ of the premise of any of the rules below, then there is a deduction from Σ of the conclusion of the rule.*

1. *For any variable v*

$$\frac{\varphi}{\forall v\varphi} \tag{res$\forall$}$$

2. *If t is free for v in φ then*

$$\frac{\forall v\varphi}{\varphi(v/t)} \tag{S}$$

(res$\forall$) stands for resulting generalization and (S) for specialization.

Proof: In both cases we extend a deduction of the premise to a deduction of the conclusion of the rule that we want to prove.

1. Let $D = (\dots, \varphi)$ be a deduction of φ from Σ. We extend D by the following formulae. ($\perp$ is the sentence falsum defined in Definition 3.2.13, however, any sentence would do.):

$$\begin{aligned} \varphi \to ((\perp \vee \neg\perp) \to \varphi) & \qquad \text{(f.-o. tautology)} \\ (\perp \vee \neg\perp) \to \varphi & \qquad \text{(MP)} \\ (\perp \vee \neg\perp) \to \forall v\varphi & \qquad (\forall) \\ \perp \vee \neg\perp & \qquad \text{(f.-o. tautology)} \\ \forall v\varphi & \qquad \text{(MP)} \end{aligned}$$

   The first and fourth formulae are first-order tautologies, the second and fifth are obtained by applying the inference rule (MP) and the third is the result of applying ($\forall$) to the preceding formula, which is possible, as $\perp \vee \neg\perp$ does not contain any free variables.

2. Let $D = (\dots, \forall v\varphi)$ be a deduction of φ from Σ. We extend D by the following formulae:

$$\begin{aligned} \forall v\varphi \to \varphi(v/t) & \qquad \text{((Q) since } t \text{ free for } v \text{ in } \varphi) \\ \varphi(v/t) & \qquad \text{(MP)} \end{aligned}$$

   The first formula is an instance of a quantifier axiom 4.2.3 since t is supposed to be free for v in φ. The second is obtained by (MP).

∎

Lemma 5.3.3. *Let $\mathcal{L}$ be a formal language, R_i an arbitrary relation symbol, f_j a function symbol of $\mathcal{L}$ and $t', t'', t_1, t_2, \ldots$ arbitrary $\mathcal{L}$-terms. If there is a deduction from Σ of the premise of any of the rules below, then there is a deduction from Σ of the conclusion of the rule.*

1.

$$\frac{t_1 \doteq t_2}{t_2 \doteq t_1} \tag{Sym}$$

2.

$$\frac{t_1 \doteq t_2, t_2 \doteq t_3}{t_1 \doteq t_3} \tag{Tr}$$

3.

$$\frac{t' \doteq t''}{R_i(t_1, \ldots, t', \ldots, t_{\lambda(i)}) \to R_i(t_1, \ldots, t'', \ldots, t_{\lambda(i)})} \tag{R_i}$$

4.

$$\frac{t' \doteq t''}{f_j(t_1, \ldots, t', \ldots, t_{\mu(j)}) \doteq f_j(t_1, \ldots, t'', \ldots, t_{\mu(j)})} \tag{f_j}$$

Proof: We prove the first assertion and leave the others to the reader as Exercise 8.5.7. Let $D = (\ldots, t_1 \doteq t_2)$ be a deduction of $t_1 \doteq t_2$ from Σ. We extend D by the following formulae

1. $v \doteq v$ (Identity axiom (I1), 4.2.2)
2. $\forall v(v \doteq v)$ ((res$\forall$) applied to 1.)
3. $t_1 \doteq t_1$ ((S) applied to 2.)
4. $v_1 \doteq v_2 \to (v_1 \doteq v_1 \to v_2 \doteq v_1)$ (Identity axiom (I2) with $v_1 = v_3$)
5. $\forall v_1(v_1 \doteq v_2 \to (v_1 \doteq v_1 \to v_2 \doteq v_1))$ ((res$\forall$) applied to 4.)
6. $(t_1 \doteq v_2 \to (t_1 \doteq t_1 \to v_2 \doteq t_1))$ ((S) applied to 5.)
7. $\forall v_2(t_1 \doteq v_2 \to (t_1 \doteq t_1 \to v_2 \doteq t_1))$ ((res$\forall$) applied to 6.)
8. $t_1 \doteq t_2 \to (t_1 \doteq t_1 \to t_2 \doteq t_1)$ ((S) applied to 7.)

9. $t_1 \doteq t_1 \to t_2 \doteq t_1$ ((MP) applied to the assumption and 8.)

10. $t_2 \doteq t_1$ ((MP) applied to 3. and 9.)

According to the comments at the end of each line, every formula is either an instance of an axiom or the conclusion of a rule. Therefore, the extension is a deduction. ■

Exercise 5.3.4. Explain what is wrong in the above proof if the 6th and the 7th line are replaced by:

6. $\forall v_2 \forall v_1 (v_1 \doteq v_2 \to (v_1 \doteq v_1 \to v_2 \doteq v_1))$ ((res$\forall$) applied to 5.)

7. $\forall v_2 (t_1 \doteq v_2 \to (t_1 \doteq t_1 \to v_2 \doteq t_1))$ ((S) applied to 6.)

5.4 Sentences suffice

In the informal discussion of the truth of a formula in Section 2.3 we noticed that the truth value depended on the interpretation of its free variables. We then defined the universal closure $\forall\varphi$ of a formula φ to obtain a sentence, that is, a formula without free variables, pertaining to φ. It has been claimed that a formula and its universal closure were closely related.

A formula and its universal closure are closely related with respect to validity, that is, to the meta language symbol $\models$. This statement is the contents of Corollary 3.2.16, which states that a formula is valid if and only if its universal closure is valid.

The following lemma asserts that the corresponding fact is true with respect to the meta language symbol $\vdash$. That is, a formula can be deduced from a set of formulae if and only if this is the case for its universal closure. The second part of the lemma states that a formula and its universal closure have the same deductive strength, that is, a formula and its universal closure allow the same formulae to be deduced.

Lemma 5.4.1. *Let $\mathcal{L}$ be a formal language, $\Sigma \subset Fml(\mathcal{L})$ and φ and ψ $\mathcal{L}$-formulae. Then the following holds:*

1. $\Sigma \vdash \varphi$ *iff* $\Sigma \vdash \forall\varphi$,

2. $\Sigma \cup \{\varphi\} \vdash \psi$ *iff* $\Sigma \cup \{\forall\varphi\} \vdash \psi$.

Proof:

1. It suffices to show $\Sigma \vdash \varphi$ iff $\Sigma \vdash \forall v\, \varphi$ for an arbitrary $v \in \mathrm{Vbl}(\mathcal{L})$.

   (a) $\Sigma \vdash \varphi$ implies $\Sigma \vdash \forall v\varphi$: let D be a deduction of φ from Σ. Applying the derived inference rule (res$\forall$) with premise φ yields the deduction $(D, \forall v\varphi)$ of $\forall v\varphi$ from Σ.

   (b) $\Sigma \vdash \forall v\varphi$ implies $\Sigma \vdash \varphi$: let D be a deduction of $\forall v\varphi$ from Σ. Applying the derived inference rule (S) with premise $\forall v\varphi$ yields the deduction $(D, \varphi(v/v))$ of $\varphi(v/v)$ from Σ. (S) can be applied, because v is free for v in φ. This proves the assertion, because $\varphi = \varphi(v/v)$.

2. It suffices to show $\Sigma \cup \{\varphi\} \vdash \psi$ iff $\Sigma \cup \{\forall v\varphi\} \vdash \psi$ for an arbitrary $v \in \mathrm{Vbl}(\mathcal{L})$.

   (a) $\Sigma \cup \{\varphi\} \vdash \psi$ implies $\Sigma \cup \{\forall v\varphi\} \vdash \psi$: if there is a deduction D of ψ from $\Sigma \cup \varphi$ in which φ does not occur, then D is also a deduction of ψ from $\Sigma \cup \forall v\varphi$ and we are finished. Therefore, we can assume that D is of the form $(\varphi, \varphi_2, \ldots, \varphi_n)$. φ being the first formula is no restriction. Then $(\forall v\varphi, \varphi, \varphi_2, \ldots, \varphi_n)$ is a deduction of ψ from $\Sigma \cup \{\forall v\varphi\}$: as above φ can be obtained from $\forall v\varphi$ with the derived inference rule (S).

   (b) $\Sigma \cup \{\forall v\varphi\} \vdash \psi$ implies $\Sigma \cup \{\varphi\} \vdash \psi$: the same argument as for the converse direction can be applied by interchanging $\forall v\varphi$ and φ and applying the derived inference rule (res$\forall$).

■

In ordinary mathematics our use of quantifiers is mostly implicit. This corresponds to a formula that contains free variables. However, the intention is that the free variables may assume any value of the structure considered.

The equivalence of a formula and its universal closure is relatively obvious with regard to validity. All possible assignments for a variable within the scope of a universal quantifier must be tested, which amounts to the variable being free.

The equivalence on the syntactic side is less obvious. In this section we have shown that the logical axioms and the inference rule imply this equivalence.

Remark 5.4.2 (Sentences suffice). Lemma 5.4.1 allows us to restrict ourselves to sentences when investigating properties of the relation $\vdash$. Formulae containing free variables can be replaced by their universal closure. That is - "Sentences suffice".

Considering only sentences has the advantage of not having to check whether certain (derived) inference rules and logical axioms can be applied, since substituting terms that are not free cannot occur anymore. Also, the Deduction Theorem (Theorem 5.5.1), which is the focus of the next section, depends on a certain formula being a sentence.

5.5 The Deduction Theorem

In this section we prove another result about the symbol $\vdash$ of the meta-language. It is easy to show that if a set of formulae Σ proves an implication $\varphi \to \psi$, then Σ together with the antecedent φ of the implication proves the succedent ψ.

That this statement and its converse are true (but the latter under the additional condition that φ is a sentence), is the contents of the following theorem which is very important in proving further results.

Theorem 5.5.1 (Deduction theorem). *Let Σ be a set of formulae, φ a sentence and ψ a formula. Then*

$$\Sigma \cup \{\varphi\} \vdash \psi \quad \textit{if and only if} \quad \Sigma \vdash \varphi \to \psi. \tag{Ded}$$

Proof: First suppose $\Sigma \vdash \varphi \to \psi$ with a deduction D. D is also a deduction of $\varphi \to \psi$ from the enlarged set $\Sigma \cup \{\varphi\}$. Then (D, φ) is a deduction from $\Sigma \cup \{\varphi\}$. Applying (MP) yields the deduction (D, φ, ψ) from $\Sigma \cup \{\varphi\}$, thus $\Sigma \cup \{\varphi\} \vdash \psi$.

For the other direction we prove the following by an induction on the length of a deduction $D = (\varphi_1, \dots, \varphi_n)$ establishing $\Sigma \cup \{\varphi\} \vdash \psi$ (hence $\varphi_n = \psi$):

The sequence $(\varphi \to \varphi_1, \dots, \varphi \to \varphi_n)$ can be supplemented to obtain a deduction $\widetilde{D}$ of $\varphi \to \varphi_n$ from Σ. $\varphi \to \varphi_1, \dots, \varphi \to \varphi_n$ will occur in $\widetilde{D}$ in the same order, additional formulae being interspersed to produce a deduction. Since $\varphi_n = \psi$, $\widetilde{D}$ will then be a deduction of $\varphi \to \psi$ from Σ.

1. $n = 1$, *that is* $D = (\varphi_1)$: this means that φ_1 is either an element of $\Sigma \cup \varphi$ or a logical axiom, and $\psi = \varphi_1$. We have to look at the cases:

   (a) φ_1 *is* φ: then $\widetilde{D} := \varphi \to \varphi_1$ is the first-order tautology $\varphi \to \varphi$. The sequence $\widetilde{D} := (\varphi \to \varphi_1)$ is the desired deduction.

   (b) φ_1 *is an element of* Σ *or a logical axiom*: then $\widetilde{D} = (\varphi_1, \varphi_1 \to (\varphi \to \varphi_1), \varphi \to \varphi_1)$ is a deduction of $\varphi \to \psi$ from Σ, as the second formula is a first-order tautology and the third is obtained from the first two by (MP).

2. $n \to n+1$, *that is, there is a deduction* $D = (\varphi_1, \dots, \varphi_{n+1})$ *from* $\Sigma \cup \varphi$ *with* $\psi = \varphi_{n+1}$: applying the induction hypothesis to the shorter deduction $D = (\varphi_1, \dots, \varphi_n)$ from $\Sigma \cup \varphi$ yields a deduction $\widetilde{D} = (\widetilde{\varphi_1}, \dots, \widetilde{\varphi_m})$ $(m \geq n)$ of the formula $\varphi \to \varphi_n$ from Σ in which the formulae $\varphi \to \varphi_1, \dots, \varphi \to \varphi_{n-1}$ occur.

   We are now going to extend $\widetilde{D}$ to a deduction from Σ of $\varphi \to \varphi_{n+1}$. Because $\varphi_{n+1} = \psi$ we have $\varphi \to \varphi_{n+1} = \varphi \to \psi$. We have to distinguish between the ways φ_{n+1} was obtained in D from formulae with smaller indices.

   (a) φ_{n+1} is either an element of $\Sigma \cup \varphi$ or a logical axiom. As above we have two subcases.

      i. φ_{n+1} is φ: then $\varphi \to \varphi_{n+1}$ is a first-order tautology from which we obtain the deduction $(\widetilde{D}, \varphi \to \varphi_{n+1})$ from Σ.

      ii. φ_{n+1} is an element of Σ or a logical axiom: then

      $$(\widetilde{D}, \varphi_{n+1}, \varphi_{n+1} \to (\varphi \to \varphi_{n+1}), \varphi \to \varphi_{n+1})$$

      is a deduction of $\varphi \to \varphi_{n+1}$ from Σ, as the second formula is a first-order tautology and the third is obtained from the first two by (MP).

