Mathematical Logic
(Math 570)
Lecture Notes

Lou van den Dries

Fall Semester 2007

Contents

Chapter 1

Preliminaries

We start with a brief overview of mathematical logic as covered in this course. Next we review some basic notions from elementary set theory, which provides a medium for communicating mathematics in a precise and clear way. In this course we develop mathematical logic using elementary set theory as given, just as one would do with other branches of mathematics, like group theory or probability theory.

For more on the course material, see

> Shoenfield, J. R., **Mathematical Logic**, Reading, Addison-Wesley, 1967.

For additional material in Model Theory we refer the reader to

> Chang, C. C. and Keisler, H. J., **Model Theory**, New York, North-Holland, 1990,

> Poizat, B., **A Course in Model Theory**, Springer, 2000,

and for additional material on Computability, to

> Rogers, H., **Theory of Recursive Functions and Effective Computability**, McGraw-Hill, 1967.

1.1 Mathematical Logic: a brief overview

Aristotle identified some simple patterns in human reasoning, and Leibniz dreamt of reducing reasoning to calculation. As a mathematical subject, however, logic is relatively recent: the 19th century pioneers were Bolzano, Boole, Cantor, Dedekind, Frege, Peano, C.S. Peirce, and E. Schröder. From our perspective we see their work as leading to boolean algebra, set theory, propositional logic, predicate logic, as clarifying the foundations of the natural and real number systems, and as introducing suggestive symbolic notation for logical operations. Also, their activity led to the view that *logic + set theory* can serve as a basis for all of mathematics. This era did not produce theorems in mathematical logic of any real depth, [1] but it did bring crucial progress of a conceptual nature, and the recognition that logic as used in mathematics obeys mathematical rules that can be made fully explicit.

In the period 1900-1950 important new ideas came from Russell, Zermelo, Hausdorff, Hilbert, Löwenheim, Ramsey, Skolem, Lusin, Post, Herbrand, Gödel, Tarski, Church, Kleene, Turing, and Gentzen. They discovered the first real theorems in mathematical logic, with those of Gödel having a dramatic impact. Hilbert (in Göttingen), Lusin (in Moscow), Tarski (in Warsaw and Berkeley), and Church (in Princeton) had many students and collaborators, who made up a large part of that generation and the next in mathematical logic. Most of these names will be encountered again during the course.

The early part of the 20th century was also marked by the so-called

$$foundational \quad crisis \quad in \quad mathematics.$$

A strong impulse for developing mathematical logic came from the attempts during these times to provide solid foundations for mathematics. Mathematical logic has now taken on a life of its own, and also thrives on many interactions with other areas of mathematics and computer science.

In the second half of the last century, logic as pursued by mathematicians gradually branched into four main areas: *model theory*, *computability theory* (or *recursion theory*), *set theory*, and *proof theory*. The topics in this course are part of the common background of mathematicians active in any of these areas.

What distinguishes mathematical logic within mathematics is that

$$statements \quad about \quad mathematical \quad objects$$

are taken seriously as mathematical objects in their own right. More generally, in mathematical logic we formalize (formulate in a precise mathematical way) notions used informally by mathematicians such as:

- **property**
- **statement** (in a given language)
- **structure**
- **truth** (what it means for a given statement to be true in a given structure)
- **proof** (from a given set of axioms)
- **algorithm**

[1] In the case of set theory one could dispute this. Even so, the main influence of set theory on the rest of mathematics was to enable simple constructions of great generality, like cartesian products, quotient sets and power sets, and this involves only very elementary set theory.

Once we have mathematical definitions of these notions, we can try to prove theorems about these formalized notions. If done with imagination, this process can lead to unexpected rewards. Of course, formalization tends to caricature the informal concepts it aims to capture, but no harm is done if this is kept firmly in mind.

Example. The notorious Goldbach Conjecture asserts that every even integer greater than 2 is a sum of two prime numbers. With the understanding that the variables range over $\mathbf{N} = \{0, 1, 2, \dots\}$, and that $0, 1, +, \cdot, <$ denote the usual arithmetic operations and relations on $\mathbf{N}$, this assertion can be expressed formally as

$$(GC) \qquad \forall x[(1+1 < x \wedge \text{even}(x)) \rightarrow \exists p \exists q(\text{prime}(p) \wedge \text{prime}(q) \wedge x = p+q)]$$

where $\text{even}(x)$ abbreviates $\exists y(x = y + y)$ and $\text{prime}(p)$ abbreviates

$$1 < p \wedge \forall r \forall s(p = r \cdot s \rightarrow (r = 1 \vee s = 1)).$$

The expression GC is an example of a formal statement (also called a *sentence*) in the *language of arithmetic*, which has symbols $0, 1, +, \cdot, <$ to denote arithmetic operations and relations, in addition to logical symbols like $=, \wedge, \vee, \neg, \rightarrow, \forall, \exists$, and variables x, y, z, p, q, r, s.

The Goldbach Conjecture asserts that this particular sentence GC is *true* in the *structure* $(\mathbf{N};\ 0, 1, +, \cdot, <)$. (No proof of the Goldbach Conjecture is known.) It also makes sense to ask whether the sentence GC is true in the structure

$$(\mathbf{R};\ 0, 1, +, \cdot, <)$$

(It's *not*, as is easily verified. That the question makes sense does not mean that it is of any interest.)

A century of experience gives us confidence that all classical number-theoretic results—old or new, proved by elementary methods or by sophisticated algebra and analysis—can be proved from the Peano axioms for arithmetic. [2] However, in our present state of knowledge, GC might be true in $(\mathbf{N};\ 0, 1, +, \cdot, <)$, but not provable from those axioms. (On the other hand, once you know what exactly we mean by

provable from the Peano axioms,

you will see that if GC is provable from those axioms, then GC is true in $(\mathbf{N};\ 0, 1, +, \cdot, <)$, and that if GC is false in $(\mathbf{N};\ 0, 1, +, \cdot, <)$, then its negation $\neg GC$ is provable from those axioms.)

The point of this example is simply to make the reader aware of the notions "true in a given structure" and "provable from a given set of axioms," and their difference. One objective of this course is to figure out the connections (and disconnections) between these notions.

[2]Here we do not count as part of classical number theory some results like Ramsey's Theorem that can be stated in the language of arithmetic, but are arguably more in the spirit of logic and combinatorics.

Some highlights (1900–1950)

The results below are among the most frequently used facts of mathematical logic. The terminology used in stating these results might be unfamiliar, but that should change during the course. What matters is to get some preliminary idea of what we are aiming for. As will become clear during the course, each of these results has stronger versions, on which applications often depend, but in this overview we prefer simple statements over strength and applicability.

We begin with two results that are fundamental in model theory. They concern the notion of *model of* Σ where Σ is a set of sentences in a language L. At this stage we only say by way of explanation that a model of Σ is a mathematical structure in which all sentences of Σ are true. For example, if Σ is the (infinite) set of axioms for fields of characteristic zero in the language of rings, then a model of Σ is just a field of characteristic zero.

Theorem of Löwenheim and Skolem. *If Σ is a countable set of sentences in some language and Σ has a model, then Σ has a countable model.*

Compactness Theorem (Gödel, Mal'cev). *Let Σ be a set of sentences in some language. Then Σ has a model if and only if each finite subset of Σ has a model.*

The next result goes a little beyond model theory by relating the notion of "model of Σ" to that of "provability from Σ":

Completeness Theorem (Gödel, 1930). *Let Σ be a set of sentences in some language L, and let σ be a sentence in L. Then σ is provable from Σ if and only if σ is true in all models of Σ.*

In our treatment we shall obtain the first two theorems as byproducts of the Completeness Theorem and its proof. In the case of the Compactness Theorem this reflects history, but the theorem of Löwenheim and Skolem predates the Completeness Theorem. The Löwenheim-Skolem and Compactness theorems do not mention the notion of provability, and thus model theorists often prefer to bypass Completeness in establishing these results; see for example Poizat's book.

Here is an important early result on a *specific* arithmetic structure:

Theorem of Presburger and Skolem. *Each sentence in the language of the structure $(\mathbf{Z};\ 0,1,+,-,<)$ that is true in this structure is provable from the axioms for ordered abelian groups with least positive element 1, augmented, for each $n = 2,3,4,...$, by an axiom that says that for every a there is a b such that $a = nb$ or $a = nb+1$ or $\ldots$ or $a = nb+1+\cdots+1$ (with n disjuncts in total). Moreover, there is an algorithm that, given any sentence in this language as input, decides whether this sentence is true in $(\mathbf{Z};\ 0,1,+,-,<)$.*

Note that in $(\mathbf{Z};\ 0,1,+,-,<)$ we have not included multiplication among the *primitives*; accordingly, nb stands for $b+\cdots+b$ (with n summands).

When we do include multiplication, the situation changes dramatically:

Incompleteness and undecidability of arithmetic. (Gödel-Church, 1930's). *One can construct a sentence in the language of arithmetic that is true in the structure* $(\mathbf{N};\ 0,1,+,\cdot,<)$*, but not provable from the Peano axioms.*

There is no algorithm that, given any sentence in this language as input, decides whether this sentence is true in $(\mathbf{N};\ 0,1,+,\cdot,<)$.

Here "there is no algorithm" is used in the mathematical sense of

there cannot exist an algorithm,

not in the weaker colloquial sense of "no algorithm is known." This theorem is intimately connected with the clarification of notions like *computability* and *algorithm* in which Turing played a key role.

In contrast to these incompleteness and undecidability results on (sufficiently rich) arithmetic, we have

Tarski's theorem on the field of real numbers (1930-1950). *Every sentence in the language of arithmetic that is true in the structure*

$$(\mathbf{R};\ 0,1,+,\cdot,<)$$

is provable from the axioms for ordered fields augmented by the axioms
- *every positive element is a square,*
- *every odd degree polynomial has a zero.*

There is also an algorithm that decides for any given sentence in this language as input, whether this sentence is true in $(\mathbf{R};\ 0,1,+,\cdot,<)$.

1.2 Sets and Maps

We shall use this section as an opportunity to fix notations and terminologies that are used throughout these notes, and throughout mathematics. In a few places we shall need more set theory than we introduce here, for example, ordinals and cardinals. The following little book is a good place to read about these matters. (It also contains an axiomatic treatment of set theory starting from scratch.)

Halmos, P. R., **Naive set theory**, New York, Springer, 1974

In an axiomatic treatment of set theory as in the book by Halmos all assertions about sets below are proved from a few simple axioms. In such a treatment the notion of set itself is left undefined, but the axioms about sets are suggested by thinking of a set as a collection of mathematical objects, called its *elements* or *members*. To indicate that an object x is an element of the set A we write $x \in A$, in words: x is in A (or: x belongs to A). To indicate that x is not in A we write $x \notin A$. We consider the sets A and B as the same set (notation: $A = B$) if and only if they have exactly the same elements. We often introduce a set via the bracket notation, listing or indicating inside the brackets its elements.

For example, $\{1, 2, 7\}$ is the set with 1, 2, and 7 as its only elements. Note that $\{1, 2, 7\} = \{2, 7, 1\}$, and $\{3, 3\} = \{3\}$: the same set can be described in many different ways. Don't confuse an object x with the set $\{x\}$ that has x as its only element: for example, the object $x = \{0, 1\}$ is a set that has exactly two elements, namely 0 and 1, but the set $\{x\} = \{\{0, 1\}\}$ has only one element, namely x.

Here are some important sets that the reader has probably encountered previously.

Examples.
(1) The empty set: $\emptyset$ (it has no elements).
(2) The set of natural numbers: $\mathbf{N} = \{0, 1, 2, 3, \dots\}$.
(3) The set of integers: $\mathbf{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$.
(4) The set of rational numbers: $\mathbf{Q}$.
(5) The set of real numbers: $\mathbf{R}$.
(6) The set of complex numbers: $\mathbf{C}$.

Remark. Throughout these notes m and n always denote natural numbers. For example, "for all m ..." will mean "for all $m \in \mathbf{N}$...".

If all elements of the set A are in the set B, then we say that A is a subset of B (and write $A \subseteq B$). Thus the empty set $\emptyset$ is a subset of every set, and each set is a subset of itself. We often introduce a set A in our discussions by defining A to be the set of all elements of a given set B that satisfy some property P. Notation:

$$A := \{x \in B : x \text{ satisfies } P\} \qquad (\text{hence } A \subseteq B).$$

Let A and B be sets. Then we can form the following sets:

(a) $A \cup B := \{x : x \in A \text{ or } x \in B\}$ (*union of A and B*);
(b) $A \cap B := \{x : x \in A \text{ and } x \in B\}$ (*intersection of A and B*);
(c) $A \smallsetminus B := \{x : x \in A \text{ and } x \notin B\}$ (*difference of A and B*);
(d) $A \times B := \{(a, b) : a \in A \text{ and } b \in B\}$ (*cartesian product of A and B*).

Thus the elements of $A \times B$ are the so-called ordered pairs (a, b) with $a \in A$ and $b \in B$. The key property of ordered pairs is that we have $(a, b) = (c, d)$ if and only if $a = c$ and $b = d$. For example, you may think of $\mathbf{R} \times \mathbf{R}$ as the set of points (a, b) in the xy-plane of coordinate geometry.

We say that A and B are disjoint if $A \cap B = \emptyset$, that is, they have no element in common.

Remark. In a definition such as we just gave: "We say that $\cdots$ if —," the meaning of "if" is actually "if and only if." We committed a similar abuse of language earlier in defining set inclusion by the phrase "If —, then we say that $\cdots$." We shall continue such abuse, in accordance with tradition, but only in similarly worded *definitions*. Also, we shall often write "iff" or "$\Leftrightarrow$" to abbreviate "if and only if."

Maps

Definition. A *map* is a triple $f = (A, B, \Gamma)$ of sets A, B, Γ such that $\Gamma \subseteq A \times B$ and for each $a \in A$ there is exactly one $b \in B$ with $(a, b) \in \Gamma$; we write $f(a)$ for this unique b, and call it the *value of f at a* (or the *image of a under f*).[3] We call A the *domain of f*, and B the *codomain of f*, and Γ the *graph of f*.[4] We write $f : A \to B$ to indicate that f is a map with domain A and codomain B, and in this situation we also say that f is a map from A to B.

Among the many synonyms of *map* are

$$mapping, \quad assignment, \quad function, \quad operator, \quad transformation.$$

Typically, "function" is used when the codomain is a set of numbers of some kind, "operator" when the elements of domain and codomain are themselves functions, and "transformation" is used in geometric situations where domain and codomain are equal. (We use *equal* as synonym for *the same* or *identical*; also *coincide* is a synonym for *being the same*.)

Examples.
(1) Given any set A we have the identity map $1_A : A \to A$ defined by $1_A(a) = a$ for all $a \in A$.
(2) Any polynomial $f(X) = a_0 + a_1X + \cdots + a_nX^n$ with real coefficients $a_0, \ldots, a_n$ gives rise to a function $x \mapsto f(x) : \mathbf{R} \to \mathbf{R}$. We often use the "maps to" symbol $\mapsto$ in this way to indicate the rule by which to each x in the domain we associate its value $f(x)$.

Definition. Given $f : A \to B$ and $g : B \to C$ we have a map $g \circ f : A \to C$ defined by $(g \circ f)(a) = g(f(a))$ for all $a \in A$. It is called the *composition* of g and f.