   (b) φ_{n+1} was obtained in D by (MP). This means there are indices $1 \leq i < j \leq n$ such that $\varphi_j = \varphi_i \to \varphi_{n+1}$. The formulae $\varphi \to \varphi_i$ and $\varphi \to (\varphi_i \to \varphi_{n+1})$ occur in $\widetilde{D}$ by the induction hypothesis. Extending $\widetilde{D}$ with the formulae

   $$\begin{aligned} (\varphi \to (\varphi_i \to \varphi_{n+1})) \to ((\varphi \to \varphi_i) \to (\varphi \to \varphi_{n+1})) \\ (\varphi \to \varphi_i) \to (\varphi \to \varphi_{n+1}) \\ \varphi \to \varphi_{n+1} \end{aligned}$$

yields the desired deduction of $\varphi \to \varphi_{n+1}$ from Σ, as the first formula of the extension is a first-order tautology. The two others are obtained by (MP).

(c) φ_{n+1} was obtained in D by ($\forall$). This means there is an index $1 \le i \le n$ such that $\varphi_i = \alpha \to \beta$ and $\varphi_{n+1} = \alpha \to \forall v\beta$ with $v \notin \mathrm{Fr}(\alpha)$. The formula $\varphi \to (\alpha \to \beta)$ occurs in $\widetilde{D}$ by the induction hypothesis.

Extending $\widetilde{D}$ with the formulae

$$\begin{array}{lr}
(\varphi \to (\alpha \to \beta)) \to ((\varphi \wedge \alpha) \to \beta) & \text{(first–order tautology)} \\
(\varphi \wedge \alpha) \to \beta & \text{(MP)} \\
(\varphi \wedge \alpha) \to \forall v\beta & (\text{res}\forall) \\
((\varphi \wedge \alpha) \to \forall v\beta) \to (\varphi \to (\alpha \to \forall v\beta)) & \text{(first–order tautology)} \\
\varphi \to (\alpha \to \forall v\beta) & \text{(MP)}
\end{array}$$

yields the desired deduction of $\varphi \to \varphi_{n+1}$ from Σ. The application of (res$\forall$) is justified, since $v \notin \mathrm{Fr}(\alpha)$ by assumption and $v \notin \mathrm{Fr}(\varphi)$ since φ is a sentence. (This is the only place in the whole proof where the assumption that φ be a sentence is used!)

■

Solve the exercises of Section 8.7!

5.6 An example of a formal proof

Let L be the standard language. First we define our set of axioms $\Sigma := \{(1), (2), (3), (4), (5)\}$ where

(1) $\forall x, y\, (x \le y \vee y \le x)$

(2) $\forall x, y, z\, (x \le y \to x + z \le y + z)$

(3) $\forall x, y\, ((0 \le x \wedge 0 \le y) \to 0 \le x \cdot y)$

(4) $\forall x, y\, (x + (-x) \doteq 0 \wedge 0 + y \doteq y)$

(5) $\forall x\, (-x \cdot -x \doteq x \cdot x)$

We want to find a deduction from Σ of the sentence $\forall x(0 \leq x \cdot x)$. First we give an informal proof:

(1) implies $0 \leq x$ or $x \leq 0$.

1. If $0 \leq x$, then (3) implies $0 \leq x \cdot x$.
2. If $x \leq 0$, setting $y := 0$ and $z := -x$ in (2), we obtain

$$x \leq 0 \to x + (-x) \leq 0 + (-x),$$

and then applying (4) yields $0 \leq -x$. With (3) and (5) we obtain $0 \overset{(3)}{\leq} -x \cdot -x \overset{(5)}{=} x \cdot x$.

Therefore $0 \leq x \cdot x$ holds for every x.

The formal proof will be much longer. Notice that every element of Σ will occur exactly once. Furthermore, we will use the first-order tautology $\varphi \to (\varphi \wedge \varphi)$.

1. $\forall x, y\,(x \leq y \vee y \leq x)$ ((1))
2. $\forall y\,(x \leq y \vee y \leq x)$ ((S) with $\varphi(x/x)$ applied to 1.)
3. $x \leq 0 \vee 0 \leq x$ ((S) with $\varphi(y/0)$ applied to 2.)
4. $\forall x, y\,((0 \leq x \wedge 0 \leq y) \to 0 \leq x \cdot y)$ ((3))
5. $\forall y\,((0 \leq x \wedge 0 \leq y) \to 0 \leq x \cdot y)$ ((S) with $\varphi(x/x)$ applied to 4.)
6. $(0 \leq x \wedge 0 \leq x) \to 0 \leq x \cdot x$ ((S) with $\varphi(y/x)$ 5.)
7. $0 \leq x \to (0 \leq x \wedge\ 0 \leq x)$ (first-order tautology)
8. $0 \leq x \to 0 \leq x \cdot x$ ((CR) applied to 7. and 6.)
9. $\forall x, y, z\,(x \leq y \to x + z \leq y + z)$ ((2))
10. $\forall y, z\,(x \leq y \to x + z \leq y + z)$ ((S) with $\varphi(x/x)$ applied to 9.)
11. $\forall z\,(x \leq 0 \to x + z \leq 0 + z)$ ((S) with $\varphi(y/0)$ applied to 10.)
12. $x \leq 0 \to x + (-x) \leq 0 + (-x)$ ((S) with $\varphi(z/-x)$ applied to 11.)
13. $\forall x, y\,(x + (-x) \doteq 0 \wedge 0 + y \doteq y)$ ((4))

14. $\forall y\,(x+(-x) \doteq 0 \wedge 0+y \doteq y)$ ((S) with $\varphi(x/x)$ applied to 13.)

15. $x+(-x) \doteq 0 \wedge 0+(-x) \doteq -x$ ((S) with $\varphi(y/-x)$ applied to 14.)

16. $x+(-x) \doteq 0$ (($\wedge B_1$) applied to 15.)

17. $x+(-x) \leq 0+(-x) \rightarrow 0 \leq 0+(-x)$ ((R_1) applied to 16.)

18. $x \leq 0 \rightarrow 0 \leq 0+(-x)$ ((CR) applied to 12. and 17.)

19. $0+(-x) = -x$ (($\wedge B_2$) applied to 15.)

20. $0 \leq 0+(-x) \rightarrow 0 \leq -x$ ((R_1) applied to 19.)

21. $x \leq 0 \rightarrow 0 \leq -x$ ((CR) applied to 18. and 20.)

22. $\forall x\,(0 \leq x \rightarrow 0 \leq x \cdot x)$ ((res$\forall$) applied to 8.)

23. $0 \leq -x \rightarrow 0 \leq -x \cdot -x$ ((S) with $\varphi(x/-x)$ applied to 22.)

24. $\forall x\,(-x \cdot -x \doteq x \cdot x)$ ((5))

25. $-x \cdot -x \doteq x \cdot x$ ((S) with $\varphi(x/x)$ applied to 24.)

26. $0 \leq -x \cdot -x \rightarrow 0 \leq x \cdot x$ ((R_1) applied to 25.)

27. $0 \leq -x \rightarrow 0 \leq x \cdot x$ ((CR) applied to 23. and 26.)

28. $x \leq 0 \rightarrow 0 \leq x \cdot x$ ((CR) applied to 21. and 27.)

29. $(x \leq 0 \vee 0 \leq x) \rightarrow 0 \leq x \cdot x$ (($\vee$) applied to 8. and 28.)

30. $0 \leq x \cdot x$ ((MP) applied to 3. and 29.)

31. $\forall x\,(0 \leq x \cdot x)$ ((res$\forall$)) applied to 30.)

Chapter 6

Completeness

In this chapter we close the remaining gap in our efforts to set up a formal system that fulfils (S1) and (S2) of Hilbert's programme.

6.1 Introduction

In Chapter 5 we showed that our deductive system is correct, that is, $\Sigma \vdash \varphi$ implies $\Sigma \models \varphi$. We paraphrased this implication by saying that our deductive system is not too strong.

However, we also want the converse, that is, $\Sigma \models \varphi$ implies $\Sigma \vdash \varphi$, which we have paraphrased as "our deductive system is not too weak". With the premise $\Sigma \models \varphi$ it should be possible to deduce φ from Σ because there is no counter-example: φ is valid in every model of Σ. This is what the relation $\Sigma \models \varphi$ means.

In other words $\Sigma \nvdash \varphi$, that is, the unprovability of φ from Σ, should only hold if there is a structure $\mathcal{A}$ which is a model of Σ in which φ is not valid. This is nothing else than the contraposition

$$\Sigma \nvdash \varphi \qquad \text{implies} \qquad \Sigma \not\models \varphi. \tag{6.1}$$

Let us consider the the following two statements:

$$\Sigma \nvdash \varphi \qquad \Longleftrightarrow \qquad \Sigma \cup \{\neg\varphi\} \text{ consistent} \tag{6.2}$$

and

$$\Sigma \cup \{\neg\varphi\} \text{ consistent} \qquad \iff \qquad \Sigma \cup \{\neg\varphi\} \text{ has a model} \tag{6.3}$$

If we assume (6.2) and (6.3), then we can prove (6.1): From the premise $\Sigma \not\vdash \varphi$ of (6.1) we obtain with (6.2) and (6.3) that there is a model $\mathcal{A}$ of $\Sigma \cup \{\varphi\}$. However, the existence of $\mathcal{A}$ means $\Sigma \not\models \varphi$, the conclusion of (6.1), since $\mathcal{A}$ is a model of Σ in which φ is not true ($\mathcal{A}$ is a model of Σ and $\neg\varphi$, and no structure can be at the same time model of a formula and its negation).

The following lemma asserts statement (6.2), that is the equivalence of $\Sigma \not\vdash \varphi$ and the consistency of $\Sigma \cup \{\neg\varphi\}$.

Lemma 6.1.1. *Let $\mathcal{L}$ be a formal language, $\Sigma \subset Fml(\mathcal{L})$ and $\varphi \in Sen(\mathcal{L})$. Then*

$$\Sigma \not\vdash \varphi \quad \textit{iff} \quad \Sigma \cup \{\neg\varphi\} \textit{ is consistent.}$$

Proof: We show the equivalence of $\Sigma \vdash \varphi$ and the inconsistency of $\Sigma \cup \{\neg\varphi\}$.

Let $\Sigma \vdash \varphi$: then it immediately follows that $\Sigma \cup \{\neg\varphi\} \vdash \varphi$ and $\Sigma \cup \{\neg\varphi\} \vdash \neg\varphi$, which means that $\Sigma \cup \{\neg\varphi\}$ is inconsistent according to Definition 4.4.5.

Let $\Sigma \cup \{\neg\varphi\}$ *be inconsistent*: with Lemma 4.4.6 we obtain $\Sigma \cup \{\neg\varphi\} \vdash \varphi$. Since φ is a sentence the Deduction Theorem (5.5.1) can be applied yielding $\Sigma \vdash \neg\varphi \to \varphi$. A deduction of $\neg\varphi \to \varphi$ from Σ can be extended by $(\neg\varphi \to \varphi) \to \varphi$ and φ. The first formula is a first-order tautology, the second is obtained by (MP). This proves the assertion. ■

It seems natural that in the relation $\Sigma \cup \{\neg\varphi\} \vdash \varphi$ the formula $\neg\varphi$ is not needed. This was confirmed in the preceding proof by the choice of an appropriate first-order tautology and (MP).

The proof of statement (6.3) is much more involved and takes up almost the remainder of this chapter!

We are going to construct a model for every consistent set of formulae Ψ. To obtain (6.3) we can then set $\Psi := \Sigma \cup \{\neg\varphi\}$.)

We start with a consistent set of $\mathcal{L}$-sentences Σ, with $\mathcal{L} = (\lambda, \mu, K)$ being a given formal language. In order to construct a model for Σ we extend Σ in two steps such that each time the extension remains consistent. The final extension Σ^* has additional properties that can be exploited to construct a model of Σ^*. From the model of Σ^* we will obtain a model of Σ.

The extension steps are quite technical and appear, when encountered for the first time, obscure. They become more intelligible when the model is actually constructed.

6.2 Introducing Witnesses

In the first step Σ is extended to Σ' such that there are witnesses for all existential sentences deducible from Σ. That is, if $\Sigma \vdash \exists v\varphi$ with $\mathrm{Fr}(\varphi) = \{v\}$, then there is a constant symbol c with $\Sigma \vdash \varphi(v/c)$. This requires that $\mathcal{L}$ is extended by constant symbols.

The following lemma concerns the extensions of a formal language by a single constant symbol.

Lemma 6.2.1. *Let $\mathcal{L}_1 = (\lambda_1, \mu_1, K_1)$ and $\mathcal{L}_2 = (\lambda_2, \mu_2, K_2)$ be two formal languages such that $\lambda_1 = \lambda_2$, $\mu_1 = \mu_2$ and $K_2 = K_1 \cup \{0\}$, $0 \notin K_1$. Further, let $D = (\varphi_1, \ldots, \varphi_n)$ be a deduction in $\mathcal{L}_2$ from the set $\Sigma = \{\varphi_1, \ldots, \varphi_m\}$, $m \leq n$[1]. Let v be a variable that does not occur in any of the φ_i. Then $(\varphi_1(c_0/v), \ldots, \varphi_n(c_0/v))$ is a deduction in $\mathcal{L}_1$ from $\{\varphi_1(c_0/v), \ldots, \varphi_m(c_0/v)\}$.*

Proof: We prove the assertion by induction on n, the length of the deduction. We can assume $m < n$, since otherwise there is nothing to show.

$n = 1$, *that is* $D = (\varphi_1)$: This means $m = 0$ and φ_1 is a logical axiom.

If φ_1 is an identity axiom, there is nothing to show, since identity axioms do not contain constants, thus $\varphi_1(c_0/v) = \varphi_1$.

If φ_1 is a first-order tautology, then also $\varphi_1(c_0/v)$.

The remaining possibility is that φ_1 is a quantifier axiom, that is,

$$\varphi_1 = \forall x\psi \to \psi(x/t),$$

with t free for x in ψ. Induction on the complexity of terms and of formulae yields:

$$\psi(x/t)(c_0/v) = \psi(c_0/v)(x/t(c_0/v)) \qquad \text{(cf. Exercise 8.4.1)}$$

[1] That is, the non-logical axioms used in the deduction are listed at the beginning. $m = n$ means that the deduction just consists of non-logical axioms.

The term $t(c_0/v)$ is free for x in $\psi(c_0/v)$, since v is supposed not to occur in φ_1. Therefore, $\varphi_1(c_0/v)$ is also a quantifier axiom, namely

$$\forall x\psi(c_0/v) \to \psi(c_0/v)(x/t(c_0/v)).$$

$n \to n+1$, *that is there is a deduction* $D = (\varphi_1, \ldots, \varphi_{n+1})$: according to the induction hypothesis $(\varphi_1(c_0/v), \ldots, \varphi_n(c_0/v))$ is a deduction in $\mathcal{L}_1$. If φ_{n+1} is a logical axiom then, as shown above, $\varphi_{n+1}(c_0/v)$ is a logical axiom as well. Therefore, the two cases where φ_{n+1} was obtained by the application of an inference rule remain to be looked at.

1. φ_{n+1} was obtained by (MP). This means there are $i, j \leq n$ such that φ_j is $\varphi_i \to \varphi_{n+1}$. This implies that $\varphi_j(c_0/v)$ is $\varphi_i(c_0/v) \to \varphi_{n+1}(c_0/v)$. Thus, $\varphi_{n+1}(c_0/v)$ is also obtained by (MP).

2. φ_{n+1} was obtained by (res$\forall$). Hence there is $i \leq n$ with $\varphi_i = \varphi \to \psi$ and $\varphi_{n+1} = \varphi \to \forall x\psi$, $x \notin \mathrm{Fr}(\varphi)$. As v is different from x, the inference rule (res$\forall$) can be applied to $\varphi_i(c_0/v)$, yielding $\varphi(c_0/v) \to \forall x\psi(c_0/v)$, which is $\varphi_{n+1}(c_0/v)$.

Notice that substituting c_0 in an $\mathcal{L}_2$-formula by a variable results in an $\mathcal{L}_1$-formula. The resulting deduction is therefore in $\mathcal{L}_1$. ■

Theorem 6.2.2. *Let* $\mathcal{L} = (\lambda, \mu, K)$ *be a formal language and* Σ *a consistent set of* $\mathcal{L}$*–sentences. Then there is a formal language* $\mathcal{L}' \supseteq \mathcal{L}$ *with* $I' = I$*,* $J' = J$ *and a consistent set* Σ' *of* $\mathcal{L}'$*–sentences with*

(i) $\Sigma \subset \Sigma'$ *and*

(ii) for every $\mathcal{L}'$*-sentence* $\exists v\varphi$ *there exists* $k \in K'$ *such that*

$$\exists v\varphi \to \varphi(v/c_k) \in \Sigma'. \tag{6.4}$$

Proof: We will obtain $\mathcal{L}'$ and Σ' by a countable sequence of extensions. For $n \in \mathbb{N}$ we define a formal language $\mathcal{L}_n$ recursively as follows: $\mathcal{L}_0$ is $\mathcal{L}$. If $\mathcal{L}_n$ is already defined we obtain $\mathcal{L}_{n+1}$ by setting $\lambda_{n+1} = \lambda_n$, $\mu_{n+1} = \mu_n$ and $K_{n+1} = K_n \cup M_{n+1}$. Here M_{n+1} is a set disjoint from K_n such that there is a bijection

$$g_{n+1} : M_{n+1} \to \{\exists v\varphi | \exists v\varphi \in \mathrm{Sen}(\mathcal{L}_n)\}$$

from M_{n+1} to the set of all existential sentences in $\mathcal{L}_n$. This means that we number one-to-one the existential sentences of $\mathcal{L}_n$ with "new" indices. Such sets M_n and bijection g_n always exist[2].

This yields an ascending chain

$$\mathcal{L}_0 \subset \mathcal{L}_1 \subset \ldots \subset \mathcal{L}_n \subset \mathcal{L}_{n+1} \subset \ldots$$

of formal languages. Finally, we define

$$\mathcal{L}' = \bigcup_{n\in\mathbb{N}} \mathcal{L}_n,$$

that is we set $\lambda' = \lambda$, $\mu' = \mu$ and $K' = \bigcup_{n\in\mathbb{N}} K_n$. Since sentences are finite strings we have

$$\mathrm{Sen}(\mathcal{L}') = \bigcup_{n\in\mathbb{N}} \mathrm{Sen}(\mathcal{L}_n). \tag{6.5}$$

If Σ' is defined as the union of an ascending sequence

$$\Sigma = \Sigma_0 \subset \Sigma_1 \subset \ldots \subset \Sigma_n \subset \Sigma_{n+1} \subset \ldots$$

such that for every $\exists v\varphi \in \mathrm{Sen}(\mathcal{L}_n)$ there is $k \in M_{n+1}$ with

$$\exists v\varphi \to \varphi(v/c_k) \in \Sigma_{n+1}, \tag{6.6}$$

then (6.4) is satisfied: if $\exists v\varphi$ is in $\mathrm{Sen}(\mathcal{L}')$, then (6.5) implies that there is $n \in \mathbb{N}$ with $\exists v\varphi \in \mathrm{Sen}(\mathcal{L}_n)$. With (6.6) it follows that there is a $k \in M_{n+1}$ with $\exists v\varphi \to \varphi(v/c_k) \in \Sigma_{n+1} \subset \Sigma'$. This is obtained by setting

$$\Sigma_{n+1} := \Sigma_n \cup \{\exists v\varphi \to \varphi(v/c_k) \,|\, k \in M_{n+1},\, g_{n+1}(k) = \exists v\varphi\} :$$

the surjectivity of g_{n+1} implies that for every $\exists v\varphi \in \mathrm{Sen}(\mathcal{L}_n)$ there is a $k \in M_{n+1}$ with $g_{n+1}(k) = \exists v\varphi$. Since $\exists v\varphi$ is a sentence $\mathrm{Fr}(\varphi) \subset \{v\}$ and therefore, $\varphi(v/c_k)$ and thus $\exists v\varphi \to \varphi(v/c_k)$ are sentences too, which means $\Sigma_{n+1} \subset \mathrm{Sen}(\mathcal{L}_{n+1})$.

The proof is finished if we can show the consistency of

$$\Sigma' := \bigcup_{n\in\mathbb{N}} \Sigma_n.$$

[2] The metamathematical frame we use to establish results about our formal system is strong enough.

This amounts to showing that every Σ_n is consistent: if Σ' is inconsistent, there is a deduction D of $\perp$ from Σ'. Since deductions are finite and the sequence $(\Sigma_n)_{n\in\mathbb{N}}$ is an increasing sequence with $\Sigma_n \subset \mathrm{Sen}(\mathcal{L}_n)$, we have that for some $n_0 \in \mathbb{N}$ D is a deduction from Σ_{n_0} in $\mathcal{L}_{n_0}$ contradicting the consistency of Σ_{n_0}.

We proceed by induction on n. For $n = 0$ the consistency of Σ_0 is a premise of the theorem since $\Sigma_0 = \Sigma$. We now assume that Σ_n is consistent but not Σ_{n+1}. This means there is $\alpha \in \mathrm{Sen}(\mathcal{L}_{n+1})$ with

$$\Sigma_{n+1} \vdash \alpha \wedge \neg\alpha. \tag{6.7}$$

As the deduction from Σ_{n+1} is finite, there is a deduction $\alpha \wedge \neg\alpha$ from Σ_n together with finitely many sentences

$$\exists v_1\varphi_1 \to \varphi_1(v_1/c_{k_1}), \ldots, \exists v_r\varphi_r \to \varphi_r(v_r/c_{k_r})$$

from Σ_{n+1} where $g_{n+1}(k_i) = \exists v_i\varphi_i \in \mathrm{Sen}(\mathcal{L}_n)$ for $1 \le i \le r$. We denote these r sentences with $\sigma_1, \ldots, \sigma_r$ to shorten the notation.

Thus we have

$$\Sigma_n \cup \{\sigma_1, \ldots, \sigma_r\} \vdash \alpha \wedge \neg\alpha, \tag{6.8}$$

with a deduction in the sublanguage $\mathcal{L}^1$ of $\mathcal{L}_{n+1}$, where $\lambda_{\mathcal{L}^1} = \lambda_{\mathcal{L}_n}$, $\mu_{\mathcal{L}^1} = \mu_{\mathcal{L}_n}$ and $K_{\mathcal{L}^1} = K_{\mathcal{L}_n} \cup \{k_1, \ldots, k_r\}$.

Our goal is now to show that already the reduced set $\Sigma_n \cup \{\sigma_2, \ldots, \sigma_r\}$ is inconsistent. With the Deduction Theorem 5.5.1 relation (6.8) implies

$$\Sigma_n \cup \{\sigma_2, \ldots, \sigma_r\} \vdash \sigma_1 \to \alpha \wedge \neg\alpha. \tag{6.9}$$

Setting $P_0 := \exists v_1\varphi_1$, $P_1 := \varphi_1(v_1/c_{k_1})$ and $P_2 := \alpha$ in the first-order tautology

$$((P_0 \to P_1) \to (P_2 \wedge \neg P_2)) \to P_0 \wedge \neg P_1$$

(6.9) implies with (MP)

$$\Sigma_n \cup \{\sigma_2, \ldots, \sigma_r\} \vdash \exists v_1\varphi_1 \wedge \neg\varphi_1(v_1/c_{k_1}). \tag{6.10}$$

Taking into account that $\exists v_1$ is an abbreviation for $\neg\forall v_1\neg$ we obtain on one hand, applying the derived rule ($\wedge B_1$),

$$\Sigma_n \cup \{\sigma_2, \ldots, \sigma_r\} \vdash \neg\forall v_1\neg\varphi_1. \tag{6.11}$$

On the other hand, we obtain applying ($\wedge B_2$),

$$\Sigma_n \cup \{\sigma_2, \ldots, \sigma_r\} \vdash \neg\varphi_1(v_1/c_{k_1}), \tag{6.12}$$

where the deductions given by (6.11) and (6.12) are in $\mathcal{L}^1$.

If we define the sublanguage $\mathcal{L}^2$ of $\mathcal{L}^1$ by setting $\lambda_{\mathcal{L}^2} = \lambda_{\mathcal{L}^1}$, $\mu_{\mathcal{L}^2} = \mu_{\mathcal{L}^1}$ and $K_{\mathcal{L}^2} = K_{\mathcal{L}_n} \cup \{k_2, \ldots, k_r\}$, then we see that $\neg\forall v_1 \neg\varphi_1$ and $\Sigma \cup \{\sigma_2, \ldots, \sigma_r\}$ are already in $\mathrm{Sen}(\mathcal{L}^2)$, which is then also the case for the deduction in (6.11).

Applying Lemma 6.2.1 to a deduction in (6.12) yields

$$\Sigma_n \cup \{\sigma_2, \ldots, \sigma_r\} \vdash \neg\varphi_1(v_1/c_{k_1})(c_{k_1}/y), \tag{6.13}$$

where y is a variable that does not occur in the deduction of $\neg\varphi(v_1/c_{k_1})$ from $\Sigma_n \cup \{\sigma_2, \ldots, \sigma_r\}$ to which Lemma 6.2.1 is applied. $\Sigma_n \cup \{\sigma_2, \ldots, \sigma_r\}$ remains unchanged since c_{k_1} does not occur in it. Therefore, the deduction given by relation (6.13) is a deduction in in $\mathcal{L}^2$.

Since $\varphi_1(v_1/c_{k_1})(c_{k_1}/y) = \varphi_1(v_1/y)$ we obtain

$$\Sigma_n \cup \{\sigma_2, \ldots, \sigma_r\} \vdash \neg\varphi_1(v_1/y).$$

Applying (res$\forall$) and (S) we first obtain

$$\Sigma_n \cup \{\sigma_2, \ldots, \sigma_r\} \vdash \forall y \neg\varphi_1(v_1/y)$$

and then

$$\Sigma_n \cup \{\sigma_2, \ldots, \sigma_r\} \vdash \neg\varphi_1(v_1/y)(y/v_1).$$

To apply (S) v_1 has to be free for y in $\varphi_1(v_1/y)$. However, this is the case since in $\varphi_1(v_1/y)$ the variable v_1 is only substituted by y if v_1 is not in the scope of a quantifier $\forall v_1$.

Since, according to Lemma 6.2.1, y was chosen such that it did not occur in $\varphi_1(v_1/c_{k_1})$ and we further assume without loss of generality that $y \neq v_1$, the equation

$$\varphi(v_1/y)(y/v_1) = \varphi_1$$

holds. Thus $\Sigma_n \cup \{\sigma_2, \ldots, \sigma_r\} \vdash \neg\varphi_1(v_1/y)$, which yields by the application of (res$\forall$)

$$\Sigma_n \cup \{\sigma_2, \ldots, \sigma_r\} \vdash \forall v_1 \neg\varphi_1, \tag{6.14}$$

(6.11) and (6.14) show that the set $\Sigma_n \cup \{\sigma_2, \ldots, \sigma_r\}$ is inconsistent in $\mathcal{L}^2$.

In the same way we reduced the inconsistency of $\Sigma_n \cup \{\sigma_1, \ldots, \sigma_r\}$ to the inconsistency of $\Sigma_n \cup \{\sigma_2, \ldots, \sigma_r\}$ we arrive by iteration at the inconsistency of Σ_n, and in fact in $\mathcal{L}$ since in each step a constant k_i is removed. ■

Exercise 6.2.3. In the extension steps in the proof of Theorem 6.2.2, why were different witnesses chosen for every existential sentence? (Instead of choosing a bijection g_{n+1} defined on the existential sentences of $\mathcal{L}_n$ one could try to extend $\mathcal{L}_n$ by a single constant symbol as a witness for all existential sentences of $\mathcal{L}_n$.)