Definition. Let $f : A \to B$ be a map. It is said to be *injective* if for all $a_1 \neq a_2$ in A we have $f(a_1) \neq f(a_2)$. It is said to be *surjective* if for each $b \in B$ there exists $a \in A$ such that $f(a) = b$. It is said to be *bijective* (or a bijection) if it is both injective and surjective. For $X \subseteq A$ we put

$$f(X) := \{f(x) : x \in X\} \subseteq B \qquad \text{(direct image of } X \text{ under } f\text{)}.$$

(There is a notational conflict here when X is both a subset of A and an element of A, but it will always be clear from the context when $f(X)$ is meant to be the the direct image of X under f; some authors resolve the conflict by denoting this direct image by $f[X]$ or in some other way.) We also call $f(A) = \{f(a) : a \in A\}$ the *image of f*. For $Y \subseteq B$ we put

$$f^{-1}(Y) := \{x \in A : f(x) \in Y\} \subseteq A \qquad \text{(inverse image of } Y \text{ under } f\text{)}.$$

Thus surjectivity of our map f is equivalent to $f(A) = B$.

---

[3] Sometimes we shall write fa instead of $f(a)$ in order to cut down on parentheses.

[4] Other words for "domain" and "codomain" are "source" and "target", respectively.

If $f : A \to B$ is a bijection then we have an *inverse* map $f^{-1} : B \to A$ given by

$$f^{-1}(b) := \text{ the unique } a \in A \text{ such that } f(a) = b.$$

Note that then $f^{-1} \circ f = 1_A$ and $f \circ f^{-1} = 1_B$. Conversely, if $f : A \to B$ and $g : B \to A$ satisfy $g \circ f = 1_A$ and $f \circ g = 1_B$, then f is a bijection with $f^{-1} = g$. (The attentive reader will notice that we just introduced a potential conflict of notation: for bijective $f : A \to B$ and $Y \subseteq B$, both the inverse image of Y under f and the direct image of Y under f^{-1} are denoted by $f^{-1}(Y)$; no harm is done, since these two subsets of A coincide.)

It follows from the definition of "map" that $f : A \to B$ and $g : C \to D$ are equal ($f = g$) if and only if $A = C$, $B = D$, and $f(x) = g(x)$ for all $x \in A$. We say that $g : C \to D$ *extends* $f : A \to B$ if $A \subseteq C$, $B \subseteq D$, and $f(x) = g(x)$ for all $x \in A$. [5]

Definition. A set A is said to be *finite* if there exists n and a bijection

$$f : \{1, \dots, n\} \to A.$$

Here we use $\{1, \dots, n\}$ as a suggestive notation for the set $\{m : 1 \le m \le n\}$. For $n = 0$ this is just $\emptyset$. If A is finite there is exactly one such n (although if $n > 1$ there will be more than one bijection $f : \{1, \dots, n\} \to A$); we call this unique n the *number of elements of* A or *the cardinality of* A, and denote it by $|A|$. A set which is not finite is said to be *infinite*.

Definition. A set A is said to be *countably infinite* if there is a bijection $\mathbf{N} \to A$. It is said to be *countable* if it is either finite or countably infinite.

Example. The sets $\mathbf{N}$, $\mathbf{Z}$ and $\mathbf{Q}$ are countably infinite, but the infinite set $\mathbf{R}$ is not countably infinite. Every infinite set has a countably infinite subset.

One of the standard axioms of set theory, the *Power Set Axiom* says:

For any set A, there is a set whose elements are exactly the subsets of A.

Such a set of subsets of A is clearly uniquely determined by A, is denoted by $\mathcal{P}(A)$, and is called the *power set of* A. If A is finite, so is $\mathcal{P}(A)$ and $|\mathcal{P}(A)| = 2^{|A|}$. Note that $a \mapsto \{a\} : A \to \mathcal{P}(A)$ is an injective map. However, there is no surjective map $A \to \mathcal{P}(A)$:

Cantor's Theorem. *Let $S : A \to \mathcal{P}(A)$ be a map. Then the set*

$$\{a \in A : a \notin S(a)\} \qquad (\textit{a subset of } A)$$

is not an element of $S(A)$.

Proof. Suppose otherwise. Then $\{a \in A : a \notin S(a)\} = S(b)$ where $b \in A$. Assuming $b \in S(b)$ yields $b \notin S(b)$, a contradiction. Thus $b \notin S(b)$; but then $b \in S(b)$, again a contradiction. This concludes the proof.

[5] We also say "$g : C \to D$ is an extension of $f : A \to B$" or "$f : A \to B$ is a restriction of $g : C \to D$."

Let I and A be sets. Then there is a set whose elements are exactly the maps $f : I \to A$, and this set is denoted by A^I. For $I = \{1, \dots, n\}$ we also write A^n instead of A^I. Thus an element of A^n is a map $a : \{1, \dots, n\} \to A$; we usually think of such an a as the n-tuple $(a(1), \dots, a(n))$, and we often write a_i instead of $a(i)$. So A^n can be thought of as the set of n-tuples $(a_1, \dots, a_n)$ with each $a_i \in A$. For $n = 0$ the set A^n has just one element — the empty tuple.

An *n-ary relation on A* is just a subset of A^n, and an *n-ary operation on A* is a map from A^n into A. Instead of "1-ary" we usually say "unary", and instead of "2-ary" we can say "binary". For example, $\{(a, b) \in \mathbf{Z}^2 : a < b\}$ is a binary relation on $\mathbf{Z}$, and integer addition is the binary operation $(a, b) \mapsto a + b$ on $\mathbf{Z}$.

Definition. $\{a_i\}_{i \in I}$ or $(a_i)_{i \in I}$ denotes a *family* of objects a_i indexed by the set I, and is just a suggestive notation for a set $\{(i, a_i) : i \in I\}$, not to be confused with the set $\{a_i : i \in I\}$. (There may be repetitions in the family, that is, it may happen that $a_i = a_j$ for distinct indices $i, j \in I$, but such repetition is not reflected in the set $\{a_i : i \in I\}$. For example, if $I = \mathbf{N}$ and $a_n = a$ for all n, then $\{(i, a_i) : i \in I\} = \{(i, a) : i \in \mathbf{N}\}$ is countably infinite, but $\{a_i : i \in I\} = \{a\}$ has just one element.) For $I = \mathbf{N}$ we usually say "sequence" instead of "family".

Given any family $(A_i)_{i \in I}$ of sets (that is, each A_i is a set) we have a set

$$\bigcup_{i \in I} A_i \ := \ \{x : x \in A_i \text{ for some } i \in I\},$$

the *union* of the family. If I is finite and each A_i is finite, then so is the union above and

$$|\bigcup_{i \in I} A_i| \leq \sum_{i \in I} |A_i|.$$

If I is countable and each A_i is countable then $\bigcup_{i \in I} A_i$ is countable.

Given any family $(A_i)_{i \in I}$ of sets we have a set

$$\prod_{i \in I} A_i \ := \{(a_i)_{i \in I} : \ a_i \in A_i \text{ for all } i \in I\},$$

the product of the family. One axiom of set theory, the Axiom of Choice, is a bit special, but we shall use it a few times. It says that for any family $(A_i)_{i \in I}$ of *nonempty* sets there is a family $(a_i)_{i \in I}$ such that $a_i \in A_i$ for all $i \in I$, that is, $\prod_{i \in I} A_i \neq \emptyset$.

Words

Definition. Let A be a set. Think of A as an *alphabet* of letters. A *word of length n on A* is an n-tuple $(a_1, \dots, a_n)$ of letters $a_i \in A$; because we think of it as a word (string of letters) we shall write this tuple instead as $a_1 \dots a_n$ (without parentheses or commas). There is a unique word of length 0 on A, the *empty word* and written ϵ. Given a word $a = a_1 \dots a_n$ of length $n \geq 1$ on A, the *first letter* (or first symbol) of a is by definition a_1, and the *last letter* (or last symbol) of a is a_n. The set of all words on A is denoted A^*:

$$A^* = \bigcup_n A^n \qquad \text{(disjoint union)}.$$

Logical expressions like formulas and terms will be introduced later as words of a special form on suitable alphabets. When $A \subseteq B$ we can identify A^* with a subset of B^*, and this will be done whenever convenient.

Definition. Given words $a = a_1 \ldots a_m$ and $b = b_1 \ldots b_n$ on A of length m and n respectively, we define their *concatenation* $ab \in A^*$:

$$ab = a_1 \ldots a_m b_1 \ldots b_n.$$

Thus ab is a word on A of length $m + n$. Concatenation is a binary operation on A^* that is associative: $(ab)c = a(bc)$ for all $a, b, c \in A^*$, with ϵ as two-sided identity: $\epsilon a = a = a\epsilon$ for all $a \in A^*$, and with two-sided cancellation: for all $a, b, c \in A^*$, if $ab = ac$, then $b = c$, and if $ac = bc$, then $a = b$.

Equivalence Relations and Quotient Sets

Given a binary relation R on a set A it is often more suggestive to write aRb instead of $(a, b) \in R$.

Definition. An *equivalence relation* on a set A is a binary relation $\sim$ on A such that for all $a, b, c \in A$:
(i) $a \sim a$ (reflexivity);
(ii) $a \sim b$ implies $b \sim a$ (symmetry);
(iii) ($a \sim b$ and $b \sim c$) implies $a \sim c$ (transitivity).

Example. Given any n we have the equivalence relation "congruence modulo n" on $\mathbf{Z}$ defined as follows: for any $a, b \in \mathbf{Z}$ we have

$$a \equiv b \mod n \iff a - b = nc \text{ for some } c \in \mathbf{Z}.$$

For $n = 0$ this is just equality on $\mathbf{Z}$.

Let $\sim$ be an equivalence relation on the set A. The *equivalence class* $a^\sim$ of an element $a \in A$ is defined by $a^\sim = \{b \in A : a \sim b\}$ (a subset of A). For $a, b \in A$ we have $a^\sim = b^\sim$ if and only if $a \sim b$, and $a^\sim \cap b^\sim = \emptyset$ if and only if $a \nsim b$. The *quotient set of A by* $\sim$ is by definition the set of equivalence classes:

$$A/{\sim} = \{a^\sim : a \in A\}.$$

This quotient set is a *partition of* A, that is, it is a collection of pairwise disjoint nonempty subsets of A whose union is A. (*Collection* is a synonym for *set*; we use it here because we don't like to say "set of ... subsets ...".) Every partition of A is the quotient set $A/{\sim}$ for a unique equivalence relation $\sim$ on A. Thus equivalence relations on A and partitions of A are just different ways to describe the same situation.

In the previous example (congruence modulo n) the equivalence classes are called *congruence classes modulo* n (or residue classes modulo n) and the corresponding quotient set is often denoted $\mathbf{Z}/n\mathbf{Z}$.

Remark. Readers familiar with some abstract algebra will note that the construction in the example above is a special case of a more general construction—that of a quotient of a group with respect to a normal subgroup.

Posets

A *partially ordered set* (short: *poset*) is a pair $(P, \leq)$ consisting of a set P and a partial ordering $\leq$ on P, that is, $\leq$ is a binary relation on P such that for all $p, q, r \in P$:

(i) $p \leq p$ (reflexivity);

(ii) if $p \leq q$ and $q \leq p$, then $p = q$ (antisymmetry);

(iii) if $p \leq q$ and $q \leq r$, then $p \leq r$ (transitivity).

If in addition we have for all $p, q \in P$,

(iv) $p \leq q$ or $q \leq p$,

then we say that $\leq$ is a *linear order* on P, or that $(P, \leq)$ is a *linearly ordered* set.[6] Each of the sets $\mathbf{N}, \mathbf{Z}, \mathbf{Q}, \mathbf{R}$ comes with its familiar linear order on it.

As an example, take any set A and its collection $\mathcal{P}(A)$ of subsets. Then

$$X \leq Y :\Longleftrightarrow X \subseteq Y \quad \text{(for subsets } X, Y \text{ of } A\text{)}$$

defines a poset $(\mathcal{P}(A), \leq)$, also referred to as the *power set of* A *ordered by inclusion*. This is not a linearly ordered set if A has more than one element.

Finite linearly ordered sets are determined "up to unique isomorphism" by their size: if $(P, \leq)$ is a linearly ordered set and $|P| = n$, then there is a unique map $\iota : P \to \{1, \ldots, n\}$ such that for all $p, q \in P$ we have: $p \leq q \iff \iota(p) \leq \iota(q)$. This map ι is a bijection.

Let $(P, \leq)$ be a poset. Here is some useful notation. For $x, y \in P$ we set

$$\begin{aligned} &x \geq y : \Longleftrightarrow y \leq x, \\ &x < y : \Longleftrightarrow y > x :\Longleftrightarrow x \leq y \text{ and } x \neq y. \end{aligned}$$

Note that $(P, \geq)$ is also a poset. A *least* element of P is a $p \in P$ such that $p \leq x$ for all $x \in P$; a *largest* element of P is defined likewise, with $\geq$ instead of $\leq$. Of course, P can have at most one least element; therefore we can refer to *the*

[6] One also uses the term *total order* instead of *linear order*.

least element of P, if P has a least element; likewise, we can refer to *the* largest element of P, if P has a largest element.

A *minimal* element of P is a $p \in P$ such that there is no $x \in P$ with $x < p$; a *maximal* element of P is defined likewise, with $>$ instead of $<$. If P has a least element, then this element is also the unique minimal element of P; some posets, however, have more than one minimal element. The reader might want to prove the following result to get a feeling for these notions:

If P is finite and nonempty, then P has a maximal element, and there is a linear order $\leq'$ on P that extends $\leq$ in the sense that

$$p \leq q \implies p \leq' q, \quad \text{for all } p, q \in P.$$

(Hint: use induction on $|P|$.)

Let $X \subseteq P$. A *lowerbound* (respectively, *upperbound*) of X in P is an element $l \in P$ (respectively, an element $u \in P$), such that $l \leq x$ for all $x \in X$ (respectively, $x \leq u$ for all $x \in X$).

We often tacitly consider X as a poset in its own right, by restricting the given partial ordering of P to X. More precisely this means that we consider the poset $(X, \leq_X)$ where the partial ordering $\leq_X$ on X is defined by

$$x \leq_X y \iff x \leq y \qquad (x, y \in X).$$

Thus we can speak of least, largest, minimal, and maximal elements of a set $X \subseteq P$, when the ambient poset $(P, \leq)$ is clear from the context. For example, when X is the collection of *nonempty* subsets of a set A and X is ordered by inclusion, then the minimal elements of X are the singletons $\{a\}$ with $a \in A$. We call X a *chain* in P if $(X, \leq_X)$ is linearly ordered.

Occasionally we shall use the following fact about posets $(P, \leq)$.

Zorn's Lemma. *Suppose P is nonempty and every nonempty chain in P has an upperbound in P. Then P has a maximal element.*

For a further discussion of Zorn's Lemma and its proof using the Axiom of Choice we refer the reader to Halmos's book on set theory.