6.3 Maximal Consistent Set Of Sentences

The second step will take place in the extension $\mathcal{L}'$ that results from the execution of the first step. We use the following theorem, which is formulated for an arbitrary language $\mathcal{L}$.

Theorem 6.3.1. *Let $\mathcal{L}$ be a formal language and $\Sigma \subset Sen(\mathcal{L})$ an arbitrary consistent set of sentences. Then there is a maximal consistent superset $\Sigma^* \subset Sen(\mathcal{L})$ of Σ, that is $\Sigma \subset \Sigma^*$, Σ^* consistent and if $\Sigma^* \subset \Sigma_1 \subset Sen(\mathcal{L})$ and Σ_1 consistent then $\Sigma^* = \Sigma_1$.*

Proof: Let us consider the collection

$$\mathcal{M} := \{\Sigma_1 \subset \mathrm{Sen}(\mathcal{L}) \,|\, \Sigma \subset \Sigma_1\,,\, \Sigma_1 \text{ consistent}\}.$$

Let $\mathcal{M}'$ be a subset of $\mathcal{M}$ such that for any $\Sigma_1, \Sigma_1' \in \mathcal{M}$ holds either $\Sigma_1 \subset \Sigma_1'$ or $\Sigma_1' \subset \Sigma_1$.

If $\mathcal{M}'$ is empty, then Σ is an upper bound of $\mathcal{M}'$ in $\mathcal{M}$, otherwise

$$\Sigma' := \bigcup_{\Sigma_1 \in \mathcal{M}'} \Sigma_1$$

is an upper bound of $\mathcal{M}'$ in $\mathcal{M}$:

Obviously, $\Sigma_1 \subset \Sigma'$ holds for every Σ_1 in $\mathcal{M}'$. The consistency of Σ' follows from the fact that only finitely many elements $\varphi_1, \ldots, \varphi_n \in \Sigma'$ can occur in a deduction D of from Σ'. Each φ_i is in a $\Sigma_i \in \mathcal{M}'$. Since the sets $\Sigma_1, \ldots, \Sigma_n$ are comparable with regard to set inclusion $\subset$ there is a Σ_{i_0} that contains all φ_i. Since Σ_{i_0} is consistent D cannot deduce a contradiction.

This shows that $\mathcal{M}$ satisfies the conditions of Zorn's lemma. Therefore $\mathcal{M}$ contains a maximal element Σ^*. According to the definition of $\mathcal{M}$ we see that $\Sigma \subset \Sigma^*$ and that Σ^* is consistent. ■

Before moving on we need some auxiliary observations on maximal consistent sets of sentences and witnesses. The first lemma is stated for an arbitrary formal language $\mathcal{L}$.

Lemma 6.3.2. *Let $\mathcal{L}$ be a formal language and $\Sigma \subset Sen(\mathcal{L})$. If Σ is maximal consistent, then Σ is deductively closed, that is, for all $\varphi \in Sen(\mathcal{L})$*

$$\Sigma \vdash \varphi \textit{ implies } \varphi \in \Sigma.$$

Proof: We show for an arbitrary $\varphi \in \mathrm{Sen}(\mathcal{L})$ that $\Sigma \vdash \varphi$ implies the consistency of $\Sigma \cup \{\varphi\}$. Since Σ is maximal consistent we then obtain $\varphi \in \Sigma$.

So assume $\Sigma \vdash \varphi$ but $\Sigma \cup \{\varphi\}$ inconsistent. The latter implies

$$\Sigma \cup \{\varphi\} \vdash \neg\varphi,$$

thus

$$\begin{aligned}
\Sigma \vdash& \varphi \to \neg\varphi && \text{(Ded)}\\
\Sigma \vdash& (\varphi \to \neg\varphi) \to \neg\varphi && \text{(first-order tautology)}\\
\Sigma \vdash& \neg\varphi && \text{(MP,)}
\end{aligned}$$

hence $\Sigma \vdash \neg\varphi$, contradicting, together with $\Sigma \vdash \varphi$, the consistency of Σ. ■

In the following the set of constant terms $\mathrm{CT}_{\mathcal{L}}$ of any formal language $\mathcal{L}$ will be crucial.

Definition 6.3.3. Let $\mathcal{L}$ be a formal language. We set

$$\mathrm{CT}_{\mathcal{L}} := \{t \in \mathrm{Tm}(\mathcal{L}) \mid t \text{ contains no variable}\}.$$

Also the next lemma is stated for an arbitrary formal language $\mathcal{L}$. Consider the following properties of a subset $\Sigma \subseteq \mathrm{Sen}(\mathcal{L})$:

(I) Σ is maximal consistent in $\mathrm{Sen}(\mathcal{L})$

(II) For every $\exists v\varphi \in \mathrm{Sen}(\mathcal{L})$ there is a $k \in K$ such that

$$\exists v\varphi \to \varphi(v/c_k) \in \Sigma.$$

Lemma 6.3.4. *Let $\mathcal{L}$ be a formal language and $\Sigma \subset Sen(\mathcal{L})$ with properties (I) and (II). Then for all φ, ψ, $\forall v\alpha \in Sen(\mathcal{L})$ the following holds:*

(1)	$\neg\varphi \in \Sigma$	*iff*	$\varphi \notin \Sigma$
(2)	$\varphi \wedge \psi \in \Sigma$	*iff*	$\varphi \in \Sigma$ *and* $\psi \in \Sigma$
(3)	$\forall v\alpha \in \Sigma$	*iff*	$\alpha(v/t) \in \Sigma$ *for all* $t \in CT_{\mathcal{L}}$

Proof: (1) If $\neg\varphi \in \Sigma$ then $\varphi \notin \Sigma$ since Σ is consistent. On the other hand let us assume $\varphi \notin \Sigma$. Lemma 6.3.2 implies $\Sigma \nvdash \varphi$, from which the consistency of $\Sigma \cup \{\neg\varphi\}$ follows by Lemma 6.1.1. Since Σ is maximal consistent we obtain $\Sigma = \Sigma \cup \{\neg\varphi\}$, thus $\neg\varphi \in \Sigma$.

(2) Obviously, $\varphi \wedge \psi \in \Sigma$ implies $\Sigma \vdash \varphi \wedge \psi$ from which, by the derived rules ($\wedge B_1$) and ($\wedge B_2$), it follows that $\Sigma \vdash \varphi$ and $\Sigma \vdash \psi$. The assertion follows with Lemma 6.3.2. The other direction is proved by applying the derived rule ($\wedge$).

(3) First, let us assume $\forall v\alpha \in \Sigma$. Applying the rule (S) yields $\Sigma \vdash \alpha(v/t)$ for $t \in \mathrm{CT}_{\mathcal{L}}$ because every term with no variables is free for any variable. With Lemma 6.3.2 it follows that $\alpha(v/t) \in \Sigma$.

To show the converse we assume $\forall v\alpha \notin \Sigma$. (1) implies

$$\neg\forall v\alpha \in \Sigma. \tag{6.15}$$

We want to show that there is $t \in \mathrm{CT}_{\mathcal{L}}$ with $\alpha(v/t) \notin \Sigma$.

Since $\neg\neg\alpha \to \alpha$ is a first-order tautology we have $\Sigma \vdash \neg\neg\alpha \to \alpha$, which implies $\Sigma \cup \{\neg\neg\alpha\} \vdash \alpha$, and thus by Lemma 5.4.1,

$$\Sigma \cup \{\forall v\neg\neg\alpha\} \vdash \forall v\alpha. \tag{6.16}$$

As $\forall v\alpha \in \mathrm{Sen}(\mathcal{L})$ the formula $\forall v\neg\neg\alpha$ is also a sentence, therefore, the Deduction Theorem 5.5.1 can be applied to (6.16) yielding

$$\Sigma \vdash \forall v\neg\neg\alpha \to \forall v\alpha,$$

from which with the derived rule (CP) we obtain

$$\Sigma \vdash \neg\forall v\alpha \to \neg\forall v\neg\neg\alpha. \tag{6.17}$$

(6.17) and (6.15) imply $\Sigma \vdash \neg\forall v\neg\neg\alpha$, which can be abbreviated to

$$\Sigma \vdash \exists v\neg\alpha. \tag{6.18}$$

According to property (II) of Σ there is $t \in \mathrm{CT}_{\mathcal{L}}$ such that

$$\Sigma \vdash \exists v \neg \alpha \to \neg \alpha(v/t). \tag{6.19}$$

(6.18) and (6.19) together with (MP) yield $\Sigma \vdash \neg\alpha(v/t)$ from which

$$\neg\alpha(v/t) \in \Sigma \tag{6.20}$$

follows with Lemma 6.3.2. (6.20) implies $\alpha(v/t) \notin \Sigma$, using (1) (or the consistency of Σ), which is what we wanted to show. ■

6.4 The Term Structure

In Section 6.3 it was shown that given a formal language $\mathcal{L}$ and a consistent set of sentences $\Sigma \subset \mathrm{Sen}(\mathcal{L})$, we may obtain by applying theorems 6.2.2 and 6.3.1 successively, first a formal language $\mathcal{L}'$ that is an extension of $\mathcal{L}$ with $\lambda_{\mathcal{L}'} = \lambda_{\mathcal{L}}$, $\mu_{\mathcal{L}'} = \mu_{\mathcal{L}}$, and then a maximal consistent set of sentences $\Sigma^* \subset \mathrm{Sen}(\mathcal{L}')$ with $\Sigma \subset \Sigma^*$.

Σ^* has the following properties:

(I) Σ^* is maximal consistent in $\mathrm{Sen}(\mathcal{L}')$

(II) For every $\exists v\varphi \in \mathrm{Sen}(\mathcal{L}')$ there is a $k \in K'$ such that

$$\exists v\varphi \to \varphi(v/c_k) \in \Sigma^*.$$

We will construct, for any set of sentences Σ^* with properties (I) and (II), an $\mathcal{L}'$-structure $\mathcal{A}_{\Sigma^*}$ in which exactly the elements of Σ^* are valid. Forgetting the interpretations of the elements in $\{c_k \,|\, k \in K' \setminus K\}$ in $|\mathcal{A}_{\Sigma^*}|$ yields an $\mathcal{L}$-structure $\mathcal{A}_\Sigma$. Since $\Sigma \subset \Sigma^*$ this structure is a model of Σ.

The following lemma describes how one can define, in a canonical way, an $\mathcal{L}$-structure $\mathcal{A}_\Sigma$ from $\mathrm{CT}_{\mathcal{L}}$ given a set of $\mathcal{L}$-sentences Σ. The structure obtained is similar to a Henkin structure (example 3.1.3). Σ is used to identify certain constant terms.

Lemma 6.4.1. *Let $\mathcal{L}$ be a formal language and $\Sigma \subset$ Sen$(\mathcal{L})$ a set of sentences. Then the following are true:*

1. *The binary relation $\approx$ on $CT_{\mathcal{L}}$ defined by*

$$t_1 \approx t_2 \qquad \textit{iff} \qquad \Sigma \vdash t_1 \doteq t_2 \tag{6.21}$$

is an equivalence relation, that is, for all $t_1, t_2, t_3 \in CT_{\mathcal{L}}$ we have

   (a) *$t_1 \approx t_1$ (reflexivity),*

   (b) *if $t_1 \approx t_2$, then $t_2 \approx t_1$ (symmetry),*

   (c) *if $t_1 \approx t_2$ and $t_2 \approx t_3$, then $t_1 \approx t_3$ (transitivity).*

2. *We obtain an $\mathcal{L}$-structure $\mathcal{A}_\Sigma$ by setting:*

   (i) *The universe of $\mathcal{A}_\Sigma$*

$$|\mathcal{A}_\Sigma| := CT_{\mathcal{L}}/\approx,$$

   the set of all equivalence classes $[t] := \{t_1 \in CT_{\mathcal{L}} \,|\, t \approx t_1\}$ of elements of $CT_{\mathcal{L}}$.

   (ii) *For every $i \in I$ and equivalence classes $[t_1], \ldots, [t_{\lambda(i)}]$*

$$R_i^{\mathcal{A}_\Sigma}([t_1], \ldots, [t_{\lambda(i)}]) \qquad \textit{iff} \qquad \Sigma \vdash R_i(t_1, \ldots, t_{\lambda(i)}).$$

   (iii) *For every $j \in J$ and equivalence classes $[t_1], \ldots, [t_{\mu(i)}]$*

$$f_j^{\mathcal{A}_\Sigma}([t_1], \ldots, [t_\mu(j)]) := [f_j(t_1, \ldots, t_{\mu(j)})].$$

   (iv) *For every $k \in K$ $c_k^{\mathcal{A}_\Sigma} := [c_k]$.*

Proof:

1. (a) Apply axiom 4.2.2 (I1) and the derived rules (res$\forall$) and (S).

   (b) Follows immediately from (Sym).

   (c) Follows immediately from (Tr).

2. We have to show that $R_i^{\mathcal{A}_\Sigma}$ and $f_j^{\mathcal{A}_\Sigma}$ are well-defined, that is, they do not depend on the representatives of the equivalence classes $[t_i]$. Notice that

$$[t_1] = [t_2] \qquad \text{iff} \qquad t_1 \approx t_2.$$

   To show that $R_i^{\mathcal{A}_\Sigma}$ does not depend on the representatives let us assume

$$t_1 \approx t_1', \ldots, t_{\lambda(i)} \approx t_{\lambda(i)}',$$

which amounts to

$$\Sigma \vdash t_\nu \doteq t'_\nu \qquad \text{for } 1 \le \nu \le \lambda(i). \tag{6.22}$$

We have to show

$$R_i^{\mathcal{A}_\Sigma}([t_1], \ldots, [t_{\lambda(i)}]) \quad \text{iff} \quad R_i^{\mathcal{A}_\Sigma}([t'_1], \ldots, [t'_{\lambda(i)}]),$$

which is equivalent to

$$\Sigma \vdash R_i(t_1, \ldots, t_{\lambda(i)}) \quad \text{iff} \quad \Sigma \vdash R_i(t'_1, \ldots, t'_{\lambda(i)}).$$

Suppose

$$\Sigma \vdash R_i(t_1, \ldots, t_{\lambda(i)}). \tag{6.23}$$

Combining deductions of (6.23) and (6.22) and applying the derived rules (R_i) and (MP) enables us to successively substitute every t_ν by t'_ν in (6.23), resulting in

$$\Sigma \vdash R_i(t'_1, \ldots, t'_\lambda).$$

The other direction follows by symmetry.