Chapter 2

Basic Concepts of Logic

2.1 Propositional Logic

Propositional logic is the fragment of logic where we construct new statements from given statements using so-called connectives like "not", "or" and "and". The truth value of such a new statement is then completely determined by the truth values of the given statements. Thus, given any statements p and q, we can form the three statements

$$
\begin{array}{rl}
\neg p & \text{(the \textit{negation} of } p\text{, pronounced as “not } p\text{”),} \\
p \vee q & \text{(the \textit{disjunction} of } p \text{ and } q\text{, pronounced as “}p \text{ or } q\text{”),} \\
p \wedge q & \text{(the \textit{conjunction} of } p \text{ and } q\text{, pronounced as “}p \text{ and } q\text{”).}
\end{array}
$$

This leads to more complicated combinations like $\neg\big(p \wedge (\neg q)\big)$. We shall regard $\neg p$ as true if and only if p is not true; also, $p \vee q$ is defined to be true if and only if p is true or q is true (including the possibility that both are true), and $p \wedge q$ is deemed to be true if and only if p is true and q is true. Instead of "not true" we also say "false". We now introduce a formalism that makes this into mathematics.

We start with the five distinct symbols

$$\top \qquad \perp \qquad \neg \qquad \vee \qquad \wedge$$

to be thought of as *true, false, not, or*, and *and*, respectively. These symbols are fixed throughout the course, and are called *propositional connectives.* In this section we also fix a set A whose elements will be called *propositional atoms* (or just atoms), such that no propositional connective is an atom. It may help the reader to think of an atom a as a variable for which we can substitute arbitrary statements, assumed to be either true or false.

A *proposition on A* is a word on the alphabet $A \cup \{\top, \perp, \neg, \vee, \wedge\}$ that can be obtained by applying the following rules:

(i) each atom $a \in A$ (viewed as a word of length 1) is a proposition on A;
(ii) $\top$ and $\bot$ (viewed as words of length 1) are propositions on A;
(iii) if p and q are propositions on A, then the concatenations $\neg p$, $\vee pq$ and $\wedge pq$ are propositions on A.

For the rest of this section "proposition" means "proposition on A", and p, q, r (sometimes with subscripts) will denote propositions.

Example. Suppose a, b, c are atoms. Then $\wedge \vee \neg ab \neg c$ is a proposition. This follows from the rules above: a is a proposition, so $\neg a$ is a proposition, hence $\vee \neg ab$ as well; also $\neg c$ is a proposition, and thus $\wedge \vee \neg ab \neg c$ is a proposition.

We defined "proposition" using the suggestive but vague phrase "can be obtained by applying the following rules". The reader should take such an informal description as shorthand for a completely explicit definition, which in the case at hand is as follows:

A *proposition* is a word w on the alphabet $A \cup \{\top, \bot, \neg, \vee, \wedge\}$ for which there is a sequence $w_1, \ldots, w_n$ of words on that same alphabet, with $n \geq 1$, such that $w = w_n$ and for each $k \in \{1, \ldots, n\}$, either $w_k \in A \cup \{\top, \bot\}$ (where each element in the last set is viewed as a word of length 1), or there are $i, j \in \{1, \ldots, k-1\}$ such that w_k is one of the concatenations $\neg w_i$, $\vee w_i w_j$, $\wedge w_i w_j$.

We let $\mathrm{Prop}(A)$ denote the set of propositions.

Remark. Having the connectives $\vee$ and $\wedge$ in *front* of the propositions they "connect" rather than in between, is called *prefix notation* or *Polish notation*. This is theoretically elegant, but for the sake of readability we usually write $p \vee q$ and $p \wedge q$ to denote $\vee pq$ and $\wedge pq$ respectively, and we also use parentheses and brackets if this helps to clarify the structure of a proposition. So the proposition in the example above could be denoted by $[(\neg a) \vee b] \wedge (\neg c)$, or even by $(\neg a \vee b) \wedge \neg c$ since we shall agree that $\neg$ binds stronger than $\vee$ and $\wedge$ in this informal way of indicating propositions. Because of the informal nature of these conventions, we don't have to give precise rules for their use; it's enough that each actual use is clear to the reader.

The *intended* structure of a proposition—how we think of it as built up from atoms via connectives—is best exhibited in the form of a tree, a two-dimensional array, rather than as a one-dimensional string. Such trees, however, occupy valuable space on the printed page, and are typographically demanding. Fortunately, our "official" prefix notation does uniquely determine the intended structure of a proposition: that is what the next lemma amounts to.

Lemma 2.1.1 (Unique Readability). *If p has length 1, then either $p = \top$, or $p = \bot$, or p is an atom. If p has length > 1, then its first symbol is either $\neg$, or $\vee$, or $\wedge$. If the first symbol of p is $\neg$, then $p = \neg q$ for a unique q. If the first symbol of p is $\vee$, then $p = \vee qr$ for a unique pair (q, r). If the first symbol of p is $\wedge$, then $p = \wedge qr$ for a unique pair (q, r).*

(Note that we used here our convention that p, q, r denote propositions.) Only the last two claims are worth proving in print, the others should require only a moment's thought. For now we shall assume this lemma without proof. At the end of this section we establish more general results which are needed also later in the course.

Remark. Rather than thinking of a proposition as a statement, it's better viewed as a *function* whose *arguments* and *values* are statements: replacing the atoms in a proposition by specific mathematical statements like "$2 \times 2 = 4$", "$\pi^2 < 7$", and "every even integer > 2 is the sum of two prime numbers", we obtain again a mathematical statement.

We shall use the following notational conventions: $p \to q$ denotes $\neg p \vee q$, and $p \leftrightarrow q$ denotes $(p \to q) \wedge (q \to p)$. By recursion on n we define

$$p_1 \vee \ldots \vee p_n = \begin{cases} \perp & \text{if } n = 0 \\ p_1 & \text{if } n = 1 \\ p_1 \vee p_2 & \text{if } n = 2 \\ (p_1 \vee \ldots \vee p_{n-1}) \vee p_n & \text{if } n > 2 \end{cases}$$

Thus $p \vee q \vee r$ stands for $(p \vee q) \vee r$. We call $p_1 \vee \ldots \vee p_n$ the *disjunction* of $p_1, \ldots, p_n$. The reason that for $n = 0$ we take this disjunction to be $\perp$ is that we want a disjunction to be true if and only if (at least) one of the disjuncts is true.

Similarly, the *conjunction* $p_1 \wedge \ldots \wedge p_n$ of $p_1, \ldots, p_n$ is defined by replacing everywhere $\vee$ by $\wedge$ and $\perp$ by $\top$ in the definition of $p_1 \vee \ldots \vee p_n$.

Definition. A *truth assignment* is a map $t : A \to \{0, 1\}$. We extend such a t to $\hat{t} : \mathrm{Prop}(A) \to \{0, 1\}$ by requiring
(i) $\hat{t}(\top) = 1$
(ii) $\hat{t}(\perp) = 0$
(iii) $\hat{t}(\neg p) = 1 - \hat{t}(p)$
(iv) $\hat{t}(p \vee q) = \max(\hat{t}(p), \hat{t}(q))$
(v) $\hat{t}(p \wedge q) = \min(\hat{t}(p), \hat{t}(q))$

Note that there is exactly one such extension $\hat{t}$ by unique readability. To simplify notation we often write t instead of $\hat{t}$. The array below is called a truth table. It shows on each row below the top row how the two leftmost entries $t(p)$ and $t(q)$ determine $t(\neg p)$, $t(p \vee q)$, $t(p \wedge q)$, $t(p \to q)$ and $t(p \leftrightarrow q)$.

p	q	$\neg p$	$p \vee q$	$p \wedge q$	$p \to q$	$p \leftrightarrow q$
0	0	1	0	0	1	1
0	1	1	1	0	1	0
1	0	0	1	0	0	0
1	1	0	1	1	1	1

Let $t : A \to \{0, 1\}$. Note that $t(p \to q) = 1$ if and only if $t(p) \leq t(q)$, and that $t(p \leftrightarrow q) = 1$ if and only if $t(p) = t(q)$.

Suppose $a_1, \ldots, a_n$ are the distinct atoms that occur in p, and we know how p is built up from those atoms. Then we can compute in a finite number of steps

$t(p)$ from $t(a_1), \dots, t(a_n)$. In particular, $t(p) = t'(p)$ for any $t' : A \to \{0,1\}$ such that $t(a_i) = t'(a_i)$ for $i = 1, \dots, n$.

Definition. We say that p is a *tautology* (notation: $\models p$) if $t(p) = 1$ for all $t : A \to \{0,1\}$. We say that p is *satisfiable* if $t(p) = 1$ for some $t : A \to \{0,1\}$.

Thus $\top$ is a tautology, and $p \vee \neg p$, $p \to (p \vee q)$ are tautologies for all p and q. By the remark preceding the definition one can verify whether any given p with exactly n distinct atoms in it is a tautology by computing 2^n numbers and checking that these numbers all come out 1. (To do this accurately by hand is already cumbersome for $n = 5$, but computers can handle somewhat larger n. Fortunately, other methods are often efficient for special cases.)

Remark. Note that $\models p \leftrightarrow q$ iff $t(p) = t(q)$ for all $t : A \to \{0,1\}$. We call p *equivalent to* q if $\models p \leftrightarrow q$. Note that "equivalent to" defines an equivalence relation on $\mathrm{Prop}(A)$. The lemma below gives a useful list of equivalences. We leave it to the reader to verify them.

Lemma 2.1.2. *For all p, q, r we have the following equivalences:*

(1) $\models (p \vee p) \leftrightarrow p$, $\quad \models (p \wedge p) \leftrightarrow p$

(2) $\models (p \vee q) \leftrightarrow (q \vee p)$, $\quad \models (p \wedge q) \leftrightarrow (q \wedge p)$

(3) $\models (p \vee (q \vee r)) \leftrightarrow ((p \vee q) \vee r)$, $\quad \models (p \wedge (q \wedge r)) \leftrightarrow ((p \wedge q) \wedge r)$

(4) $\models (p \vee (q \wedge r)) \leftrightarrow (p \vee q) \wedge (p \vee r)$, $\quad \models (p \wedge (q \vee r)) \leftrightarrow (p \wedge q) \vee (p \wedge r)$

(5) $\models (p \vee (p \wedge q)) \leftrightarrow p$, $\quad \models (p \wedge (p \vee q)) \leftrightarrow p$

(6) $\models (\neg(p \vee q)) \leftrightarrow (\neg p \wedge \neg q)$, $\quad \models (\neg(p \wedge q)) \leftrightarrow (\neg p \vee \neg q)$

(7) $\models (p \vee \neg p) \leftrightarrow \top$, $\quad \models (p \wedge \neg p) \leftrightarrow \bot$

(8) $\models \neg\neg p \leftrightarrow p$

Items (1), (2), (3), (4), (5), and (6) are often referred to as *the idempotent law*, *commutativity*, *associativity*, *distributivity*, *the absorption law*, and *the De Morgan law*, respectively. Note the left-right symmetry in (1)–(7) : the so-called *duality* of propositional logic. We shall return to this issue in the more algebraic setting of *boolean algebras*.

Some notation: let $(p_i)_{i \in I}$ be a family of propositions with finite index set I, choose a bijection $k \mapsto i(k) : \{1, \dots, n\} \to I$ and set

$$\bigvee_{i \in I} p_i := p_{i(1)} \vee \cdots \vee p_{i(n)}, \quad \bigwedge_{i \in I} p_i := p_{i(1)} \wedge \cdots \wedge p_{i(n)}.$$

If I is clear from context we just write $\bigvee_i p_i$ and $\bigwedge_i p_i$ instead. Of course, the notations $\bigvee_{i \in I} p_i$ and $\bigwedge_{i \in I} p_i$ can only be used when the particular choice of bijection of $\{1, \dots, n\}$ with I does not matter; this is usually the case, because the equivalence class of $p_{i(1)} \vee \cdots \vee p_{i(n)}$ does not depend on this choice, and the same is true for the equivalence class of $p_{i(1)} \wedge \cdots \wedge p_{i(n)}$.

Next we define "model of Σ" and "tautological consequence of Σ".

Definition. Let $\Sigma \subseteq \mathrm{Prop}(A)$. A *model of* Σ is a truth assignment $t : A \to \{0,1\}$ such that $t(\sigma) = 1$ for all $\sigma \in \Sigma$. We say p is a *tautological consequence* of Σ (written $\Sigma \models p$) if $t(p) = 1$ for every model t of Σ. Note that $\models p$ is the same as $\emptyset \models p$.)

Lemma 2.1.3. *Let* $\Sigma \subseteq \text{Prop}(A)$ *and* $p, q \in \text{Prop}(A)$. *Then*
(1) $\Sigma \models p \wedge q \iff \Sigma \models p$ *and* $\Sigma \models q$.
(2) $\Sigma \models p \implies \Sigma \models p \vee q$.
(3) $\Sigma \cup \{p\} \models q \iff \Sigma \models p \to q$.
(4) *If* $\Sigma \models p$ *and* $\Sigma \models p \to q$, *then* $\Sigma \models q$. (Modus Ponens.)

Proof. We will prove (3) here and leave the rest as exercise.
($\Rightarrow$) Assume $\Sigma \cup \{p\} \models q$. To derive $\Sigma \models p \to q$ we consider any model $t : A \longrightarrow \{0,1\}$ of Σ, and need only show that then $t(p \to q) = 1$. If $t(p) = 1$ then $t(\Sigma \cup \{p\}) \subseteq \{1\}$, hence $t(q) = 1$ and thus $t(p \to q) = 1$. If $t(p) = 0$ then $t(p \to q) = 1$ by definition.
($\Leftarrow$) Assume $\Sigma \models p \to q$. To derive $\Sigma \cup \{p\} \models q$ we consider any model $t : A \longrightarrow \{0,1\}$ of $\Sigma \cup \{p\}$, and need only derive that $t(q) = 1$. By assumption $t(p \to q) = 1$ and in view of $t(p) = 1$, this gives $t(q) = 1$ as required. $\square$

We finish this section with the promised general result on unique readability. We also establish facts of similar nature that are needed later.

Definition. Let F be a set of symbols with a function $a : F \to \mathbf{N}$ (called the *arity function*). A symbol $f \in F$ is said to have *arity* n if $a(f) = n$. A word on F is said to be *admissible* if it can be obtained by applying the following rules:
(i) If $f \in F$ has arity 0, then f viewed as a word of length 1 is admissible.
(ii) If $f \in F$ has arity $m > 0$ and $t_1, \ldots, t_m$ are admissible words on F, then the concatenation $ft_1 \ldots t_m$ is admissible.

Below we just write "admissible word" instead of "admissible word on F". Note that the empty word is not admissible, and that the last symbol of an admissible word cannot be of arity > 0.

Example. Take $F = A \cup \{\top, \bot, \neg, \vee, \wedge\}$ and define $\text{arity} : F \to \mathbf{N}$ by

$$\text{arity}(x) = 0 \text{ for } x \in A \cup \{\top, \bot\}, \quad \text{arity}(\neg) = 1, \quad \text{arity}(\vee) = \text{arity}(\wedge) = 2.$$

Then the set of admissible words is just $\text{Prop}(A)$.

Lemma 2.1.4. *Let* $t_1, \ldots, t_m$ *and* $u_1, \ldots, u_n$ *be admissible words and* w *any word on* F *such that* $t_1 \ldots t_m w = u_1 \ldots u_n$. *Then* $m \le n$, $t_i = u_i$ *for* $i = 1, \ldots, m$, *and* $w = u_{m+1} \cdots u_n$.

Proof. By induction on the length of $u_1 \ldots u_n$. If this length is 0, then $m = n = 0$ and w is the empty word. Suppose the length is > 0, and assume the lemma holds for smaller lengths. Note that $n > 0$. If $m = 0$, then the conclusion of the lemma holds, so suppose $m > 0$. The first symbol of t_1 equals the first symbol of u_1. Say this first symbol is $h \in F$ with arity k. Then $t_1 = ha_1 \ldots a_k$ and $u_1 = hb_1 \ldots b_k$ where $a_1, \ldots, a_k$ and $b_1, \ldots, b_k$ are admissible words. Cancelling the first symbol h gives

$$a_1 \ldots a_k t_2 \ldots t_m w = b_1 \ldots b_k u_2 \ldots u_n.$$

(Caution: any of $k, m-1, n-1$ could be 0.) We have $\text{length}(b_1 \dots b_k u_2 \dots u_n) = \text{length}(u_1 \dots u_n) - 1$, so the induction hypothesis applies. It yields $k + m - 1 \le k + n - 1$ (so $m \le n$), $a_1 = b_1, \dots, a_k = b_k$ (so $t_1 = u_1$), $t_2 = u_2, \dots, t_m = u_m$, and $w = u_{m+1} \cdots u_n$. □

Here are two immediate consequences that we shall use:

1. Let $t_1, \dots, t_m$ and $u_1, \dots, u_n$ be admissible words such that $t_1 \dots t_m = u_1 \dots u_n$. Then $m = n$ and $t_i = u_i$ for $i = 1, \dots, m$.

2. Let t and u be admissible words and w a word on F such that $tw = u$. Then $t = u$ and w is the empty word.

Lemma 2.1.5 (Unique Readability).
Each admissible word equals $ft_1 \dots t_m$ for a unique tuple $(f, t_1, \dots, t_m)$ where $f \in F$ has arity m and $t_1, \dots, t_m$ are admissible words.

Proof. Suppose $ft_1 \dots t_m = gu_1 \dots u_n$ where $f, g \in F$ have arity m and n respectively, and $t_1, \dots, t_m, u_1, \dots, u_n$ are admissible words on F. We have to show that then $f = g$, $m = n$ and $t_i = u_i$ for $i = 1, \dots, m$. Observe first that $f = g$ since f and g are the first symbols of two equal words. After cancelling the first symbol of both words, the first consequence of the previous lemma leads to the desired conclusion. □

Given words $v, w \in F^*$ and $i \in \{1, \dots, \text{length}(w)\}$, we say that v *occurs in w at starting position i* if $w = w_1 v w_2$ where $w_1, w_2 \in F^*$ and w_1 has length $i - 1$. (For example, if $f, g \in F$ are distinct, then the word fgf has exactly two occurrences in the word $fgfgf$, one at starting position 1, and the other at starting position 3; these two occurrences overlap, but such overlapping is impossible with *admissible* words, see exercise 5 at the end of this section.) Given $w = w_1 v w_2$ as above, and given $v' \in F^*$, *the result of replacing v in w at starting position i by v'* is by definition the word $w_1 v' w_2$.

Lemma 2.1.6. *Let w be an admissible word and $1 \le i \le \text{length}(w)$. Then there is a unique admissible word that occurs in w at starting position i.*

Proof. We prove existence by induction on $\text{length}(w)$. Uniqueness then follows from the fact stated just before Lemma 2.1.5. Clearly w is an admissible word occurring in w at starting position 1. Suppose $i > 1$. Then we write $w = ft_1 \dots t_n$ where $f \in F$ has arity $n > 0$, and $t_1, \dots, t_n$ are admissible words, and we take $j \in \{1, \dots, n\}$ such that

$$1 + \text{length}(t_1) + \cdots + \text{length}(t_{j-1}) < i \le 1 + \text{length}(t_1) + \cdots + \text{length}(t_j).$$

Now apply the inductive assumption to t_j. □

Remark. Let $w = ft_1 \dots t_n$ where $f \in F$ has arity $n > 0$, and $t_1, \dots, t_n$ are admissible words. Put $l_j := 1 + \text{length}(t_1) + \cdots + \text{length}(t_j)$ for $j = 0, \dots, n$ (so $l_0 = 1$). Suppose $l_{j-1} < i \le l_j$, $1 \le j \le n$, and let v be the admissible word that occurs in w at starting position i. Then the proof of the last lemma shows that this occurrence is entirely inside t_j, that is, $i - 1 + \text{length}(v) \le l_j$.

Corollary 2.1.7. *Let w be an admissible word and $1 \le i \le \text{length}(w)$. Then the result of replacing the admissible word v in w at starting position i by an admissible word v' is again an admissible word.*

This follows by a routine induction on $\text{length}(w)$, using the last remark.

Exercises. In the exercises below, $A = \{a_1, \dots, a_n\}$, $|A| = n$.

(1) (*Disjunctive Normal Form*) Each p is equivalent to a disjunction

$$p_1 \vee \cdots \vee p_k$$

where each disjunct p_i is a conjunction $a_1^{\epsilon_1} \wedge \ldots \wedge a_n^{\epsilon_n}$ with all $\epsilon_j \in \{-1, 1\}$ and where for an atom a we put $a^1 := a$ and $a^{-1} := \neg a$.

(2) (*Conjunctive Normal Form*) Same as last problem, except that the signs $\vee$ and $\wedge$ are interchanged, as well as the words "disjunction" and "conjunction," and also the words "disjunct" and "conjunct."

(3) To each p associate the function $f_p : \{0,1\}^A \to \{0,1\}$ defined by $f_p(t) = t(p)$. (Think of a truth table for p where the 2^n rows correspond to the 2^n truth assignments $t : A \to \{0,1\}$, and the column under p records the values $t(p)$.) Then for every function $f : \{0,1\}^A \to \{0,1\}$ there is a p such that $f = f_p$.

(4) Let $\sim$ be the equivalence relation on $\text{Prop}(A)$ given by

$$p \sim q \;:\Longleftrightarrow\; \models p \leftrightarrow q.$$

Then the quotient set $\text{Prop}(A)/\sim$ is finite; determine its cardinality as a function of $n = |A|$.

(5) Let w be an admissible word and $1 \le i < i' \le \text{length}(w)$. Let v and v' be the admissible words that occur at starting positions i and i' respectively in w. Then these occurrences are either nonoverlapping, that is, $i - 1 + \text{length}(v) < i'$, or the occurrence of v' is entirely inside that of v, that is,

$$i' - 1 + \text{length}(v') \le i - 1 + \text{length}(v).$$

2.2 Completeness for Propositional Logic

In this section we introduce a *proof system* for propositional logic, state the completeness of this proof system, and then prove this completeness.

As in the previous section we fix a set A of atoms, and the conventions of that section remain in force.

A *propositional axiom* is by definition a proposition that occurs in the list below, for some choice of p, q, r:

1. $\top$
2. $p \to (p \vee q)$; $\qquad p \to (q \vee p)$
3. $\neg p \to \big(\neg q \to \neg(p \vee q)\big)$

4. $(p \wedge q) \to p; \qquad (p \wedge q) \to q$
5. $p \to \big(q \to (p \wedge q)\big)$
6. $\big(p \to (q \to r)\big) \to \big((p \to q) \to (p \to r)\big)$
7. $p \to (\neg p \to \bot)$
8. $(\neg p \to \bot) \to p$

Each of items 2–8 describes infinitely many propositional axioms. That is why we do not call these items *axioms*, but *axiom schemes*. For example, if $a, b \in A$, then $a \to (a \vee \bot)$ and $b \to (b \vee (\neg a \wedge \neg b))$ are distinct propositional axioms, and both *instances* of axiom scheme 2. It is easy to check that all propositional axioms are tautologies.

Here is our single rule of inference for propositional logic:

Modus Ponens (*MP*): from p and $p \to q$, infer q.

In the rest of this section Σ denotes a set of propositions, that is, $\Sigma \subseteq \mathrm{Prop}(A)$.

Definition. A *formal proof*, or just *proof*, of p from Σ is a sequence $p_1, \dots, p_n$ with $n \geq 1$ and $p_n = p$, such that for $k = 1, \dots, n$:
(i) either $p_k \in \Sigma$,
(ii) or p_k is a propositional axiom,
(iii) or there are $i, j \in \{1, \dots, k-1\}$ such that p_k can be inferred from p_i and p_j by MP.

If there exists a proof of p from Σ, then we write $\Sigma \vdash p$, and say Σ *proves* p. For $\Sigma = \emptyset$ we also write $\vdash p$ instead of $\Sigma \vdash p$.

Lemma 2.2.1. $\vdash p \to p$.

Proof. The proposition $p \to \big((p \to p) \to p\big)$ is a propositional axiom by axiom scheme 2. By axiom scheme 6,

$$\{p \to \big((p \to p) \to p\big)\} \to \{\big(p \to (p \to p)\big) \to (p \to p)\}$$

is a propositional axiom. Applying MP to these two axioms yields

$$\vdash \big(p \to (p \to p)\big) \to (p \to p).$$

Since $p \to (p \to p)$ is also a propositional axiom by scheme 2, we can apply MP again to obtain $\vdash p \to p$. $\square$

Proposition 2.2.2. *If $\Sigma \vdash p$, then $\Sigma \models p$.*

This should be clear from earlier facts that we stated and which the reader was asked to verify. The converse is true but less obvious:

Theorem 2.2.3 (Completeness - first form).

$$\Sigma \vdash p \iff \Sigma \models p$$

Remark. There is some arbitrariness in our choice of axioms and rule, and thus in our notion of formal proof. This is in contrast to the definition of $\models$, which merely formalizes the basic underlying idea of propositional logic as stated in the introduction to the previous section. However, the equivalence of $\vdash$ and $\models$ (Completeness Theorem) shows that our choice of axioms and rule yields a *complete proof system*. Moreover, this equivalence has consequences which can be stated in terms of $\models$ alone. An example is the Compactness Theorem.

Theorem 2.2.4 (Compactness of Propositional Logic). *If* $\Sigma \models p$ *then there is a finite subset* Σ_0 *of* Σ *such that* $\Sigma_0 \models p$.

It is convenient to prove first a variant of the Completeness Theorem.

Definition. We say that Σ is *inconsistent* if $\Sigma \vdash \perp$, and otherwise (that is, if $\Sigma \nvdash \perp$) we call Σ *consistent*.

Theorem 2.2.5 (Completeness - second form).
Σ *is consistent if and only if* Σ *has a model.*

From this second form of the Completenenes Theorem we obtain easily an alternative form of the Compactness of Propositional Logic:

Corollary 2.2.6. Σ *has a model* $\iff$ *every finite subset of* Σ *has a model.*

We first show that the second form of the Completeness Theorem implies the first form. For this we need a technical lemma that will also be useful later in the course.

Lemma 2.2.7 (Deduction Lemma). *Suppose* $\Sigma \cup \{p\} \vdash q$. *Then* $\Sigma \vdash p \to q$.

Proof. By induction on proofs.

If q is a propositional axiom, then $\Sigma \vdash q$, and since $q \to (p \to q)$ is a propositional axiom, MP yields $\Sigma \vdash p \to q$. If $q \in \Sigma \cup \{p\}$, then *either* $q \in \Sigma$ in which case the same argument as before gives $\Sigma \vdash p \to q$, *or* $q = p$ and then $\Sigma \vdash p \to q$ since $\vdash p \to p$ by the lemma above.

Now assume that q is obtained by MP from r and $r \to q$, where $\Sigma \cup \{p\} \vdash r$ and $\Sigma \cup \{p\} \vdash r \to q$ and where we assume inductively that $\Sigma \vdash p \to r$ and $\Sigma \vdash p \to (r \to q)$. Then we obtain $\Sigma \vdash p \to q$ from the propositional axiom

$$\big(p \to (r \to q)\big) \to \big((p \to r) \to (p \to q)\big)$$

by applying MP twice. □

Corollary 2.2.8. $\Sigma \vdash p$ *if and only if* $\Sigma \cup \{\neg p\}$ *is inconsistent.*

Proof. ($\Rightarrow$) Assume $\Sigma \vdash p$. Since $p \to (\neg p \to \perp)$ is a propositional axiom, we can apply MP twice to get $\Sigma \cup \{\neg p\} \vdash \perp$. Hence $\Sigma \cup \{\neg p\}$ is inconsistent.
($\Leftarrow$) Assume $\Sigma \cup \{\neg p\}$ is inconsistent. Then $\Sigma \cup \{\neg p\} \vdash \perp$, and so by the Deduction Lemma we have $\Sigma \vdash \neg p \to \perp$. Since $(\neg p \to \perp) \to p$ is a propositional axiom, MP yields $\Sigma \vdash p$. □

Corollary 2.2.9. *The second form of Completeness (Theorem 2.2.5) implies the first form (Theorem 2.2.3).*

Proof. Assume the second form of Completeness holds, and that $\Sigma \models p$. We want to show that then $\Sigma \vdash p$. From $\Sigma \models p$ it follows that $\Sigma \cup \{\neg p\}$ has no model. Hence by the second form of Completeness, the set $\Sigma \cup \{\neg p\}$ is inconsistent. Then by Corollary 2.2.8 we have $\Sigma \vdash p$. $\square$

Definition. We say that Σ is *complete* if Σ is consistent, and for each p either $\Sigma \vdash p$ or $\Sigma \vdash \neg p$.

Completeness as a property of a set of propositions should not be confused with the completeness of our proof system as expressed by the Completeness Theorem. (It is just a historical accident that we use the same word.)

Below we use Zorn's Lemma to show that any consistent set of propositions can be extended to a complete set of propositions.

Lemma 2.2.10 (Lindenbaum). *Suppose Σ is consistent. Then $\Sigma \subseteq \Sigma'$ for some complete $\Sigma' \subseteq \mathrm{Prop}(A)$.*

Proof. Let P be the collection of all consistent subsets of $\mathrm{Prop}(A)$ that contain Σ. In particular $\Sigma \in P$. We consider P as partially ordered by inclusion. Any totally ordered subcollection $\{\Sigma_i : i \in I\}$ of P with $I \neq \emptyset$ has an upper bound in P, namely $\bigcup\{\Sigma_i : i \in I\}$. (To see this it suffices to check that $\bigcup\{\Sigma_i : i \in I\}$ is consistent. Suppose otherwise, that is, suppose $\bigcup\{\Sigma_i : i \in I\} \vdash \perp$. Since a proof can use only finitely many of the axioms in $\bigcup\{\Sigma_i : i \in I\}$, there exists $i \in I$ such that $\Sigma_i \vdash \perp$, contradicting the consistency of Σ_i.)
Thus by Zorn's lemma P has a maximal element Σ'. We claim that then Σ' is complete. For any p, if $\Sigma' \nvdash p$, then by Corollary 2.2.8 the set $\Sigma' \cup \{\neg p\}$ is consistent, hence $\neg p \in \Sigma'$ by maximality of Σ', and thus $\Sigma' \vdash \neg p$. $\square$

Suppose A is countable. For this case we can give a proof of Lindenbaum's Lemma without using Zorn's Lemma as follows.