To show that $f_j^{\mathcal{A}_\Sigma}$ does not depend on the representatives one has to show that

$$t_1 \approx t'_1, \ldots, t_{\mu(i)} \approx t'_{\mu(i)},$$

together imply

$$[f(t_1, \ldots, t_{\mu(j)})] = [f(t'_1, \ldots, t'_{\mu(j)})].$$

That is

$$\Sigma \vdash t_\nu \doteq t'_\nu \qquad \text{for } 1 \le \nu \le \mu(i),$$

must be shown to imply

$$\Sigma \vdash f(t_1, \ldots, t_{\mu(j)}) = f(t'_1, \ldots, t'_{\mu(j)}).$$

This is achieved by proceeding as in the case for $R_i^{\mathcal{A}}$, using the derived rules (f_j) and (Tr). ■

Solve the exercises of Section 8.8!

6.5 The Completeness Theorem

Now, we have at our disposal all the means to prove the Completeness Theorem.

Theorem 6.5.1 (Completeness). *Let $\mathcal{L}$ be a formal language, $\Sigma \subset Sen(\mathcal{L})$ a set of sentences and $\varphi \in Sen(\mathcal{L})$. If $\Sigma \models \varphi$ then $\Sigma \vdash \varphi$.*

Proof: As described in Section 6.1 it suffices to show that every consistent set of sentences Σ possesses a model.

Suppose $\Sigma \subset \mathrm{Sen}(\mathcal{L})$ consistent. Let $\Sigma^* \subset \mathrm{Sen}(\mathcal{L}')$ be the set of sentences satisfying (I) and (II) obtained by Theorems 6.2.2 and 6.3.1, where $\mathcal{L}'$ is the extension of $\mathcal{L}$ by the constant symbols provided by Theorem 6.2.2.

If we can show that for the $\mathcal{L}'$-structure $\mathcal{A}_{\Sigma^*}$ obtained with Lemma 6.4.1, all $\varphi \in \mathrm{Sen}(\mathcal{L})$ and all variable assignments h with values in $\mathcal{A}_{\Sigma^*}$ we have

$$\mathcal{A}_{\Sigma^*} \models \varphi[h] \qquad \text{iff} \qquad \varphi \in \Sigma^*, \tag{6.24}$$

then we have established that $\mathcal{A}_{\Sigma^*}$ is a model of Σ'.[3]

Before proving (6.24) by induction on the number of logical symbols $\neg$, $\wedge$ and $\forall$ that occur in φ we show

$$\overline{h}(t) = [t] \tag{6.25}$$

for all $t \in \mathrm{CT}_{\mathcal{L}'}$ and all variable assignments h with values in $\mathcal{A}_{\Sigma^*}$. $\overline{h}$ is the term assignment function generated by h (Definition 3.2.3). By induction on the definition of a term we obtain

$$\begin{aligned}
\overline{h}(f_j(t_1,\ldots,t_{\mu(j)})) &= f_j^{\mathcal{A}_{\Sigma^*}}(\overline{h}(t_1),\ldots,\overline{h}(t_{\mu(j)})) \\
&= f_j^{\mathcal{A}_{\Sigma^*}}([t_1],\ldots,[t_{\mu(j)}]) \\
&= [f_j(t_1,\ldots,t_{\mu(j)})].
\end{aligned}$$

Let now the above number of logical symbols be 0. This means that φ is an

[3]Notice the difference between applying square brackets to a variable assignment function h and to a term t!

atomic sentence. We have the following cases:

$$\begin{aligned}
\mathcal{A}_{\Sigma^*} \models t_1 \doteq t_2[h] &\iff \overline{h}(t_1) = \overline{h}(t_2) && \text{(Definition 3.2.6)}\\
&\iff [t_1] = [t_2] && \text{(Equation (6.25))}\\
&\iff \Sigma^* \vdash t_1 \doteq t_2 && \text{(definition of } \mathcal{A}_{\Sigma^*}\text{)}\\
&\iff t_1 \doteq t_2 \in \Sigma^* && \text{(Lemma 6.3.2)}
\end{aligned}$$

$$\begin{aligned}
\mathcal{A}_{\Sigma^*} \models R_i(t_1, \ldots, t_{\lambda(i)})[h] &\iff R_i^{\mathcal{A}_{\Sigma^*}}(\overline{h}(t_1), \ldots, \overline{h}(t_{\lambda(i)}))\\
& && \text{(Definition 3.2.6)}\\
&\iff R_i^{\mathcal{A}_{\Sigma^*}}([t_1], \ldots, [t_{\lambda(i)}])\\
& && \text{(Equation (6.25))}\\
&\iff \Sigma^* \vdash R_i(t_1, \ldots, t_{\lambda(i)})\\
& && \text{(definition of } \mathcal{A}_{\Sigma^*}\text{)}\\
&\iff R_i(t_1, \ldots, t_{\lambda(i)}) \in \Sigma^* && \text{(Lemma 6.3.2)}
\end{aligned}$$

Let φ now be $\neg\alpha$ respectively $\alpha \wedge \beta$. Then we obtain

$$\begin{aligned}
\mathcal{A}_{\Sigma^*} \models \neg\alpha[h] &\iff \mathcal{A}_{\Sigma^*} \not\models \alpha[h] && \text{(Definition 3.2.6)}\\
&\iff \alpha \notin \Sigma^* && \text{(induction hypothesis)}\\
&\iff \neg\alpha \in \Sigma^* && \text{(Lemma 6.3.4)}
\end{aligned}$$

and

$$\begin{aligned}
\mathcal{A}_{\Sigma^*} \models \alpha \wedge \beta[h] &\iff \mathcal{A}_{\Sigma^*} \models \alpha[h] \text{ and } \mathcal{A}_{\Sigma^*} \models \beta[h] && \text{(Definition 3.2.6)}\\
&\iff \alpha \in \Sigma^* \text{ and } \beta \in \Sigma^* && \text{(induction hypothesis)}\\
&\iff \alpha \wedge \beta \in \Sigma^* && \text{(Lemma 6.3.4)}
\end{aligned}$$

Finally, we have to consider $\varphi = \forall v\alpha$. As φ is a sentence we have $\mathrm{Fr}(\varphi) \subset \{v\}$. Therefore $\alpha(v/t)$, $t \in \mathrm{CT}_{\mathcal{L}'}$, is again a sentence, in fact one with fewer logical symbols.

$$
\begin{aligned}
\mathcal{A}_{\Sigma^*} \models \forall v \alpha[h] \quad &\iff \quad \mathcal{A}_{\Sigma^*} \models \alpha[h\binom{v}{a}] \text{ for all } a \in |\mathcal{A}_{\Sigma^*}| \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{(Definition 3.2.6)} \\
&\iff \quad \mathcal{A}_{\Sigma^*} \models \alpha[h\binom{v}{[t]}] \text{ for all } t \in \mathrm{CT}_{\mathcal{L}'} \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{(definition of } \mathcal{A}_{\Sigma^*}) \\
&\iff \quad \mathcal{A}_{\Sigma^*} \models \alpha(v/t)[h] \text{ for all } t \in \mathrm{CT}_{\mathcal{L}'} \quad \text{(Lemma 5.1.2)} \\
&\iff \quad \alpha(v/t) \in \Sigma^* \text{ for all } t \in \mathrm{CT}_{\mathcal{L}'} \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{(induction hypothesis)} \\
&\iff \quad \forall v \alpha \in \Sigma^* \qquad\qquad\qquad \text{(Lemma 6.3.4)}
\end{aligned}
$$

From $\mathcal{A}_{\Sigma^*}$ we can obtain a model $\mathcal{A}_{\mathcal{L}}$ of Σ by forgetting the interpretations of the constant symbols $\{c_k \mid k \in K' \setminus K\}$. Obviously all $\mathcal{L}$-sentences $\varphi \in \Sigma$ hold in the $\mathcal{L}$-structure

$$\mathcal{A}_{\mathcal{L}} := \langle |\mathcal{A}_{\Sigma^*}|, (R_i^{\mathcal{A}_{\Sigma^*}})_{i \in I}, (f_j^{\mathcal{A}_{\Sigma^*}})_{j \in J}, (c_k^{\mathcal{A}_{\Sigma^*}})_{k \in K} \rangle,$$

which is therefore a model of Σ. ■

The following corollaries combine correctness and completeness of our deductive system.

Corollary 6.5.2. *Let $\mathcal{L}$ be a formal language, $\Sigma \subset Sen(\mathcal{L})$ and $\varphi \in Fml(\mathcal{L})$. Then*

$$\Sigma \vdash \varphi \qquad \textit{iff} \qquad \Sigma \models \varphi$$

Proof: Combine Theorems 5.1.4 and 6.5.1. ■

Corollary 6.5.3. *Let $\mathcal{L}$ be a formal language, $\Sigma \subset Sen(\mathcal{L})$. Then Σ is consistent if and only if it has a model.*

Proof: If Σ has a model, then correctness implies that Σ is consistent. The other direction was shown in the proof of Theorem 6.5.1. ■

We finish this section with a theorem that we will prove by making use of the finiteness of a deduction. It is also called *Compactness Theorem.*

Theorem 6.5.4 (Finiteness Theorem, Compactness Theorem). *A set of sentences Σ possesses a model if and only if every finite subset of Σ has a model.*

Proof: With Σ obviously every finite subset possesses a model. Let us assume that Σ does not possess a model. According to corollary 6.5.3 Σ is inconsistent. Therefore there is a deduction $D = (\varphi_1, \ldots, \varphi_n)$ of a contradiction $\alpha \wedge \neg\alpha$ from Σ. The finite subset $\{\varphi_1, \ldots, \varphi_n\}$ does not have a model. This proves the converse. ∎

We proved the finiteness theorem using a corollary to the Completeness theorem, thus referring to the notion of deduction, thereby relying on its finiteness. There are also proofs of the finiteness theorem that refer only to the notions formal language and model! Such a proof can be obtained using so-called *ultra products*. We think that the path we chose yields a deeper understanding of the finiteness theorem.

To test your knowledge, solve the multiple choice test of this chapter.

Chapter 7

Herbrand Structures

In Section 5.2 we saw that there is no procedure for deciding whether a formula φ can be deduced from a set of formulae Σ (cf. Theorem 5.2.4). As a corollary (cf. Corollary 5.2.5) we found that a decision procedure for the satisfiability of a formula φ cannot exist.

In this chapter we make the best of a bad job with regard to a decision procedure for satisfiability. In the first section we introduce different normal forms of a first-order formula. We can then assign to a formula φ its so-called *Skolem form* $\mathsf{sf}(\varphi)$ which can be satisfied if and only if φ can be satisfied. The crucial property of a formula in Skolem form is the absence of existential quantifiers. This is exploited in Section 7.3 where we show that the search of a model of a formula in Skolem form can be restricted to so-called *Herbrand models*. By considering the so-called *Herbrand expansion* of a formula in Skolem form we finally obtain an algorithm that stops if a formula is unsatisfiable and runs forever otherwise.

7.1 Normal forms

In this section we consider formulae of an arbitrary first-order language $\mathcal{L}$.

In the first two definitions we introduce normal forms for first-order formulae.

Definition 7.1.1. Let τ and θ be $\mathcal{L}$-formulae. If there are atomic formulae or negated atomic $\mathcal{L}$-formulae τ_{ij}, $1 \leq j \leq m_\tau$, $1 \leq i \leq n_\tau$ and ϑ_{ij}, $1 \leq j \leq$

m_ϑ, $1 \leq i \leq n_\vartheta$ such that

$$\tau = \bigwedge_{i=1}^{n} \bigvee_{j=1}^{m} \tau_{ij},$$

$$\vartheta := \bigvee_{i=1}^{n} \bigwedge_{j=1}^{m} \vartheta_{ij},$$

then τ is said to be in *conjunctive normal form* and ϑ in *disjunctive normal form*.

Definition 7.1.2. Let $Q_i \in \{\forall, \exists\}, 1 \leq i \leq n$, be a sequence of quantifiers. If the formula ψ is in conjunctive normal form then the formula

$$\varphi := Q_1 v_1 Q_2 v_2 \ldots Q_n v_n \psi$$

is said to be in *prenex normal form*.

Example 7.1.3. Let $\mathcal{L}$ be the language introduced in Example 2.1.5. Then the $\mathcal{L}$-formula

$$\forall x \exists y ((x < y \vee x = y \vee x > y) \wedge (x - y \geq 0 \vee y - x > 0))$$

is in prenex normal form.

Definition 7.1.4. A formula φ is said to be *proper* if no variable is at same time free and bound in φ and if all quantified variables of φ are pairwise distinct.

Example 7.1.5. Let $\mathcal{L}$ be the language introduced in Example 2.1.5. Then the $\mathcal{L}$-formula

$$u \doteq 1 \wedge \forall x \exists y (\, x < y \,)$$

is proper, whereas the $\mathcal{L}$-formula

$$x \doteq 1 \wedge \forall x \exists y (\, x < y \,)$$

is not proper.