Proof. Because A is countable, $\mathrm{Prop}(A)$ is countable. Take an enumeration $(p_n)_{n \in \mathbb{N}}$ of $\mathrm{Prop}(A)$. We construct an increasing sequence $\Sigma = \Sigma_0 \subseteq \Sigma_1 \subseteq \dots$ of consistent subsets of $\mathrm{Prop}(A)$ as follows. Given a consistent $\Sigma_n \subseteq \mathrm{Prop}(A)$ we define

$$\Sigma_{n+1} = \begin{cases} \Sigma_n \cup \{p_n\} & \text{if } \Sigma_n \vdash p_n, \\ \Sigma_n \cup \{\neg p_n\} & \text{if } \Sigma_n \nvdash p_n, \end{cases}$$

so Σ_{n+1} remains consistent by earlier results. Thus $\Sigma_\infty := \bigcup\{\Sigma_n : n \in \mathbf{N}\}$ is consistent and also complete: for any n either $p_n \in \Sigma_{n+1} \subseteq \Sigma_\infty$ or $\neg p_n \in \Sigma_{n+1} \subseteq \Sigma_\infty$. $\square$

Define the truth assignment $t_\Sigma : A \to \{0, 1\}$ by

$$t_\Sigma(a) = 1 \text{ if } \Sigma \vdash a, \text{ and } t_\Sigma(a) = 0 \text{ otherwise.}$$

Lemma 2.2.11. *Suppose Σ is complete. Then for each p we have*

$$\Sigma \vdash p \iff t_\Sigma(p) = 1.$$

In particular, t_Σ is a model of Σ.

Proof. We proceed by induction on the number of connectives in p. If p is an atom or $p = \top$ or $p = \bot$, then the equivalence follows immediately from the definitions. It remains to consider the three cases below.

Case 1: $p = \neg q$, and (inductive assumption) $\Sigma \vdash q \iff t_\Sigma(q) = 1$.
($\Rightarrow$) Suppose $\Sigma \vdash p$. Then $t_\Sigma(p) = 1$: Otherwise, $t_\Sigma(q) = 1$, so $\Sigma \vdash q$ by the inductive assumption; since $q \to (p \to \bot)$ is a propositional axiom, we can apply MP twice to get $\Sigma \vdash \bot$, which contradicts the consistency of Σ.
($\Leftarrow$) Suppose $t_\Sigma(p) = 1$. Then $t_\Sigma(q) = 0$, so $\Sigma \nvdash q$, and thus $\Sigma \vdash p$ by completeness of Σ.

Case 2: $p = q \vee r$, $\Sigma \vdash q \iff t_\Sigma(q) = 1$, and $\Sigma \vdash r \iff t_\Sigma(r) = 1$.
($\Rightarrow$) Suppose that $\Sigma \vdash p$. Then $t_\Sigma(p) = 1$: Otherwise, $t_\Sigma(p) = 0$, so $t_\Sigma(q) = 0$ and $t_\Sigma(r) = 0$, hence $\Sigma \nvdash q$ and $\Sigma \nvdash r$, and thus $\Sigma \vdash \neg q$ and $\Sigma \vdash \neg r$ by completeness of Σ; since $\neg q \to (\neg r \to \neg p)$ is a propositional axiom, we can apply MP twice to get $\Sigma \vdash \neg p$, which in view of the propositional axiom $p \to (\neg p \to \bot)$ and MP yields $\Sigma \vdash \bot$, which contradicts the consistency of Σ.
($\Leftarrow$) Suppose $t_\Sigma(p) = 1$. Then $t_\Sigma(q) = 1$ or $t_\Sigma(r) = 1$. Hence $\Sigma \vdash q$ or $\Sigma \vdash r$. Using MP and the propositional axioms $q \to p$ and $r \to p$ we obtain $\Sigma \vdash p$.

Case 3: $p = q \wedge r$, $\Sigma \vdash q \iff t_\Sigma(q) = 1$, and $\Sigma \vdash r \iff t_\Sigma(r) = 1$.
We leave this case as an exercise. ☐

We can now finish the proof of Completeness (second form):
Suppose Σ is consistent. Then by Lindenbaum's Lemma Σ is a subset of a complete set Σ' of propositions. By the previous lemma, such a Σ' has a model, and such a model is also a model of Σ.

The converse—if Σ has a model, then Σ is consistent—is left to the reader.

Application to coloring infinite graphs. What follows is a standard use of compactness of propositional logic, one of many. Let (V, E) be a graph, by which we mean here that V is a set (of vertices) and E (the set of edges) is a binary relation on V that is irreflexive and symmetric, that is, for all $v, w \in V$ we have $(v, v) \notin E$, and if $(v, w) \in E$, then $(w, v) \in E$. Let some $n \geq 1$ be given. Then an *n-coloring of* (V, E) is a function $c : V \to \{1, \dots, n\}$ such that $c(v) \neq c(w)$ for all $(v, w) \in E$: neighboring vertices should have different colors.

Suppose for every finite $V_0 \subseteq V$ there is an n-coloring of (V_0, E_0), where $E_0 := E \cap (V_0 \times V_0)$. We claim that there exists an n-coloring of (V, E).

Proof. Take $A := V \times \{1, \dots, n\}$ as the set of atoms, and think of an atom (v, i) as representing the statement that v has color i. Thus for (V, E) to have an n-coloring means that the following set $\Sigma \subseteq \operatorname{Prop} A$ has a model:

$$\Sigma := \{(v,1) \vee \cdots \vee (v,n) : \ v \in V\} \cup \{\neg\big((v,i) \wedge (v,j)\big) : \ v \in V, 1 \leq i < j \leq n\} \\ \cup \{\neg\big((v,i) \wedge (w,i)\big) : \ (v,w) \in E, 1 \leq i \leq n\}.$$

The assumption that all finite subgraphs of (V, E) are n-colorable yields that every finite subset of Σ has a model. Hence by compactness Σ has a model.

Exercises.

(1) Suppose $\Sigma \subseteq \text{Prop}(A)$ is such that for each truth assignment $t : A \to \{0, 1\}$ there is $p \in \Sigma$ with $t(p) = 1$. Then there are $p_1, \dots, p_n \in \Sigma$ such that $p_1 \vee \cdots \vee p_n$ is a tautology. (The interesting case is when A is infinite.)

2.3 Languages and Structures

Propositional Logic captures only one aspect of mathematical reasoning. We also need the capability to deal with *predicates* (also called *relations*), *variables*, and the *quantifiers* "for all" and "there exists." We now begin setting up a framework for Predicate Logic (or First-Order Logic), which has these additional features and has a claim on being a complete logic for mathematical reasoning.

Definition. A *language*[1] L is a disjoint union of:

(i) a set L^r of *relation symbols*; each $R \in L^r$ has associated arity $a(R) \in \mathbf{N}$;

(ii) a set L^f of *function symbols*; each $F \in L^f$ has associated arity $a(F) \in \mathbf{N}$.

An m-ary relation or function symbol is one that has arity m. Instead of "0-ary", "1-ary", "2-ary" we say "nullary", "unary", "binary". A *constant symbol* is a function symbol of arity 0.

Examples.

(1) The language $L_{\text{Gr}} = \{1, {}^{-1}, \cdot\}$ of groups has constant symbol 1, unary function symbol ${}^{-1}$, and binary function symbol $\cdot$.

(2) The language $L_{\text{Ab}} = \{0, -, +\}$ of (additive) abelian groups has constant symbol 0, unary function symbol $-$, and binary function symbol $+$.

(3) The language $L_{\text{O}} = \{<\}$ has just one binary relation symbol $<$.

(4) The language $L_{\text{OAb}} = \{<, 0, -, +\}$ of ordered abelian groups.

(5) The language $L_{\text{Rig}} = \{0, 1, +, \cdot\}$ of rigs (or semirings) has constant symbols 0 and 1, and binary function symbols $+$ and $\cdot$.

(6) The language $L_{\text{Ri}} = \{0, 1, -, +, \cdot\}$ of rings. The symbols are those of the previous example, plus the unary function symbol $-$.

From now on, let L denote a language.

Definition. A *structure $\mathcal{A}$ for L* (or *L-structure*) is a triple

$$\left(A;\ (R^{\mathcal{A}})_{R \in L^r}, (F^{\mathcal{A}})_{F \in L^f}\right)$$

consisting of:

(i) a nonempty set A, the *underlying set of $\mathcal{A}$*;[2]

(ii) for each m-ary $R \in L^r$ a set $R^{\mathcal{A}} \subseteq A^m$ (an m-ary relation on A), the *interpretation of R in $\mathcal{A}$*;

[1] What we call here a *language* is also known as a *signature*, or a *vocabulary*.

[2] It is also called the *universe* of $\mathcal{A}$; we prefer less grandiose terminology.

(iii) for each n-ary $F \in L^f$ an operation $F^{\mathcal{A}} : A^n \longrightarrow A$ (an n-ary operation on A), the *interpretation of F in $\mathcal{A}$*.

Remark. The interpretation of a constant symbol c of L is a function

$$c^{\mathcal{A}} : A^0 \longrightarrow A.$$

Since A^0 has just one element, $c^{\mathcal{A}}$ is uniquely determined by its value at this element; we shall identify $c^{\mathcal{A}}$ with this value, so $c^{\mathcal{A}} \in A$.

Given an L-structure $\mathcal{A}$, the relations $R^{\mathcal{A}}$ on A (for $R \in L^r$), and operations $F^{\mathcal{A}}$ on A (for $F \in L^f$) are called the *primitives* of $\mathcal{A}$. When $\mathcal{A}$ is clear from context we often omit the superscript $\mathcal{A}$ in denoting the interpretation of a symbol of L in $\mathcal{A}$. The reader is supposed to keep in mind the distinction between symbols of L and their interpretation in an L-structure, even if we use the same notation for both.

Examples.

(1) Each group is considered as an L_{Gr}-structure by interpreting the symbols 1, $^{-1}$, and $\cdot$ as the identity element of the group, its group inverse, and its group multiplication, respectively.

(2) Let $\mathcal{A} = (A;\ 0, -, +)$ be an abelian group; here $0 \in A$ is the zero element of the group, and $- : A \to A$ and $+ : A^2 \to A$ denote the group operations of $\mathcal{A}$. We consider $\mathcal{A}$ as an L_{Ab}-structure by taking as interpretations of the symbols $0, -$ and $+$ of L_{Ab} the group operations 0, $-$ and $+$ on A. (We took here the liberty of using the same notation for possibly entirely different things: $+$ is an element of the set L_{Ab}, but also denotes in this context its interpretation as a binary operation on the set A. Similarly with 0 and $-$.) In fact, *any* set A in which we single out an element, a unary operation on A, and a binary operation on A, can be construed as an L_{Ab}-structure if we choose to do so.

(3) $(\mathbf{N};\ <)$ is an L_{O}-structure where we interpret $<$ as the usual ordering relation on $\mathbf{N}$. Similarly for $(\mathbf{Z};\ <)$, $(\mathbf{Q};\ <)$ and $(\mathbf{R};\ <)$. (Here we take even more notational liberties, by letting $<$ denote five different things: a symbol of L_{O}, and the usual orderings of $\mathbf{N}$, $\mathbf{Z}$, $\mathbf{Q}$, and $\mathbf{R}$ respectively.) Again, any nonempty set A equipped with a binary relation on it can be viewed as an L_{O}-structure.

(4) $(\mathbf{Z};\ <, 0, -, +)$ and $(\mathbf{Q}, <, 0, -, +)$ are both L_{OAb}-structures.

(5) $(\mathbf{N};\ 0, 1, +, \cdot)$ is an L_{Rig}-structure.

(6) $(\mathbf{Z};\ 0, 1, -, +, \cdot)$ is an L_{Ri}-structure.

Let $\mathcal{B}$ be an L-structure with underlying set B, and let A be a nonempty subset of B such that $F^{\mathcal{B}}(A^n) \subseteq A$ for every n-ary function symbol F of L. Then A is the underlying set of an L-structure $\mathcal{A}$ defined by letting

$$F^{\mathcal{A}} := F^{\mathcal{B}} |_{A^n} : A^n \to A, \qquad \text{for } n\text{-ary } F \in L^f,$$
$$R^{\mathcal{A}} := R^{\mathcal{B}} \cap A^m \qquad \text{for } m\text{-ary } R \in L^r.$$

Definition. Such an L-structure $\mathcal{A}$ is said to be a *substructure* of $\mathcal{B}$, notation: $\mathcal{A} \subseteq \mathcal{B}$. We also say in this case that $\mathcal{B}$ is an *extension* of $\mathcal{A}$, or *extends* $\mathcal{A}$.

Examples.
(1) $(\mathbf{Z};\ 0, 1, -, +, \cdot) \subseteq (\mathbf{Q};\ 0, 1, -, +, \cdot) \subseteq (\mathbf{R};\ 0, 1, -, +, \cdot)$
(2) $(\mathbf{N};\ <, 0, 1, +, \cdot) \subseteq (\mathbf{Z};\ <, 0, 1, +, \cdot)$

Definition. Let $\mathcal{A} = (A; \dots)$ and $\mathcal{B} = (B; \dots)$ be L-structures.
A *homomorphism* $h : \mathcal{A} \to \mathcal{B}$ is a map $h : A \to B$ such that
(i) for each m-ary $R \in L^r$ and each $(a_1, \dots, a_m) \in A^m$ we have

$$(a_1, \dots, a_m) \in R^{\mathcal{A}} \Longrightarrow (ha_1, \dots, ha_m) \in R^{\mathcal{B}};$$

(ii) for each n-ary $F \in L^f$ and each $(a_1, \dots, a_n) \in A^n$ we have

$$h(F^{\mathcal{A}}(a_1, \dots, a_n)) = F^{\mathcal{B}}(ha_1, \dots, ha_n).$$

Replacing $\Longrightarrow$ in (i) by $\Longleftrightarrow$ yields the notion of a *strong homomorphism*. An *embedding* is an injective strong homomorphism; an *isomorphism* is a bijective strong homomorphism. An *automorphism* of $\mathcal{A}$ is an isomorphism $\mathcal{A} \to \mathcal{A}$.

If $\mathcal{A} \subseteq \mathcal{B}$, then the inclusion $a \mapsto a : A \to B$ is an embedding $\mathcal{A} \to \mathcal{B}$. Conversely, a homomorphism $h : \mathcal{A} \to \mathcal{B}$ yields a substructure $h(\mathcal{A})$ of $\mathcal{B}$ with underlying set $h(A)$, and if h is an embedding we have an isomorphism $a \mapsto h(a) :\ \mathcal{A} \to h(\mathcal{A})$.

If $i : \mathcal{A} \to \mathcal{B}$ and $j : \mathcal{B} \to \mathcal{C}$ are homomorphisms (strong homomorphisms, embeddings, isomorphisms, respectively), then so is $j \circ i : \mathcal{A} \to \mathcal{C}$. The identity map 1_A on A is an automorphism of $\mathcal{A}$. If $i : \mathcal{A} \to \mathcal{B}$ is an isomorphism then so is the map $i^{-1} : \mathcal{B} \to \mathcal{A}$. Thus the automorphisms of $\mathcal{A}$ form a group $\mathrm{Aut}(\mathcal{A})$ under composition with identity 1_A.

Examples.

1. Let $\mathcal{A} = (\mathbf{Z};\ 0, -, +)$. Then $k \mapsto -k$ is an automorphism of $\mathcal{A}$.
2. Let $\mathcal{A} = (\mathbf{Z};\ <)$. The map $k \mapsto k + 1$ is an automorphism of $\mathcal{A}$ with inverse given by $k \longmapsto k - 1$.

If $\mathcal{A}$ and $\mathcal{B}$ are groups (viewed as structures for the language L_{Gr}), then a homomorphism $h : \mathcal{A} \to \mathcal{B}$ is exactly what in algebra is called a homomorphism from the group $\mathcal{A}$ to the group $\mathcal{B}$. Likewise with rings, and other kinds of algebraic structures.

A *congruence* on the L-structure $\mathcal{A}$ is an equivalence relation $\sim$ on its underlying set A such that
(i) if $R \in L^r$ is m-ary and $a_1 \sim b_1, \dots, a_m \sim b_m$, then

$$(a_1, \dots, a_m) \in R^{\mathcal{A}} \iff (b_1, \dots, b_m) \in R^{\mathcal{A}};$$

(ii) if $F \in L^f$ is n-ary and $a_1 \sim b_1, \dots, a_n \sim b_n$, then

$$F^{\mathcal{A}}(a_1, \dots, a_n) \sim F^{\mathcal{A}}(b_1, \dots, b_n).$$

Note that a strong homomorphism $h : \mathcal{A} \to \mathcal{B}$ yields a congruence $\sim_h$ on $\mathcal{A}$ as follows: for $a_1, a_2 \in A$ we put

$$a_1 \sim_h a_2 \iff h(a_1) = h(a_2).$$

Given a congruence $\sim$ on the L-structure $\mathcal{A}$ we obtain an L-structure $\mathcal{A}/\!\sim$ (the *quotient of* $\mathcal{A}$ *by* $\sim$) as follows:

(i) the underlying set of $\mathcal{A}/\!\sim$ is the quotient set $A/\!\sim$;

(ii) the interpretation of an m-ary $R \in L^r$ in $\mathcal{A}/\!\sim$ is the m-ary relation

$$\{(a_1^\sim, \dots, a_m^\sim) : (a_1, \dots, a_m) \in R^{\mathcal{A}}\}$$

on $A/\!\sim$;

(iii) the interpretation of an n-ary $F \in L^f$ in $\mathcal{A}/\sim$ is the n-ary operation

$$(a_1^\sim, \dots, a_n^\sim) \mapsto F^{\mathcal{A}}(a_1, \dots, a_n)^\sim$$

on $A/\!\sim$.

Note that then we have a strong homomorphism $a \mapsto a^\sim : \mathcal{A} \to \mathcal{A}/\!\sim$.

Products. Let $(\mathcal{B}_i)_{i \in I}$ be a family of L-structures, $\mathcal{B}_i = (B_i; \dots)$ for $i \in I$. The product

$$\prod_{i \in I} \mathcal{B}_i$$

is defined to be the L-structure $\mathcal{B}$ whose underlying set is the product set $\prod_{i \in I} B_i$, and where the basic relations and functions are defined coordinatewise: for m-ary $R \in L^r$ and elements $b_1 = (b_{1i}), \dots, b_m = (b_{mi}) \in \prod_{i \in I} B_i$,

$$(b_1, \dots, b_m) \in R^{\mathcal{B}} \iff (b_{1i}, \dots, b_{mi}) \in R^{\mathcal{B}_i} \text{ for all } i \in I,$$

and for n-ary $F \in L^f$ and $b_1 = (b_{1i}), \dots, b_n = (b_{ni}) \in \prod_{i \in I} B_i$,

$$F^{\mathcal{B}}(b_1, \dots, b_n) := \big(F^{\mathcal{B}_i}(b_{1i}, \dots, b_{ni})\big)_{i \in I}.$$

For $j \in I$ the projection map to the jth factor is the homomorphism

$$\prod_{i \in I} \mathcal{B}_i \to \mathcal{B}_j, \quad (b_i) \mapsto b_j.$$

This product construction makes it possible to combine several homomorphisms with a common domain into a single one: if for each $i \in I$ we have a homomorphism $h_i : \mathcal{A} \to \mathcal{B}_i$ we obtain a homomorphism

$$h = (h_i) : \mathcal{A} \to \prod_{i \in I} \mathcal{B}_i, \quad h(a) := \big(h_i(a_i)\big).$$

2.4 Variables and Terms

Throughout this course

$$\mathrm{Var} = \{\mathsf{v}_0, \mathsf{v}_1, \mathsf{v}_2, \dots\}$$

is a countably infinite set of symbols whose elements will be called *variables*; we assume that $\mathsf{v}_m \neq \mathsf{v}_n$ for $m \neq n$, and that no variable is a function or relation symbol in any language. We let x, y, z (sometimes with subscripts or superscripts) denote variables, unless indicated otherwise.

Remark. Chapters 2–4 go through if we take as our set Var of variables any infinite (possibly uncountable) set; in model theory this can even be convenient. For this more general Var we still insist that no variable is a function or relation symbol in any language. In the few cases in chapters 2–4 that this more general set-up requires changes in proofs, this will be pointed out.

The results in Chapter 5 on undecidability presuppose a numbering of the variables; our $\mathrm{Var} = \{\mathsf{v}_0, \mathsf{v}_1, \mathsf{v}_2, \dots\}$ comes equipped with such a numbering.

Definition. An L-*term* is a word on the alphabet $L^f \cup \mathrm{Var}$ obtained as follows:
(i) each variable (viewed as a word of length 1) is an L-term;
(ii) whenever $F \in L^f$ is n-ary and $t_1, \dots, t_n$ are L-terms, then the concatenation $Ft_1 \dots t_n$ is an L-term.

Note: constant symbols of L are L-terms of length 1, by clause (ii) for $n = 0$. The L-terms are the admissible words on the alphabet $L^f \cup \mathrm{Var}$ where each variable has arity 0. Thus "unique readability" is available.

We often write $t(x_1, \dots, x_n)$ to indicate an L-term t in which no variables other than $x_1, \dots, x_n$ occur. When using this notation we always assume that $x_1, \dots, x_n$ are distinct. Note that we do not require that each of $x_1, \dots, x_n$ actually occurs in $t(x_1, \dots, x_n)$.

If a term is written as an admissible word, then it may be hard to see how it is built up from subterms. In practice we shall therefore use parentheses and brackets in denoting terms, and avoid prefix notation if tradition dictates otherwise.

Example. The word $\cdot + x - yz$ is an L_{Ri}-term. For easier reading we indicate this term instead by $(x + (-y)) \cdot z$ or even $(x - y)z$.

Definition. Let $\mathcal{A}$ be an L-structure and $t = t(\vec{x})$ be an L-term where $\vec{x} = (x_1, \dots, x_m)$. Then we associate to the ordered pair $(t, \vec{x})$ a function $t^{\mathcal{A}} : A^m \to A$ as follows
(i) If t is the variable x_i then $t^{\mathcal{A}}(a) = a_i$ for $a = (a_1, \dots, a_m) \in A^m$.
(ii) If $t = Ft_1 \dots t_n$ where $F \in L^f$ is n-ary and $t_1, \dots, t_n$ are L-terms, then $t^{\mathcal{A}}(a) = F^{\mathcal{A}}(t_1^{\mathcal{A}}(a), \dots, t_n^{\mathcal{A}}(a))$ for $a \in A^m$.
This inductive definition is justified by unique readability. Note that if $\mathcal{B}$ is a second L-structure and $\mathcal{A} \subseteq \mathcal{B}$, then $t^{\mathcal{A}}(a) = t^{\mathcal{B}}(a)$ for t as above and $a \in A^m$.

Example. Consider $\mathbf{R}$ as a ring in the usual way, and let $t(x, y, z)$ be the L_{Ri}-term $(x-y)z$. Then the function $t^{\mathbf{R}} : \mathbf{R}^3 \to \mathbf{R}$ is given by $t^{\mathbf{R}}(a, b, c) = (a-b)c$.

A term is said to be *variable-free* if no variables occur in it. Let t be a variable-free L-term and $\mathcal{A}$ an L-structure. Then the above gives a nullary function $t^{\mathcal{A}} : A^0 \to A$, identified as usual with its value at the unique element of A^0, so $t^{\mathcal{A}} \in A$. In other words, if t is a constant symbol c, then $t^{\mathcal{A}} = c^{\mathcal{A}} \in A$, where $c^{\mathcal{A}}$ is as in the previous section, and if $t = Ft_1 \dots t_n$ with n-ary $F \in L^f$ and variable-free L-terms $t_1, \dots, t_n$, then $t^{\mathcal{A}} = F^{\mathcal{A}}(t_1^{\mathcal{A}}, \dots, t_n^{\mathcal{A}})$.

Generators. Let $\mathcal{B}$ be an L-structure and let $(a_i)_{i \in I}$ be a family of elements of B, and assume also that L has a constant symbol or that $I \neq \emptyset$. Then the set of all elements of the form $t^{\mathcal{B}}(a_{i_1}, \dots, a_{i_m})$ where $t(x_1, \dots, x_m)$ is an L-term and $i_1, \dots, i_m \in I$ is the underlying set of some $\mathcal{A} \subseteq \mathcal{B}$, and this $\mathcal{A}$ is clearly a substructure of any $\mathcal{B}$ that has all a_i in its underlying set. We call this $\mathcal{A}$ the *substructure of* $\mathcal{B}$ *generated by* (a_i); if $\mathcal{A} = \mathcal{B}$, then we say that $\mathcal{B}$ is *generated by* (a_i).

Exercises.

(1) For every L_{Ab}-term $t(x_1, \dots, x_n)$ there are integers $k_1, \dots, k_n$ such that for every abelian group $\mathcal{A} = (A;\ 0, -, +)$,

$$t^{\mathcal{A}}(a_1, \dots, a_n) = k_1 a_1 + \dots + k_n a_n, \quad \text{for all } (a_1, \dots, a_n) \in A^n.$$

Conversely, for any integers $k_1, \dots, k_n$ there is an L_{Ab}-term $t(x_1, \dots, x_n)$ such that in every abelian group $\mathcal{A} = (A;\ 0, -, +)$ the above displayed identity holds.

(2) For every L_{Ri}-term $t(x_1, \dots, x_n)$ there is a polynomial

$$P(x_1, \dots, x_n) \in \mathbf{Z}[x_1, \dots, x_n]$$

such that for every commutative ring $\mathcal{R} = (R;\ 0, 1, -, +, \cdot)$,

$$t^{\mathcal{R}}(r_1, \dots, r_n) = P(r_1, \dots, r_n), \quad \text{for all } (r_1, \dots, r_n) \in R^n.$$

Conversely, for any polynomial $P(x_1, \dots, x_n) \in \mathbf{Z}[x_1, \dots, x_n]$ there is an L_{Ri}-term $t(x_1, \dots, x_n)$ such that in every commutative ring $\mathcal{R} = (R;\ 0, 1, -, +, \cdot)$ the above displayed identity holds.

(3) Let $\mathcal{A}$ and $\mathcal{B}$ be L-structures, $h : \mathcal{A} \to \mathcal{B}$ a homomorphism, and $t = t(x_1, \dots, x_n)$ an L-term. Then

$$h\big(t^{\mathcal{A}}(a_1, \dots, a_n)\big) = t^{\mathcal{B}}(ha_1, \dots, ha_n), \quad \text{for all } (a_1, \dots, a_n) \in A^n.$$

(If $\mathcal{A} \subseteq \mathcal{B}$ and $h : \mathcal{A} \to \mathcal{B}$ is the inclusion, this gives $t^{\mathcal{A}}(a_1, \dots, a_n) = t^{\mathcal{B}}(a_1, \dots, a_n)$ for all $(a_1, \dots, a_n) \in A^n$.)

(4) Consider the L-structure $\mathcal{A} = (\mathbf{N};\ 0, 1, +, \cdot)$ where $L = L_{\mathrm{Rig}}$.
(a) Is there an L-term $t(x)$ such that $t^{\mathcal{A}}(0) = 1$ and $t^{\mathcal{A}}(1) = 0$?
(b) Is there an L-term $t(x)$ such that $t^{\mathcal{A}}(n) = 2^n$ for all $n \in \mathbf{N}$?
(c) Find all the substructures of $\mathcal{A}$.

2.5 Formulas and Sentences

Besides variables we also introduce the eight distinct *logical symbols*

$$\top \quad \bot \quad \neg \quad \vee \quad \wedge \quad = \quad \exists \quad \forall$$

The first five of these we already met when discussing propositional logic. None of these eight symbols is a variable, or a function or relation symbol of any language. Below L denotes a language. To distinguish the logical symbols from those in L, the latter are often referred to as the *non-logical symbols*.

Definition. The *atomic L-formulas* are the following words on the alphabet $L \cup \mathrm{Var} \cup \{\top, \bot, =\}$:
(i) $\top$ and $\bot$,
(ii) $Rt_1 \dots t_m$, where $R \in L^r$ is m-ary and $t_1, \dots, t_m$ are L-terms,
(iii) $= t_1 t_2$, where t_1 and t_2 are L-terms.

The *L-formulas* are the words on the larger alphabet

$$L \cup \mathrm{Var} \cup \{\top, \bot, \neg, \vee, \wedge, =, \exists, \forall\}$$

obtained as follows:
(i) every atomic L-formula is an L-formula;
(ii) if φ, ψ are L-formulas, then so are $\neg\varphi$, $\vee\varphi\psi$ and $\wedge\varphi\psi$;
(iii) if φ is a L-formula and x is a variable, then $\exists x\varphi$ and $\forall x\varphi$ are L-formulas.

Note that all L-formulas are admissible words on the alphabet

$$L \cup \mathrm{Var} \cup \{\top, \bot, \neg, \vee, \wedge, =, \exists, \forall\},$$

where $=$, $\exists$ and $\forall$ are given arity 2 and the other symbols have the arities assigned to them earlier. This fact makes the results on unique readability applicable to L-formulas. (However, not all admissible words on this alphabet are L-formulas: the word $\exists xx$ is admissible but not an L-formula.)

The notational conventions introduced in the section on propositional logic go through, with the role of propositions there taken over by formulas here. (For example, given L-formulas φ and ψ we shall write $\varphi \vee \psi$ to indicate $\vee\varphi\psi$, and $\varphi \to \psi$ to indicate $\neg\varphi \vee \psi$.)

The reader should distinguish between different ways of using the symbol $=$. Sometimes it denotes one of the eight formal logical symbols, but we also use it to indicate equality of mathematical objects in the way we have done already many times. The context should always make it clear what our intention is in this respect without having to spell it out. To increase readability we usually write an atomic formula $= t_1 t_2$ as $t_1 = t_2$ and its negation $\neg = t_1 t_2$ as $t_1 \neq t_2$, where t_1, t_2 are L-terms. The logical symbol $=$ is treated just as a binary relation symbol, but its interpretation in a structure will always be the equality relation on its underlying set. This will become clear later.

Definition. Let φ be a formula of L. Written as a word on the alphabet above we have $\varphi = s_1 \dots s_m$. A *subformula of* φ is a subword of the form $s_i \dots s_k$ where $1 \le i \le k \le m$ which also happens to be a formula of L.