Definition 7.1.6. Two formulae φ and ψ are said to be *equivalent* if $\vdash \varphi \leftrightarrow \psi$ holds. According to Corollary 6.5.2 the last statement amounts to $\mathcal{A} \models \varphi$ if and only if $\mathcal{A} \models \psi$ for all $\mathcal{L}$-structures $\mathcal{A}$.

Notation: We write $\varphi \approx \psi$ if φ and ψ are equivalent.

The following definition expresses a property that is preserved under Algorithm 1, as will be stated in Theorem 7.1.11.

Definition 7.1.7. Let $\mathcal{L}$ and $\mathcal{L}'$ be formal languages. $\varphi \in \mathrm{Fml}(\mathcal{L})$ and $\psi \in \mathrm{Fml}(\mathcal{L}')$ are said to be *satisfiability equivalent* under the condition that φ is satisfiable if and only if ψ is satisfiable.

Remark 7.1.8. Obviously, if formulae φ and ψ are equivalent then they are also satisfiability equivalent. The converse is not true as the following example illustrates:

Let $\mathcal{L}$ be a formal language with the unary function symbol f as its only non-logical symbol. If we define the $\mathcal{L}$-formulae $\varphi := \forall x \exists y (x \doteq y)$ and $\psi := \forall x (x \doteq f(x))$, then φ and ψ are satisfiability equivalent but not equivalent.

φ is satisfiable since every $\mathcal{L}$-structure is a model for φ. If $|\mathcal{A}|$ is any set and $f^{\mathcal{A}}$ is interpreted as the identity on $|\mathcal{A}|$, then the resulting $\mathcal{L}$-structure $\mathcal{A}$ is a model of ψ. Thus, also ψ has a model. Therefore, φ is satisfiable if and only if ψ is satisfiable.

However, φ and ψ are not equivalent: φ is valid, whereas this is not the case for ψ. If $|\mathcal{A}|$ is a set with at least two elements and $f^{\mathcal{A}} : |\mathcal{A}| \to |\mathcal{A}|$ such that $f(x) \neq x$ for an $x \in |\mathcal{A}|$ then the resulting $\mathcal{L}$-structure $\mathcal{A}$ is not a model of ψ.

The following theorem states that when looking for a satisfiability equivalent formula for a formula φ, the formula φ can be assumed to be in proper prenex normal form.

Theorem 7.1.9. *For every $\mathcal{L}$-formula φ there is an equivalent proper $\mathcal{L}$-formula $\widetilde{\varphi}$ in prenex normal form.*

Proof: The proof is left as Exercise 8.10.1. ■

The following algorithm assigns to an $\mathcal{L}$-formula φ in proper prenex normal form an $\mathcal{L}'$-formula ψ, $\mathcal{L}'$ an extension of $\mathcal{L}$, such that

- φ and ψ are satisfiability equivalent,
- ψ no longer contains any existential quantifiers.

Algorithm 1. [Transformation into Skolem form: $\mathsf{sf}()$]

Input: Formula φ.

Step 0: Transform φ in proper prenex normal form, that is

$$\varphi = \forall v_1 \dots \forall v_n \exists y \psi \qquad \psi \in \mathrm{Fml}(\mathcal{L}),$$

where $\exists y$ is the leftmost existential quantifier in φ. $n = 0$ is possible, which means that if φ contains quantifiers, then its leftmost quantifier is existential. ψ can contain further existential quantifiers.

Step 1: Choose an n-ary function symbol f such that $f \notin \mathcal{L}$.

Step 2: Define the extension $\mathcal{L}' := \mathcal{L} \cup \{f\}$.

Step 3: Define the $\mathcal{L}'$-formula

$$\varphi' := \forall v_1 \dots \forall v_n \psi(y/f(v_1, \dots, v_n)),$$

that is, every occurrence of y is replaced by $f(v_1, \dots, v_n)$.

Step 4: Repeat steps 1 to 3 until there are no existential quantifiers left.

Output: Formula $\mathrm{step}_4(\varphi)$.

In step 1 if $n = 0$ the function symbol f is 0-ary, that is, f is a constant symbol.

In the following we write $\mathsf{sf}(\varphi)$ to denote the formula which is obtained when the algorithm is applied to the formula φ.

Definition 7.1.10. Let φ be an $\mathcal{L}$-formula. The Skolem form of φ is the $\mathcal{L}_1$-formula $\varphi_1 := \mathsf{sf}(\varphi)$, that is, the result of the application of Algorithm 1 to φ. $\mathcal{L}_1$ contains pairwise distinct function symbols that do not belong to $\mathcal{L}$ in place of the existential quantifiers occurring in φ. In φ_1 only universal quantifiers occur.

The following theorem states that a formula and its Skolem form are satisfiability equivalent.

Theorem 7.1.11. *Let φ be formula. Then φ and its Skolem form φ_1 are satisfiability equivalent.*

Proof: Due to Theorem 7.1.9 it suffices to show that $\varphi := \forall v_1 \dots \forall v_n \exists y \psi$ is in proper prenex normal form and $\varphi' := \forall v_1 \dots \forall v_n \psi(y/f(v_1, \dots, v_n))$, which is the formula obtained from φ after one application of Step 1 to Step 3 in Algorithm 1, are satisfiabilty equivalent.

1. *Let φ' be satisfiable.* Then there exists an $\mathcal{L}'$-structure $\mathcal{A}'$ such that $\mathcal{A}' \models \varphi'[h]$ for an assignment $h : \mathrm{Vbl}(\mathcal{L}') \to |\mathcal{A}'|$, whereby $\mathrm{Vbl}(\mathcal{L}) = \mathrm{Vbl}(\mathcal{L}')$. Therefore, we have $\mathcal{A}' \models \psi(y/f(v_1, \dots, v_n))[h\binom{v_1}{a_1} \dots \binom{v_n}{a_n}]$ for all $a_1, \dots, a_n \in |\mathcal{A}'|$, which implies $\mathcal{A}' \models \psi[h\binom{v_1}{a_1} \dots \binom{v_n}{a_n}\binom{y}{f^{\mathcal{A}'}(a_1,\dots,a_n)}]$ for all $a_1, \dots, a_n \in |\mathcal{A}'|$.

   Let $\mathcal{A}$ be the $\mathcal{L}$-structure that is obtained from $\mathcal{A}'$ by removing $f^{\mathcal{A}'}$, which is the interpretation of f in $\mathcal{A}'$. We then have
   $$\mathcal{A} \models \psi[h\binom{v_1}{a_1} \cdots \binom{v_n}{a_n}\binom{y}{f^{\mathcal{A}}(a_1, \dots, a_n)}]$$
   for all $a_1, \dots, a_n \in |\mathcal{A}|$ since $|\mathcal{A}| = |\mathcal{A}'|$. This means that φ is satisfiable.

2. *Let φ be satisfiable.* Then there exists an $\mathcal{L}$-structure $\mathcal{A}$ such that $\mathcal{A} \models \varphi[h]$ for an assignment $h : \mathrm{Vbl}(\mathcal{L}) \to |\mathcal{A}|$. This means that for all $a_1, \dots, a_n \in |\mathcal{A}|$ we have $\mathcal{A} \models \exists y \psi[h\binom{v_1}{a_1} \dots \binom{v_n}{a_n}]$. Therefore, there exists for each $(a_1, \dots, a_n) \in |\mathcal{A}|^n$ an $a \in |\mathcal{A}|$ such that $\mathcal{A} \models \psi[h\binom{v_1}{a_1} \dots \binom{v_n}{a_n}\binom{y}{a}]$. As a consequence of the axiom of choice a function $F : |\mathcal{A}|^n \to |\mathcal{A}|$ that chooses an $a \in |\mathcal{A}|$ for every n-tuple $(a_1, \dots, a_n) \in |\mathcal{A}|^n$, such an element a exists. That is, we have $\mathcal{A} \models \psi[h\binom{v_1}{a_1} \dots \binom{v_n}{a_n}\binom{y}{F(a_1,\dots,a_n)}]$ for all $a_1, \dots, a_n \in |\mathcal{A}|$.

   We define $|\mathcal{A}'| := |\mathcal{A}|$ and $f^{\mathcal{A}'} := F$. Then $\mathcal{A}'$ is an $\mathcal{L} \cup \{f\}$-structure and h a variable assignment into $\mathcal{A}'$ with (according to Lemma 5.1.2)
   $$\mathcal{A}' \models \forall v_1 \dots \forall v_n \psi(y/f(v_1, \dots, v_n))[h],$$
   which means that φ' is satisfiable.

■

The following theorem summarizes the previous results of this section.

Theorem 7.1.12. *Let φ be an $\mathcal{L}$-formula. Then there is an extension $\mathcal{L}_1$ that extends $\mathcal{L}$ by finitely many function symbols and an $\mathcal{L}_1$-sentence φ_1 in proper prenex normal form with no existential quantifiers that is satisfiability equivalent with φ.*

Proof: Our proof consists of two steps.

1. If $\mathrm{Fr}(\varphi) \subseteq \{v_1, \ldots, v_n\}$ we define $\varphi' := \forall v_1 \ldots \forall v_n \varphi$. φ' is an $\mathcal{L}$-sentence that is equivalent to φ according to Corollary 3.2.16.

2. According to Theorem 7.1.11 there is an extension $\mathcal{L}_1$ of $\mathcal{L}$ by finitely many function symbols and an $\mathcal{L}_1$-sentence φ_1 with no existential quantifiers such that φ_1 is satisfiability equivalent with φ'.

Since φ_1 is satisfiability equivalent with φ the proof is complete. ■

Remark 7.1.13. The standard form of φ_1 in Theorem 7.1.12 is $\varphi_1 = \forall v_1 \ldots \forall v_n \psi$ such that ψ contains no quantifiers and is equal to

$$(\vartheta_{11} \vee \ldots \vee \vartheta_{1s(1)}) \wedge \ldots \wedge (\vartheta_{m1} \vee \ldots \vee \vartheta_{ms(m)}),$$

where ϑ_{ij} are atomic or negated atomic formulae.

A formula $\tau_1 \vee \ldots \vee \tau_k$, where τ_i is either an atomic or negated atomic formula, is called a *clause*.

7.2 Herbrand models

For the rest of this chapter we consider a finite formal language $\mathcal{L} = (\lambda, \mu, K)$ that contains at least one constant symbol. This means that the sets λ, μ and K are finite, $K \neq \emptyset$ and that the variable symbols are built with the symbols v and $'$ (cf. Remark 2.1.2).

If $\mathcal{L}$ does not contain any constant symbols, we add a constant symbol a. We can therefore assume that $\mathcal{L}$ contains a distinguished constant symbol a.

Definition 7.2.1. Let φ be an $\mathcal{L}$-sentence in Skolem form. The *Herbrand universe* $\mathbf{D}(\varphi)$ is either the set

$$D(\varphi) := \{t \in \mathrm{Tm}(L) \,|\, t \text{ only contains constant and function symbols occurring in } \varphi\}$$

or

$$D(\varphi) := \{t \in \mathrm{Tm}(L) \,|\, t \text{ only contains function symbols occurring in } \varphi \text{ and } a\}$$

depending on whether φ contains at least one constant symbol.

The set $D(\varphi)$ can be defined inductively:

1. All constant symbols of φ belong to $D(\varphi)$. If φ does not contain any constant symbol, then a belongs to $D(\varphi)$.

2. If f is an n-ary function symbol occurring in φ and $t_1, \ldots, t_n$ belong to $D(\varphi)$, then $f(t_1, \ldots, t_n) \in D(\varphi)$.

Example 7.2.2. Let φ be the formula $\forall x \forall y \forall z P(x, f(y), g(z, x))$, assuming that $\mathcal{L}$ contains the relation symbol P and the function symbols f and g. The function symbols occurring in φ are f and g, whereas no constant symbol occurs in φ. This gives us

$$\begin{aligned} D(\varphi) = \{&a, f(a), g(a,a), f(f(a)), f(g(a,a)), g(a, f(a)), \\ &g(f(a), a), g(f(a), f(a)), g(a, (g(a,a)), \ldots\} \end{aligned}$$

Example 7.2.3. Let φ be the formula $\forall x \forall y Q(b, f(x), h(y, c))$, assuming that $\mathcal{L}$ contains the symbols Q, f, g, b, c. The function symbols occurring in φ are f and h, the constant symbols are b and c. This gives us

$$\begin{aligned} D(\varphi) = \{&b, c, f(b), f(c), h(b,b), h(b,c), h(c,b), h(c,c), f(h(b,b)), f(h(b,c)), \\ &f(h(c,b)), f(h(c,c)), h(f(b), b), h(f(b), c)), h(f(c), b), \ldots\} \end{aligned}$$

Definition 7.2.4. Let φ be an $\mathcal{L}$-sentence in Skolem form and let $\mathcal{L}_\varphi$ be the sublanguage of $\mathcal{L}$ containing only the non-logical symbols of $\mathcal{L}$ that occur in φ (and additionally a in case that φ does not contain any constant symbols). An $\mathcal{L}_\varphi$-structure $\mathcal{A}$ is called a *Herbrand structure (for φ)* if

1. $|\mathcal{A}| = D(\varphi)$,

2. $c^{\mathcal{A}} = c$ if $c \in \mathcal{L}_\varphi$ is a constant symbol,

3. $f^{\mathcal{A}}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$ if $f \in \mathcal{L}_\varphi$ is an n-ary function symbol and $t_1, \ldots, t_n \in D(\varphi)$.

Remark 7.2.5. The interpretation of the relation symbols of $\mathcal{L}_\varphi$ in a Herbrand structure is not determined. Whether a Herbrand structure is a model of φ depends on the interpretation of the relation symbols. Such a model is called a *Herbrand model of φ*.