An occurrence of a variable x in φ at the j-th place (that is, $s_j = x$) is said to be a *bound occurrence* if φ has a subformula $s_i s_{i+1} \dots s_k$ with $i \le j \le k$ that is of the form $\exists x \psi$ or $\forall x \psi$. If an occurrence is not bound then it is said to be a *free occurrence*.

At this point the reader is invited to do the first exercise at the end of this section, which gives another useful characterization of subformulas.

Example. In the formula $\big(\exists x(x = y)\big) \wedge x = 0$, where x and y are distinct, the first two occurrences of x are bound, the third is free, and the only occurrence of y is free. (Note: the formula is actually the string $\wedge \exists x = xy = x0$, and the occurrences of x and y are really the occurrences in this string.)

Definition. A *sentence* is a formula in which all occurrences of variables are bound occurrences.

We write $\varphi(x_1, \dots, x_n)$ to indicate a formula φ such that all variables that occur free in φ are among $x_1, \dots, x_n$. In using this notation it is understood that $x_1, \dots, x_n$ are distinct variables, but it is not required that each of $x_1, \dots, x_n$ occurs free in φ. (This is like indicating a polynomial in the indeterminates $x_1, \dots, x_n$ by $p(x_1, \dots, x_n)$, where one allows that some of these indeterminates do not actually occur in p.)

Definition. Let φ be an L-formula, let $x_1, \dots, x_n$ be distinct variables, and let $t_1, \dots, t_n$ be L-terms. Then $\varphi(t_1/x_1, \dots, t_n/x_n)$ is the word obtained by replacing all the free occurences of x_i in φ by t_i, *simultaneously for* $i = 1, \dots, n$. If φ is given in the form $\varphi(x_1, \dots, x_n)$, then we write $\varphi(t_1, \dots, t_n)$ as a shorthand for $\varphi(t_1/x_1, \dots, t_n/x_n)$.

We have the following lemma whose routine proof is left to the reader.

Lemma 2.5.1. *Suppose φ is an L-formula, $x_1, \dots, x_n$ are distinct variables, and $t_1, \dots, t_n$ are L-terms. Then $\varphi(t_1/x_1, \dots, t_n/x_n)$ is an L-formula. If $t_1, \dots, t_n$ are variable-free and $\varphi = \varphi(x_1, \dots, x_n)$, then $\varphi(t_1, \dots, t_n)$ is an L-sentence.*

In the definition of $\varphi(t_1/x_1, \dots, t_n/x_n)$ the "replacing" should be *simultaneous*, because it can happen that $\varphi(t_1/x_1)(t_2/x_2) \ne \varphi(t_1/x_1, t_2/x_2)$.

Let $\mathcal{A}$ be an L-structure with underlying set A, and let $C \subseteq A$. We extend L to a language L_C by adding a constant symbol $\underline{c}$ for each $c \in C$, called the *name* of c. These names are symbols not in L. We make $\mathcal{A}$ into an L_C-structure by keeping the same underlying set and interpretations of symbols of L, and by interpreting each name $\underline{c}$ as the element $c \in C$. The L_C-structure thus obtained is indicated by $\mathcal{A}_C$. Hence for each variable-free L_C-term t we have a corresponding element $t^{\mathcal{A}_C}$ of A, which for simplicity of notation we denote instead by $t^{\mathcal{A}}$. All this applies in particular to the case $C = A$, where in L_A we have a name $\underline{a}$ for each $a \in A$.

Definition. We can now define what it means for an L_A-sentence σ to be *true in the L-structure* $\mathcal{A}$ (notation: $\mathcal{A} \models \sigma$, also read as $\mathcal{A}$ *satisfies* σ or σ *holds in* $\mathcal{A}$, or σ *is valid in* $\mathcal{A}$). First we consider atomic L_A-sentences:

(i) $\mathcal{A} \models \top$, and $\mathcal{A} \not\models \bot$;
(ii) $\mathcal{A} \models Rt_1 \dots t_m$ if and only if $(t_1^{\mathcal{A}}, \dots, t_m^{\mathcal{A}}) \in R^{\mathcal{A}}$, for m-ary $R \in L^r$, and variable free L_A-terms $t_1, \dots, t_m$;
(iii) $\mathcal{A} \models \ t_1 = t_2$ if and only if $t_1^{\mathcal{A}} = t_2^{\mathcal{A}}$, for variable free L_A-terms t_1, t_2.

We extend the definition inductively to arbitrary L_A-sentences as follows:

(i) Suppose $\sigma = \neg\sigma_1$. Then $\mathcal{A} \models \sigma$ if and only if $\mathcal{A} \not\models \sigma_1$.
(ii) Suppose $\sigma = \sigma_1 \vee \sigma_2$. Then $\mathcal{A} \models \sigma$ if and only if $\mathcal{A} \models \sigma_1$ or $\mathcal{A} \models \sigma_2$.
(iii) Suppose $\sigma = \sigma_1 \wedge \sigma_2$. Then $\mathcal{A} \models \sigma$ if and only if $\mathcal{A} \models \sigma_1$ and $\mathcal{A} \models \sigma_2$.
(iv) Suppose $\sigma = \exists x \varphi(x)$. Then $\mathcal{A} \models \sigma$ if and only if $\mathcal{A} \models \varphi(\underline{a})$ for some $a \in A$.
(v) Suppose $\sigma = \forall x \varphi(x)$. Then $\mathcal{A} \models \sigma$ if and only if $\mathcal{A} \models \varphi(\underline{a})$ for all $a \in A$.

Even if we just want to define $\mathcal{A} \models \sigma$ for L-sentences σ, one can see that if σ has the form $\exists x \varphi(x)$ or $\forall x \varphi(x)$, the inductive definition above forces us to consider L_A-sentences $\varphi(\underline{a})$. This is why we introduced names. We didn't say so explicitly, but "inductive" refers here to *induction with respect to the number of logical symbols in* σ. For example, the fact that $\varphi(\underline{a})$ has fewer logical symbols than $\exists x \varphi(x)$ is crucial for the above to count as a definition.

It is easy to check that for an L_A-sentence $\sigma = \exists x_1 \dots \exists x_n \varphi(x_1, \dots, x_n)$,

$$\mathcal{A} \models \sigma \iff \mathcal{A} \models \varphi(\underline{a}_1, \dots, \underline{a}_n) \text{ for some } (a_1, \dots, a_n) \in A^n,$$

and that for an L_A-sentence $\sigma = \forall x_1 \dots \forall x_n \varphi(x_1, \dots, x_n)$,

$$\mathcal{A} \models \sigma \iff \mathcal{A} \models \varphi(\underline{a}_1, \dots, \underline{a}_n) \text{ for all } (a_1, \dots, a_n) \in A^n.$$

Definition. Given an L_A-formula $\varphi(x_1, \dots, x_n)$ we let $\varphi^{\mathcal{A}}$ be the following subset of A^n:

$$\varphi^{\mathcal{A}} = \{(a_1, \dots, a_n) : \mathcal{A} \models \varphi(\underline{a}_1, \dots, \underline{a}_n)\}$$

The formula $\varphi(x_1, \dots, x_n)$ is said to *define* the set $\varphi^{\mathcal{A}}$ in $\mathcal{A}$. A set $S \subseteq A^n$ is said to be *definable* in $\mathcal{A}$ if $S = \varphi^{\mathcal{A}}$ for some L_A-formula $\varphi(x_1, \dots, x_n)$. If moreover φ can be chosen to be an L-formula then S is said to be *0-definable* in $\mathcal{A}$.

Examples.

(1) The set $\{r \in \mathbf{R} : r < \sqrt{2}\}$ is 0-definable in $(\mathbf{R};\ <, 0, 1, +, -, \cdot)$: it is defined by the formula $(x^2 < 1+1) \vee (x < 0)$. (Here x^2 abbreviates the term $x \cdot x$.)
(2) The set $\{r \in \mathbf{R} : r < \pi\}$ is definable in $(\mathbf{R};\ <, 0, 1, +, -, \cdot)$: it is defined by the formula $x < \underline{\pi}$.

We now single out formulas by certain syntactical conditions. These conditions have semantic counterparts in terms of the behaviour of these formulas under various kinds of homomorphisms, as shown in some exercises below.

An L-formula is said to be *quantifier-free* if it has no occurrences of $\exists$ and no occurrences of $\forall$. An L-formula is said to be *existential* if it has the form

$\exists x_1 \dots \exists x_m \varphi$ with distinct $x_1, \dots, x_m$ and a quantifier-free L-formula φ. An L-formula is said to be *universal* if it has the form $\forall x_1 \dots \forall x_m \varphi$ with distinct $x_1, \dots, x_m$ and a quantifier-free LL-formula φ. An L-formula is said to be *positive* if it has no occurrences of $\neg$ (but it can have occurrences of $\bot$).

Exercises.

(1) Let φ and ψ be L-formulas; put $\mathrm{sf}(\varphi) :=$ set of subformulas of φ.
  (a) If φ is atomic, then $\mathrm{sf}(\varphi) = \{\varphi\}$.
  (b) $\mathrm{sf}(\neg\varphi) = \{\neg\varphi\} \cup \mathrm{sf}(\varphi)$.
  (c) $\mathrm{sf}(\varphi \vee \psi) = \{\varphi \vee \psi\} \cup \mathrm{sf}(\varphi) \cup \mathrm{sf}(\psi)$, and $\mathrm{sf}(\varphi \wedge \psi) = \{\varphi \wedge \psi\} \cup \mathrm{sf}(\varphi) \cup \mathrm{sf}(\psi)$.
  (d) $\mathrm{sf}(\exists x\varphi) = \{\exists x\varphi\} \cup \mathrm{sf}(\varphi)$, and $\mathrm{sf}(\forall x\varphi) = \{\forall x\varphi\} \cup \mathrm{sf}(\varphi)$.

(2) If $t(x_1, \dots, x_n)$ is an L_A-term and $a_1, \dots, a_n \in A$, then

$$t(\underline{a}_1, \dots, \underline{a}_n)^{\mathcal{A}} = t^{\mathcal{A}}(a_1, \dots, a_n).$$

(3) Suppose that $S_1 \subseteq A^n$ and $S_2 \subseteq A^n$ are defined in $\mathcal{A}$ by the L_A-formulas $\varphi_1(x_1, \dots, x_n)$ and $\varphi_2(x_1, \dots, x_n)$ respectively. Then:
  (a) $S_1 \cup S_2$ is defined in $\mathcal{A}$ by $(\varphi_1 \vee \varphi_2)(x_1, \dots, x_n)$.
  (b) $S_1 \cap S_2$ is defined in $\mathcal{A}$ by $(\varphi_1 \wedge \varphi_2)(x_1, \dots, x_n)$.
  (c) $A^n \smallsetminus S_1$ is defined in $\mathcal{A}$ by $\neg\varphi_1(x_1, \dots, x_n)$.
  (d) $S_1 \subseteq S_2 \Longleftrightarrow \mathcal{A} \models \forall x_1 \dots \forall x_n \big(\varphi_1 \to \varphi_2\big)$.

(4) Let $\pi : A^{m+n} \to A^m$ be the projection map given by

$$\pi(a_1, \dots, a_{m+n}) = (a_1, \dots, a_m),$$

and for $S \subseteq A^{m+n}$ and $a \in A^m$, put

$$S(a) := \{b \in A^n : (a, b) \in S\} \qquad \text{(a \textit{section} of } S).$$

Suppose that $S \subseteq A^{m+n}$ is defined in $\mathcal{A}$ by the L_A-formula $\varphi(x, y)$ where $x = (x_1, \dots, x_m)$ and $y = (y_1, \dots, y_n)$. Then $\exists y_1 \dots \exists y_n \varphi(x, y)$ defines in $\mathcal{A}$ the subset $\pi(S)$ of A^m, and $\forall y_1 \dots \forall y_n \varphi(x, y)$ defines in $\mathcal{A}$ the set

$$\{a \in A^m : S(a) = A^n\}.$$

(5) The following sets are 0-definable in the corresponding structures:
  (a) The ordering relation $\{(m, n) \in \mathbf{N}^2 : m < n\}$ in $(\mathbf{N};\ 0, +)$.
  (b) The set $\{2, 3, 5, 7, \dots\}$ of prime numbers in the semiring $\mathcal{N} = (\mathbf{N};\ 0, 1, +, \cdot)$.
  (c) The set $\{2^n : n \in \mathbf{N}\}$ in the semiring $\mathcal{N}$.
  (d) The set $\{a \in \mathbf{R} : f \text{ is continuous at } a\}$ in $(\mathbf{R}; <, f)$ where $f : \mathbf{R} \to \mathbf{R}$ is any function.

(6) Let $\mathcal{A} \subseteq \mathcal{B}$. Then we consider L_A to be a sublanguage of L_B in such a way that each $a \in A$ has the same name in L_A as in L_B. This convention is in force throughout these notes.
  (a) For each variable free L_A-term t we have $t^{\mathcal{A}} = t^{\mathcal{B}}$.
  (b) If the L_A-sentence σ is quantifier-free, then $\mathcal{A} \models \sigma \Leftrightarrow \mathcal{B} \models \sigma$.
  (c) If σ is an existential L_A-sentence, then $\mathcal{A} \models \sigma \Rightarrow \mathcal{B} \models \sigma$
  (d) If σ is a universal L_A-sentence, then $\mathcal{B} \models \sigma \Rightarrow \mathcal{A} \models \sigma$.

(7) Suppose $h : \mathcal{A} \longrightarrow \mathcal{B}$ is a homomorphism of L-structures. For each L_A-term t, let t_h be the L_B-term obtained from t by replacing each occurrence of a name $\underline{a}$ of an element $a \in A$ by the name $\underline{ha}$ of the corresponding element $ha \in B$. Similarly, for each L_A-formula φ, let φ_h be the L_B-formula obtained from φ by replacing each occurrence of a name $\underline{a}$ of an element $a \in A$ by the name $\underline{ha}$ of the corresponding element $ha \in B$. Note that if φ is a sentence, so is φ_h. Then:
(a) if t is a variable-free L_A-term, then $h(t^{\mathcal{A}}) = t_h^{\mathcal{B}}$;
(b) if σ is an L_A-sentence containing no negation symbol and no $\forall$-symbol, then $\mathcal{A} \models \sigma \Rightarrow \mathcal{B} \models \sigma_h$;
(c) if σ is a positive L_A-sentence and h is surjective, then $\mathcal{A} \models \sigma \Rightarrow \mathcal{B} \models \sigma_h$;
(d) if σ is an L_A-sentence and h is an isomorphism, then $\mathcal{A} \models \sigma \Leftrightarrow \mathcal{B} \models \sigma_h$;

In particular, isomorphic L-structures satisfy exactly the same L-sentences.

2.6 Models

In the rest of this chapter L is a language, $\mathcal{A}$ is an L-structure (with underlying set A), and, unless indicated otherwise, t is an L-term, φ, ψ, and θ are L-formulas, σ is an L-sentence, and Σ is a set of L-sentences. We drop the prefix L in "L-term" and "L-formula" and so on, unless this would cause confusion.