Example 7.2.6. (Example 7.2.2 continued.) We have $|\mathcal{A}| = D(\varphi)$ and for instance

$$f^{\mathcal{A}}(a) = f(a) \qquad f^{\mathcal{A}}(f(a)) = f(f(a)) \qquad g^{\mathcal{A}}(a, g(a,a)) = g(a, g(a,a)).$$

If we now define $P^{\mathcal{A}}$ by

$$P^{\mathcal{A}} := \{(t_1, t_2, t_3) \in D(\varphi) \times D(\varphi) \times D(\varphi) \mid t_1 = t_2\},$$

then the resulting Herbrand structure $\mathcal{A}$ is not a model of φ:

Consider $t_1 := a$, $t_2 := f(a)$, $t_3 := g(a,a)$. Then $(t_1, t_2, t_3) \notin P^{\mathcal{A}}$ since $a \neq f(a)$ (the terms are syntactically different). Therefore,

$$\mathcal{A} \not\models P(x, f(y), g(z,x))(x/a)(y/f(a))(z/g(a,a)),$$

which means
$\mathcal{A} \not\models \forall x \forall y \forall z P(x, f(y), g(z,x)) = \varphi$.

If, on the other hand, we define

$$P^{\mathcal{A}} := \{(t_1, t_2, t_3) \in D(\varphi) \times D(\varphi) \times D(\varphi) \mid t_1 = u_1, t_2 = f(u_2), \\ t_3 = g(u_3, u_1), u_1, u_2, u_3 \in D(\varphi)\}$$

then $\mathcal{A} \models \varphi$.

Remark 7.2.7. Herbrand structures distinguish themselves by interpreting terms that do not contain any variables as the term itself: We have $t^{\mathcal{A}} = t$ for every term t with no variables from $\mathcal{L}_\varphi$.

Theorem 7.2.8. *Let φ be a sentence in Skolem form. Then φ has a model iff φ has a Herbrand model.*

Proof: The proof contains two steps.

1. Let us assume that φ has a Herbrand model $\mathcal{A}$ which is an $\mathcal{L}_\varphi$-structure. We now interpret the non-logical symbols of $\mathcal{L} \setminus \mathcal{L}_\varphi$ arbitrarily in $|\mathcal{A}|$. This yields an $\mathcal{L}$-structure that is still a model of φ.

2. Let $\mathcal{A}$ be an $\mathcal{L}$-structure with $\mathcal{A} \models \varphi$. Then the interpretation $m := a^{\mathcal{A}} \in |\mathcal{A}|$ is determined. Next we construct a Herbrand structure $\mathcal{B}$ for φ. This means that we have to fix the interpretation $P^{\mathcal{B}}$ for the

relation symbol $P \in \mathcal{L}_\varphi$. For an n-ary relation symbol $P \in \mathcal{L}_\varphi$ we define

$$(t_1, \ldots, t_n) \in P^{\mathcal{B}} \qquad \Longleftrightarrow \qquad (t_1^{\mathcal{A}}, \ldots, t_n^{\mathcal{A}}) \in P^{\mathcal{A}}$$

We are going to show for all $\mathcal{L}_\varphi$-sentences ϑ in Skolem form that $\mathcal{A} \models \vartheta$ implies $\mathcal{B} \models \vartheta$. This will hold especially for $\vartheta = \varphi$, thus, $\mathcal{B} \models \varphi$ and $\mathcal{B}$ is a Herbrand model of φ.

We proceed by induction on the number n of universal quantifiers in ϑ.

- $n = 0$. This means that ϑ does not contain any quantifiers since it is assumed to be in Skolem form. From the definition of $P^{\mathcal{B}}$ the relation $\mathcal{B} \models \vartheta$ follows (carrying this out in all detail would involve induction on the complexity of ϑ).
- $n-1 \to n$. Consider $\vartheta = \forall v \beta$ such that β contains $n-1$ universal quantifiers. Since v may occur freely in β, and β therefore need not be a sentence, the induction hypothesis cannot be applied to β.

  We have assumed $\mathcal{A} \models \vartheta$, thus $\mathcal{A} \models \forall v \beta$. Therefore, $\mathcal{A} \models \beta[h\binom{v}{b}]$ for all valuations h and all $b \in |\mathcal{A}|$, especially also

  $$\mathcal{A} \models \beta[h\begin{pmatrix} v \\ t^{\mathcal{A}} \end{pmatrix}] \tag{7.1}$$

  for all $t \in D(\varphi)$. Since $t \in D(\varphi)$ does not contain any variables we have

  $$\bar{h}(t) = t^{\mathcal{A}}. \tag{7.2}$$

  Applying Lemma 5.1.2 to (7.1) and (7.2) yields $\mathcal{A} \models \beta(v/t)[h]$ for all valuations h and therefore

  $$\mathcal{A} \models \beta(v/t) \tag{7.3}$$

  for all $t \in D(\varphi)$. For $t \in D(\varphi)$ the formula $\beta(v/t)$ is an $\mathcal{L}_\varphi$-sentence in Skolem form. Therefore, the induction hypothesis can be applied to (7.3) and we obtain $\mathcal{B} \models \beta(v/t)$ for all $t \in D(\varphi)$. Since $t^{\mathcal{B}} = t$ and, thus, $\bar{h}(t) = t^{\mathcal{B}}$ for all $t \in D(\varphi)$, we obtain $\mathcal{B} \models \beta[h\binom{v}{t^{\mathcal{B}}}]$ for all $t \in D(\varphi)$ and valuations h, which is $\mathcal{B} \models \forall v \beta$. This finishes the proof since $\vartheta = \forall v \beta$.

■

7.3 Herbrand expansions

We start with a definition in which a set of formulae is assigned to a formula in Skolem form.

Definition 7.3.1. Let $\varphi = \forall v_1 \dots \forall v_n \vartheta$ be a sentence in Skolem form. Then the *Herbrand expansion* $E(\varphi)$ *of* φ is the following set of formulas:

$$E(\varphi) := \{\vartheta(v_1/t_1)(v_2/t_2)\dots(v_n/t_n) \in \mathrm{Fml}(\mathcal{L}_\varphi) \,|\, t_1, \dots, t_n \in D(\varphi)\}$$

We illustrate the preceding definition with an example.

Example 7.3.2. (Example 7.2.2 continued.) $E(\varphi)$ for instance contains

$P(a, f(a), g(a, a))$, substituting $(x/a)(y/a)(z/a)$, and

$P(g(a, a), f(f(a)), g(a, g(a, a)))$, substituting $(x/g(a, a))(y/f(a))(z/a))$.

The reason for introducing the Herbrand extension $E(\varphi)$ of a formula φ is the following theorem, which is the basis for the semi decision procedure, which will be presented later (cf. Corollary 7.3.6).

Theorem 7.3.3 (Gödel-Herbrand-Skolem). *Let φ be a sentence in Skolem form. Then φ is satisfiable iff $E(\varphi)$ is satisfiable.*

Proof: According to Theorem 7.2.8 it suffices to show that φ has a Herbrand model iff $E(\varphi)$ is satisfiable.

Let $\varphi = \forall v_1 \dots \forall v_n \vartheta$. The following sequence of equivalences establishes the assertion:

$$\begin{aligned}
\mathcal{A} \text{ is a Herbrand model of } \varphi \quad &\Longleftrightarrow \quad \mathcal{A} \models \vartheta(v_1/t_1)\dots(v_n/t_n) \text{ for all} \\
& \quad t_1, \dots, t_n \in D(\varphi) \\
&\Longleftrightarrow \quad \mathcal{A} \models \beta \text{ for all } \beta \in E(\varphi) \\
&\Longleftrightarrow \quad \mathcal{A} \text{ is a model of } E(\varphi)
\end{aligned}$$

The first equivalence follows from Lemma 5.1.2, cf. also the proof of Theorem 7.2.8. ∎

Remark 7.3.4. The results obtained so far can be summarized as follows: A first-order sentence is satisfiability equivalent to a (generally infinite) set of formulae that are conjunctions of disjunctions of atomic or negative atomic formulae without variables.

Another application of the finiteness theorem is the important theorem of Herbrand.

Theorem 7.3.5 (Herbrand). *Let φ be a sentence in Skolem form. φ is unsatisfiable if and only if there is a* finite *subset of $E(\varphi)$ that is not satisfiable.*

Proof: The assertion follows from Theorem 7.3.3 with the Finiteness Theorem (cf. Theorem 6.5.4): $E(\varphi)$ has a model iff every finite subset of $E(\varphi)$ has a model. ∎

Corollary 7.3.6. *There is a semi decision procedure for the unsatisfiability of first-order formulae. That is, there is a procedure that produces for input φ the output "no" after a finite time if φ is not satisfiable and does not stop if φ is satisfiable.*

Proof: *Input*: An $\mathcal{L}$-formula φ.

> *Step 1*: Construct an $\mathcal{L}_1$-sentence β in Skolem form in an appropriate extension $\mathcal{L}_1$ of $\mathcal{L}$ that is satisfiability equivalent to φ. This is possible according to Theorem 7.1.12.
>
> *Step 2*: Construct with an appropriate procedure the set $E(\beta) = \{\pi_1, \pi_2, \ldots\}$ (inductively on the complexity of the terms that occur in π_i). Test after having constructed π_i whether the formula $\pi_1 \wedge \ldots \wedge \pi_i$ is satisfiable.
>
> The latter is done by assigning truth values to the atomic formulae occurring in $\pi_1, \ldots, \pi_i$. (Atomic formulae occurring in $E(\varphi)$ in Example 7.3.2 are for instance $P(a, f(a), g(a, a))$ and $P(g(a, a), f(f(a)), g(a, g(a, a)))$. Assigning truth values to these two formulae amounts to determining the interpretation of the relation symbol P in a Herbrand structure.

Output: "unsatisfiable" if there is an $i \in \mathbb{N}$ such that $\pi_1 \wedge \ldots \wedge \pi_{i_0}$ for $i_0 \in \mathbb{N}$ is unsatisfiable. ∎

Remark 7.3.7. Corollary 7.3.6 is the best possible result concerning the decidability of the satisfiability of a formula φ. If there was a complementary procedure that would produce the output "yes" if φ is satisfiable and does not stop otherwise, such a procedure together with the procedure of Corollary 7.3.6 would yield a procedure deciding satisfiability. However, this would contradict Corollary 5.2.5!

Solve the exercises of Section 8.10!

Chapter 8

Exercises

8.1 Syntax

Exercise 8.1.1. Let the formal language $\mathcal{L}'$ be an extension of the formal language $\mathcal{L}$. Show that

1. $\mathrm{Tm}(\mathcal{L}) \subseteq \mathrm{Tm}(\mathcal{L}')$
2. $\mathrm{Fml}(\mathcal{L}) \subseteq \mathrm{Fml}(\mathcal{L}')$
3. $\mathrm{Sen}(\mathcal{L}) \subseteq \mathrm{Sen}(\mathcal{L}')$

Exercise 8.1.2. Let $\mathcal{L}$ be a formal language containing the constant symbols c_0, c_1, the binary function symbols f_0, f_1, f_2 and a binary relation symbol R. Furthermore we define

$$\begin{aligned}\varphi := {} & \forall v_1 \forall v_2 \forall v_3 \neg(\neg\neg R(f_2(f_0(v_2, v_2), f_0(f_1(f_1(c_1, c_1), f_1(c_1, c_1)), \\ & f_0(v_1, v_3))), c_0) \wedge \neg\neg\forall v_0 \neg f_1(f_0(v_0, f_0(v_1, v_0)), f_1(f_0(v_2, v_0), v_3)) \doteq c_0).\end{aligned}$$

1. Determine the sets $\mathrm{Fr}(\varphi)$, $\mathrm{Bnd}(\varphi)$ and $\mathrm{Subfml}(\varphi)$ defined in Definition 2.3.1 and Exercise 2.3.4.
2. Simplify φ by using the abbreviations and conventions of Subsection 2.2.3 as well as the fact that φ and $\neg\neg\varphi$ are equivalent. (The latter is a simple property of our deductive system.)

3. We consider the following $\mathcal{L}$-structure $\mathcal{A}$:

$$\begin{aligned}
|\mathcal{A}| &:= \mathbb{R} \\
f_0^{\mathcal{A}} &:= \cdot \qquad \text{(multiplication)} \\
f_1^{\mathcal{A}} &:= + \qquad \text{(addition)} \\
f_2^{\mathcal{A}} &:= - \qquad \text{(subtraction)} \\
c_0^{\mathcal{A}} &:= 0 \\
c_1^{\mathcal{A}} &:= 1
\end{aligned}$$

   Now, describe in our meta language the mathematical property of the real numbers expressed by φ.

Exercise 8.1.3. For a formal language $\mathcal{L}$ and a $\mathcal{L}$–term t, let $\sharp_f(t)$ denote the number of function symbols in t, and $\sharp_p(t)$ the number of parentheses in t. Prove that

$$\sharp_p(t) = 2 \cdot \sharp_f(t)$$

for any $\mathcal{L}$–term t.

8.2 Semantics

Exercise 8.2.1. Let $\mathcal{L}$ be a formal language with a binary relation symbol R and a unary function symbol f. Furthermore, we define
$\varphi := \forall v \exists w R(v, w, f(v, w))$

Find $\mathcal{L}$-structures $\mathcal{A}_1$ and $\mathcal{A}_2$ such that

$$\mathcal{A}_1 \models \varphi \qquad \text{and} \qquad \mathcal{A}_2 \not\models \varphi$$

Exercise 8.2.2. Define a formal language $\mathcal{L}$ and an $\mathcal{L}$-sentence φ such that for every $\mathcal{L}$-structure $\mathcal{A}$ we have:

$$\mathcal{A} \models \varphi \qquad \text{iff} \qquad |\mathcal{A}| \text{ has more than five elements}$$

Exercise 8.2.3. Let $\mathcal{L}$ be a formal language and $\varphi \in \mathrm{Sen}(\mathcal{L})$.

Show that

$$\varphi \text{ cannot be satisfied} \qquad \text{iff} \qquad \neg\varphi \text{ is valid.}$$

Exercise 8.2.4.

1. Let $\mathcal{L}$ be a formal language. Show that every $\mathcal{L}$-formula of the form $\varphi \to \forall v \varphi$ has a model.

2. Define a formal language $\mathcal{L}$, an $\mathcal{L}$-formula φ and an $\mathcal{L}$-structure $\mathcal{A}$ such that $\varphi \to \forall v \varphi$ is not valid in $\mathcal{A}$.

8.3 Propositional formulae

Exercise 8.3.1. 1. Set up the truth tables for the connectives $\vee$, $\rightarrow$ and $\leftrightarrow$, which were introduced as abbreviations in Subsection 2.2.3.

2. Let the following four propositional valuations $s_1, \ldots, s_4$ with

$$\begin{array}{ll} s_1(P_0) = \mathsf{t} & s_1(P_1) = \mathsf{t} \\ s_2(P_0) = \mathsf{t} & s_2(P_1) = \mathsf{f} \\ s_3(P_0) = \mathsf{f} & s_3(P_1) = \mathsf{t} \\ s_4(P_0) = \mathsf{f} & s_4(P_1) = \mathsf{f} \end{array}$$

and $s_i(P_n) = \mathsf{f}$ for $n \geq 2$ and $1 \leq i \leq 4$ be given. Determine $\bar{s_i}(\Psi_j)$ for $1 \leq i \leq 4$, $1 \leq j \leq 2$ and

- $\Psi_1 := (P_0 \wedge P_1) \leftrightarrow \neg(\neg P_0 \vee \neg P_1)$
- $\Psi_2 := (P_0 \rightarrow P_1) \leftrightarrow \neg(P_1 \rightarrow P_0)$

8.4 Substitution

Exercise 8.4.1. Show that if y does not occur in φ, then

$$(\varphi(x/t))(c_0/y) = (\varphi(c_0/y))(x/t(c_0/y)).$$

8.5 Deductions and consistency

Exercise 8.5.1. Let $\mathcal{L}$ be a formal language and Σ a consistent set of $\mathcal{L}$-sentences. Show that the two statements

- "For all $\mathcal{L}$-sentences α either $\Sigma \vdash \alpha$ or $\Sigma \vdash \neg\alpha$"
- "$\Sigma \vdash \alpha \vee \beta$ implies $\Sigma \vdash \alpha$ or $\Sigma \vdash \beta$."

are equivalent.

Does this equivalence still hold if Σ is inconsistent?

Exercise 8.5.2. Let $\mathcal{L}$ be a formal language and let us assume that φ, ψ and ρ are $\mathcal{L}$-formulae such that there are deductions for $\varphi \leftrightarrow \rho$ and $\psi \leftrightarrow \rho$.

Construct from these deductions a deduction for $\varphi \leftrightarrow \psi$.

Exercise 8.5.3. Let $\mathcal{L}$ be a formal language containing at least the constant symbol c. Show that the set $\{\neg\exists v(v \doteq c)\}$ is inconsistent.

Exercise 8.5.4. Let $\mathcal{L}$ be a formal language. Show that Σ is a consistent set of $\mathcal{L}$-sentences iff $\Sigma \nvdash \exists v(\neg v \doteq v)$.

Exercise 8.5.5. Solve Exercise 4.4.4 (of the text).

Exercise 8.5.6. Prove the cases of Lemma 5.3.1 that have not been treated in the text.

Exercise 8.5.7. Prove (Tr), (R_i) and (f_j) of Lemma 5.3.3.

Exercise 8.5.8. Let $\mathcal{L}$ be a formal language, α and β $\mathcal{L}$-formulae. Show

1. Show $\vdash \exists v(\alpha \vee \beta) \leftrightarrow (\exists v\alpha \vee \exists v\beta)$
2. Show that $\forall v(\alpha \vee \beta) \leftrightarrow (\forall v\alpha \vee \forall v\beta)$ does not hold.

8.6 Models as counter-examples

Exercise 8.6.1. Let the formal language $\mathcal{L}$ contain only the binary relation symbol $<$. Show that the five $\mathcal{L}$-sentences $\alpha_1, \ldots, \alpha_5$

$\alpha_1 :=$	$\forall v(\neg(v < v))$	irreflexivity
$\alpha_2 :=$	$\forall v, w, x((v < w \wedge w < x) \rightarrow v < x)$	transitivity
$\alpha_3 :=$	$\forall v, w(v < w \vee v = w \vee v > w)$	totality of the order
$\alpha_4 :=$	$\forall v, w\exists x(v < w \rightarrow (v < x \wedge x < w))$	density
$\alpha_5 :=$	$\forall v\exists w, x(w < v \wedge v < x)$	no end points

are independent, that is, $\{\alpha_1, \ldots, \alpha_5\} \setminus \{\alpha_i\} \nvdash \alpha_i$ for $i = 1, \ldots, 5$.

Exercise 8.6.2. Let the formal language $\mathcal{L} = \{R\}$ be given with R being a binary relation symbol. Furthermore, let $\Sigma := \{\alpha, \beta, \gamma\}$ with

$$\begin{aligned}
\alpha &:= \forall v R(v, v) \\
\beta &:= \forall v, w(R(v, w) \rightarrow R(w, v)) \\
\gamma &:= \forall v, w, x((R(v, w) \wedge R(w, x)) \rightarrow R(v, x))
\end{aligned}$$

1. Show
$$\{\alpha, \beta\} \nvdash \gamma \qquad \{\alpha, \gamma\} \nvdash \beta \qquad \{\beta, \gamma\} \nvdash \alpha$$
2. Do we have $\Sigma \vdash \forall v, w\exists x(R(v, x) \wedge R(w, x))$?

8.7 The Deduction Theorem

Exercise 8.7.1. Analyze the proof of the Deduction Theorem in order to replace the condition $Fr(\varphi) := \emptyset$ by minimal conditions which φ and ψ have to satisfy.

Exercise 8.7.2. Let $\mathcal{L}$ be a formal language, α and β $\mathcal{L}$-formulae such that v is the only free variable. Show

$$\vdash \forall v(\alpha \to \beta) \to (\exists v\alpha \to \exists v\beta)$$

by applying the Deduction Theorem (cf. Theorem 5.5.1) twice.

8.8 Henkin structures

Exercise 8.8.1. Let $\mathcal{L}$ be a formal language and Σ a set of $\mathcal{L}$-sentences. Show that the interpretation of the function symbols of $\mathcal{L}$ is well-defined in $|\mathcal{A}_\Sigma| := \mathrm{CT}_\mathcal{L}/\approx$ as used in the proof of Lemma 6.4.1.

Exercise 8.8.2. Let $\mathcal{L}$ be a formal language and Σ an inconsistent set of $\mathcal{L}$-sentences. Describe the set $|\mathcal{A}_\Sigma|$.

Exercise 8.8.3. Let $\mathcal{L}$ be a formal language containing the constant symbols $0, 1$, two binary function symbols $\sqcap$ and $\sqcup$ and a unary function symbol $*$. Further let $\Theta \subset \mathrm{Sen}(\mathcal{L})$ consist of the following sentences:

$$\begin{array}{ll}
(I) & \forall x(x \sqcap x \doteq x) \wedge \forall x(x \sqcup x \doteq x) \\
(C) & \forall x, y(x \sqcap y \doteq y \sqcap x) \wedge \forall x, y(x \sqcup v \doteq y \sqcup x) \\
(A) & \forall x, y, z((x \sqcap y) \sqcap z \doteq x \sqcap (y \sqcap z)) \wedge \\
 & \forall x, y, z((x \sqcup y) \sqcup z \doteq x \sqcup (y \sqcup z)) \\
(Ab) & \forall x, y(x \sqcap (x \sqcup y) \doteq x) \wedge \forall x, y(x \sqcup (x \sqcap y) \doteq x) \\
(D) & \forall x, y, z(x \sqcap (y \sqcup z) \doteq (x \sqcap y) \sqcup (x \sqcap z)) \wedge \\
 & \forall x, y, z(x \sqcup (y \sqcap z) \doteq (x \sqcup y) \sqcap (x \sqcup z)) \\
(Co) & \forall x(x \sqcup x* = 1) \wedge \forall x(x \sqcap x* = 0) \\
(U) & \forall x(x \sqcap 1 = x) \wedge \forall x(x \sqcup 0 = x)
\end{array}$$

1. We define for an arbitrary set X the $\mathcal{L}$-structure $\mathcal{A}$ by setting

$$\begin{array}{lll} |\mathcal{A}| := P(X) & & \\ \sqcap^{\mathcal{A}} := \bigcap & & \text{(set intersection)} \\ \sqcup^{\mathcal{A}} := \bigcup & & \text{(set union)} \\ *^{\mathcal{A}} \ := \ ^{\mathcal{C}} & & \text{(set complement)} \\ 0^{\mathcal{A}} \ := \emptyset & & \\ 1^{\mathcal{A}} \ := X & & \end{array}$$

   Show: $\mathcal{A} \models \Theta$

2. Determine $\mathrm{CT}_{\mathcal{L}}/\approx$ with respect to Θ.

3. Extend $\mathcal{L}$ with a new constant symbol a to a formal language $\mathcal{L}'$. Determine $\mathrm{CT}_{\mathcal{L}'}/\approx$ with respect to Θ.

4. Extend $\mathcal{L}'$ with a new constant symbol b to a formal language $\mathcal{L}''$ and define

$$\Sigma := \Theta \cup \{\forall x, y, z(((x \not\doteq 0) \wedge (y \not\doteq 0) \wedge (z \not\doteq 0)) \rightarrow \\ ((x \sqcap y \not\doteq 0) \vee (x \sqcap z \not\doteq 0) \vee (y \sqcap z \doteq 0)))\}$$

   Determine $\mathrm{CT}_{\mathcal{L}''}/\approx$ with respect to Σ.

8.9 A non-standard model for the natural numbers

Exercise 8.9.1. Let $\mathcal{L}$ be a formal language containing the constant symbols $0, 1$, two binary function symbols $+$ and $\cdot$ and the binary relation symbol $\leq$. Furthermore, let $\Sigma \subset \mathrm{Sen}(\mathcal{L})$ consist of all sentences that hold in the natural numbers under the standard interpretation of the non-logical symbols of $\mathcal{L}$.

Now, let us define $\mathcal{L}'$ by extending $\mathcal{L}$ with a new constant symbol c and set

$$\Theta := \Sigma \cup \{\neg\, c \doteq 0, \neg\, c \doteq 1, \ldots, \neg\, c \doteq n, \ldots\},$$

with n being an abbreviation of the term $\underbrace{(1 + \ldots + 1)}_{n-\text{times}}$.

1. Show that Θ is consistent and that we have $\mathcal{A} \models n \leq c$ for all $n \in \mathbb{N}$ and all models $\mathcal{A}$ of Θ.

2. The first part of this exercise can be understood by saying that a model $\mathcal{A}$ of Θ contains "infinitely big" elements (e.g $c^{\mathcal{A}}$).

   Show that the set of these "infinitely big" elements of $|\mathcal{A}|$ does not contain a smallest element.

3. We know that every nonempty subset of the natural numbers contains a smallest element. This sentence thus holds in $\mathbb{N}$ but does not hold in a model of Θ as shown above.

   How do you comment on this?

8.10 Herbrand structures

Exercise 8.10.1. In this exercise the goal is to prove Theorem 7.1.9. The task is split into several subtasks.

1. Prove the following lemma:

   Lemma. If α is a formula and x, y are variables such that $x \in \mathrm{Fr}(\alpha)$ and $y \notin \mathrm{Fr}(\alpha) \cup \mathrm{Bnd}(\alpha)$, then we have

   $$\vdash \forall x \alpha \leftrightarrow \forall y \alpha(x/y).$$

   This implies that there is for every formula α an equivalent formula α' with the property that all bound variables are pairwise different and also different from the free variables. (The lemma also holds for the existential quantifier $\exists$ since it is an abbreviation for $\neg\forall\neg$.)

2. Prove the following lemma:

   Lemma. Let $\alpha, \beta, \gamma, \delta$ be formulae such that δ is obtained from γ by replacing some occurrences of the subformula α with the formula β. Then we have:

   $$\vdash \alpha \leftrightarrow \beta \qquad \textit{implies} \qquad \vdash \gamma \leftrightarrow \delta$$

3. Show that if $Q \in \{\forall, \exists\}$ we have

$$\vdash Qx(\alpha \wedge \beta) \leftrightarrow (\alpha \wedge Qx\beta) \qquad x \notin \mathrm{Fr}(\alpha)$$

$$\vdash Qx(\alpha \vee \beta) \leftrightarrow (\alpha \vee Qx\beta) \qquad x \notin \mathrm{Fr}(\alpha)$$

$$\vdash Qx(\alpha \rightarrow \beta) \leftrightarrow (\alpha \rightarrow Qx\beta) \qquad x \notin \mathrm{Fr}(\alpha)$$

4. Now prove Theorem 7.1.9 using 1., 2. and 3. as well as the fact that every propositional formula is equivalent to a formula in conjunctive normal form.

Exercise 8.10.2. Let $\mathcal{L}$ be a formal language containing the unary and binary R_1 and R_2 relation symbols, respectively. Find for each of the formulae below an equivalent formulae in proper prenex normal form.

- $\neg(R_1(x) \wedge \neg\forall y R_2(x, y))$
- $\forall x(R_1(x) \wedge R_2(x, y)) \wedge \neg(\exists y R_1(y) \wedge \exists z R_2(y, z))$
- $(\forall x R_2(x, y) \wedge \neg\exists y R_1(y)) \wedge \neg\forall x \forall y R_2(x, y)$
- $\forall x R_1(x) \rightarrow \forall y R_2(x, y)$
- $\exists x R_2(x, y) \vee (R_1(x) \rightarrow \neg\exists z R_2(x, z))$

Bibliography

[hermes78] Hermes, Hans: Aufzhlbarkeit, Entscheidbarkeit, Berechenbarkeit. Springer, Berlin, Heidelberg, New York 1978.