Definition. We say that $\mathcal{A}$ is a *model of* Σ or Σ *holds in* $\mathcal{A}$ (denoted $\mathcal{A} \models \Sigma$) if $\mathcal{A} \models \sigma$ for each $\sigma \in \Sigma$.

To discuss examples it is convenient to introduce some notation. Suppose L contains (at least) the constant symbol 0 and the binary function symbol $+$. Given any terms $t_1, \dots, t_n$ we define the term $t_1 + \cdots + t_n$ inductively as follows: it is the term 0 if $n = 0$, the term t_1 if $n = 1$, and the term $(t_1 + \cdots + t_{n-1}) + t_n$ for $n > 1$. We write nt for the term $t + \cdots + t$ with n summands, in particular, $0t$ and $1t$ denote the terms 0 and t respectively. Suppose L contains the constant symbol 1 and the binary function symbol $\cdot$ (the multiplication sign). Then we have similar notational conventions for $t_1 \cdot \ldots \cdot t_n$ and t^n; in particular, for $n = 0$ both stand for the term 1, and t^1 is just t.

Examples. Fix three distinct variables x, y, z.
(1) *Groups* are the L_{Gr}-structures that are models of

$$\mathrm{Gr} := \{\forall x(x \cdot 1 = x \wedge 1 \cdot x = x), \forall x(x \cdot x^{-1} = 1 \wedge x^{-1} \cdot x = 1), \\ \forall x \forall y \forall z((x \cdot y) \cdot z = x \cdot (y \cdot z))\}$$

(2) *Abelian groups* are the L_{Ab}-structures that are models of

$$\mathrm{Ab} := \{\forall x(x + 0 = x), \forall x(x + (-x) = 0), \forall x \forall y(x + y = y + x), \\ \forall x \forall y \forall z((x + y) + z = x + (y + z))\}$$

(3) *Torsion-free abelian groups* are the L_{Ab}-structures that are models of

$$\mathrm{Ab} \cup \{\forall x(nx = 0 \to x = 0) \, : \, n = 1, 2, 3, \ldots\}$$

(4) *Rings* are the L_{Ri}-structures that are models of

$$\begin{aligned}\mathrm{Ri} := \mathrm{Ab} \cup \{&\forall x \forall y \forall z\,\big((x\cdot y)\cdot z = x\cdot(y\cdot z)\big), \forall x\big(x\cdot 1 = x \wedge 1\cdot x = x\big),\\ &\forall x\forall y\forall z\big((x\cdot(y+z) = x\cdot y + x\cdot z \wedge (x+y)\cdot z = x\cdot z + y\cdot z)\big)\}\end{aligned}$$

(5) *Fields* are the L_{Ri}-structures that are models of

$$\mathrm{Fl} = \mathrm{Ri} \cup \{\forall x\forall y(x\cdot y = y\cdot x), 1 \neq 0, \forall x\big(x\neq 0 \to \exists y\,(x\cdot y = 1)\big)\}$$

(6) *Fields of characteristic* 0 are the L_{Ri}-structures that are models of

$$\mathrm{Fl}(0) := \mathrm{Fl} \cup \{n1 \neq 0 \,:\, n = 2,3,5,7,11,\dots\}$$

(7) *Algebraically closed fields* are the L_{Ri}-structures that are models of

$$\mathrm{ACF} := \mathrm{Fl} \cup \{\forall u_1 \dots \forall u_n \exists x(x^n + u_1 x^{n-1} + \dots + u_n = 0) \,:\, n = 2,3,4,5,\dots\}$$

Here $u_1, u_2, u_3, \dots$ is some fixed infinite sequence of distinct variables, distinct also from x, and $u_i x^{n-i}$ abbreviates $u_i \cdot x^{n-i}$, for $i = 1,\dots,n$.

(8) *Algebraically closed fields of characteristic* 0 are the L_{Ri}-structures that are models of $\mathrm{ACF}(0) := \mathrm{ACF} \cup \{n1 \neq 0 \,:\, n = 2,3,5,7,11,\dots\}$.

Definition. We say that σ *is a logical consequence of* Σ (written $\Sigma \models \sigma$) if σ is true in every model of Σ.

Example. It is well-known that in any ring R we have $x \cdot 0 = 0$ for all $x \in R$. This can now be expressed as $\mathrm{Ri} \models \forall x(\,x\cdot 0 = 0)$.

We defined what it means for a *sentence* σ to hold in a given structure $\mathcal{A}$. We now extend this to arbitrary *formulas*.

First define an $\mathcal{A}$*-instance of a formula* $\varphi = \varphi(x_1,\dots,x_m)$ to be an L_A-sentence of the form $\varphi(\underline{a}_1,\dots,\underline{a}_m)$ with $a_1,\dots,a_m \in A$. Of course φ can also be written as $\varphi(y_1,\dots,y_n)$ for another sequence of variables $y_1,\dots,y_n$, for example, $y_1,\dots,y_n$ could be obtained by permuting $x_1,\dots,x_m$, or it could be $x_1,\dots,x_m,x_{m+1}$, obtained by adding a variable x_{m+1}. Thus for the above to count as a definition of "$\mathcal{A}$-instance," the reader should check that these different ways of specifying variables (including at least the variables occurring free in φ) give the same $\mathcal{A}$-instances.

Definition. A formula φ is said to be *valid in* $\mathcal{A}$ (notation: $\mathcal{A} \models \varphi$) if all its $\mathcal{A}$-instances are true in $\mathcal{A}$.

The reader should check that if $\varphi = \varphi(x_1,\dots,x_m)$, then

$$\mathcal{A} \models \varphi \iff \mathcal{A} \models \forall x_1 \dots \forall x_m \varphi.$$

We also extend the notion of "logical consequence of Σ" to formulas.

Definition. We say that φ *is a logical consequence of* Σ (notation: $\Sigma \models \varphi$) if $\mathcal{A} \models \varphi$ for all models $\mathcal{A}$ of Σ.

One should not confuse the notion of "logical consequence of Σ" with that of "provable from Σ." We shall give a definition of *provable from* Σ in the next section. The two notions will turn out to be equivalent, but that is hardly obvious from their definitions: we shall need much of the next chapter to prove this equivalence, which is called the Completeness Theorem for Predicate Logic. We finish this section with two basic facts:

Lemma 2.6.1. *Let $\alpha(x_1,\dots,x_m)$ be an L_A-term, and recall that α defines a map $\alpha^{\mathcal{A}}: A^m \to A$. Let $t_1,\dots,t_m$ be variable-free L_A-terms, with $t_i^{\mathcal{A}} = a_i \in A$ for $i = 1,\dots,m$. Then $\alpha(t_1,\dots,t_m)$ is a variable-free L_A-term, and*

$$\alpha(t_1,\dots,t_m)^{\mathcal{A}} \;=\; \alpha(\underline{a}_1,\dots \underline{a}_m)^{\mathcal{A}} \;=\; \alpha^{\mathcal{A}}(t_1^{\mathcal{A}},\dots,t_m^{\mathcal{A}}).$$

This follows by a straightforward induction on α.

Lemma 2.6.2. *Let $t_1,\dots,t_m$ be variable-free L_A-terms with $t_i^{\mathcal{A}} = a_i \in A$ for $i = 1,\dots,m$. Let $\varphi(x_1,\dots,x_m)$ be an L_A-formula. Then the L_A-formula $\varphi(t_1,\dots,t_m)$ is a sentence and*

$$\mathcal{A} \models \varphi(t_1,\dots,t_m) \iff \mathcal{A} \models \varphi(\underline{a}_1,\dots,\underline{a}_m).$$

Proof. To keep notations simple we give the proof only for $m = 1$ with $t = t_1$ and $x = x_1$. We proceed by induction on the number of logical symbols in $\varphi(x)$.

Suppose that φ is atomic. The case where φ is $\top$ or $\bot$ is obvious. Assume φ is $R\alpha_1 \dots \alpha_m$ where $R \in L^r$ is m-ary and $\alpha_1(x),\dots,\alpha_m(x)$ are L_A-terms. Then $\varphi(t) = R\alpha_1(t)\dots\alpha_m(t)$ and $\varphi(\underline{a}) = R\alpha_1(\underline{a})\dots\alpha_m(\underline{a})$. We have $\mathcal{A} \models \varphi(t)$ iff $(\alpha_1(t)^{\mathcal{A}},\dots,\alpha_m(t)^{\mathcal{A}}) \in R^{\mathcal{A}}$ and also $\mathcal{A} \models \varphi(\underline{a})$ iff $(\alpha_1(\underline{a})^{\mathcal{A}},\dots,\alpha_m(\underline{a})^{\mathcal{A}}) \in R^{\mathcal{A}}$. As $\alpha_i(t)^{\mathcal{A}} = \alpha_i(\underline{a})^{\mathcal{A}}$ for all i by the previous lemma, we have $\mathcal{A} \models \varphi(t)$ iff $\mathcal{A} \models \varphi(\underline{a})$. The case that $\varphi(x)$ is $\alpha(x) = \beta(x)$ is handled the same way.

It is also clear that the desired property is inherited by disjunctions, conjunctions and negations of formulas $\varphi(x)$ that have the property. Suppose now that $\varphi(x) = \exists y\, \psi$.

Case $y \neq x$: Then $\psi = \psi(x,y)$, $\varphi(t) = \exists y\psi(t,y)$ and $\varphi(\underline{a}) = \exists y\psi(\underline{a},y)$. As $\varphi(t) = \exists y\psi(t,y)$, we have $\mathcal{A} \models \varphi(t)$ iff $\mathcal{A} \models \psi(t,\underline{b})$ for some $b \in A$. By the inductive hypothesis the latter is equivalent to $\mathcal{A} \models \psi(\underline{a},\underline{b})$ for some $b \in A$, hence equivalent to $\mathcal{A} \models \exists y\psi(\underline{a},y)$. As $\varphi(\underline{a}) = \exists y\psi(\underline{a},y)$, we conclude that $\mathcal{A} \models \varphi(t)$ iff $\mathcal{A} \models \varphi(\underline{a})$.

Case $y = x$: Then x does not occur free in $\varphi(x) = \exists x\psi$. So $\varphi(t) = \varphi(\underline{a}) = \varphi$ is an L_A-sentence, and $\mathcal{A} \models \varphi(t) \Leftrightarrow \mathcal{A} \models \varphi(\underline{a})$ is obvious.

When $\varphi(x) = \forall y\, \psi$ then one can proceed exactly as above by distinguishing two cases. $\square$

2.7 Logical Axioms and Rules; Formal Proofs

In this section we introduce a *proof system* for predicate logic and state its completeness. We then derive as a consequence the compactness theorem and some of its corollaries. The completeness is proved in the next chapter.

A *propositional axiom of* L is by definition a formula that for some φ, ψ, θ occurs in the list below:

1. $\top$
2. $\varphi \to (\varphi \vee \psi)$; $\qquad \varphi \to (\psi \vee \varphi)$
3. $\neg\varphi \to \big(\neg\psi \to \neg(\varphi \vee \psi)\big)$
4. $(\varphi \wedge \psi) \to \varphi$; $\qquad (\varphi \wedge \psi) \to \psi$
5. $\varphi \to \big(\psi \to (\varphi \wedge \psi)\big)$
6. $\big(\varphi \to (\psi \to \theta)\big) \to \big((\varphi \to \psi) \to (\varphi \to \theta)\big)$
7. $\varphi \to (\neg\varphi \to \bot)$
8. $(\neg\varphi \to \bot) \to \varphi$

Each of items 2–8 is a scheme describing infinitely many axioms. Note that this list is the same as the list in Section 2.2 except that instead of propositions p, q, r we have formulas φ, ψ, θ.

The *logical axioms* of L are the propositional axioms of L and the equality and quantifier axioms of L as defined below.

Definition. The *equality axioms* of L are the following formulas:
(i) $x = x$,
(ii) $x = y \to y = x$,
(iii) $(x = y \wedge y = z) \to x = z$,
(iv) $(x_1 = y_1 \wedge \ldots \wedge x_m = y_m \wedge Rx_1 \ldots x_m) \to Ry_1 \ldots y_m$,
(v) $(x_1 = y_1 \wedge \ldots \wedge x_n = y_n) \to Fx_1 \ldots x_n = Fy_1 \ldots y_n$,
with the following restrictions on the variables and symbols of L: x, y, z are distinct in (ii) and (iii); in (iv), $x_1, \ldots, x_m$, $y_1, \ldots, y_m$ are distinct and $R \in L^r$ is m-ary; in (v), $x_1, \ldots, x_n$, $y_1, \ldots, y_n$ are distinct, and $F \in L^f$ is n-ary. Note that (i) represents an axiom scheme rather than a single axiom, since different variables x give different formulas $x = x$. Likewise with (ii)–(v).

Let x and y be distinct variables, and let $\varphi(y)$ be the formula $\exists x(x \neq y)$. Then $\varphi(y)$ is valid in all $\mathcal{A}$ with $|A| > 1$, but $\varphi(x/y)$ is invalid in all $\mathcal{A}$. Thus substituting x for the free occurrences of y does not always preserve validity. To get rid of this anomaly, we introduce the following restriction on substitutions of a term t for free occurrences of y.

Definition. We say that t *is free for* y *in* φ, if no variable in t can become bound upon replacing the free occurrences of y in φ by t, more precisely: whenever x is a variable in t, then there are no occurrences of subformulas in φ of the form $\exists x\psi$ or $\forall x\psi$ that contain an occurrence of y that is free in φ.

Note that if t is variable-free, then t is free for y in φ. We remark that "free for" abbreviates "free to be substituted for." In exercise 2 the reader is asked to show that, with this restriction, substitution of a term for the free occurrences of a variable does preserve validity.

Definition. The *quantifier axioms of L* are the formulas $\varphi(t/y) \to \exists y\varphi$ and $\forall y\varphi \to \varphi(t/y)$ where t is free for y in φ.

These axioms have been chosen to have the following property.

Proposition 2.7.1. *The logical axioms of L are valid in every L-structure.*

We first prove this for the propositional axioms of L. Let $\alpha_1, \dots, \alpha_n$ be distinct propositional atoms not in L. Let $p = p(\alpha_1, \dots, \alpha_n) \in \text{Prop}\{\alpha_1, \dots, \alpha_n\}$. Let $\varphi_1, \dots, \varphi_n$ be formulas and let $p(\varphi_1, \dots, \varphi_n)$ be the word obtained by replacing each occurrence of α_i in p by φ_i for $i = 1, \dots, n$. One checks easily that $p(\varphi_1, \dots, \varphi_n)$ is a formula.

Lemma 2.7.2. *Suppose $\varphi_i = \varphi_i(x_1, \dots, x_m)$ for $1 \le i \le n$ and let $a_1, \dots, a_m \in A$. Define a truth assignment $t : \{\alpha_1, \dots, \alpha_n\} \longrightarrow \{0, 1\}$ by $t(\alpha_i) = 1$ iff $\mathcal{A} \models \varphi_i(\underline{a}_1, \dots, \underline{a}_m)$. Then $p(\varphi_1, \dots, \varphi_n)$ is an L-formula and*

$$\begin{aligned} p(\varphi_1, \dots, \varphi_n)(\underline{a}_1/x_1, \dots, \underline{a}_m/x_m) &= p(\varphi_1(\underline{a}_1, \dots, \underline{a}_m), \dots, \varphi_n(\underline{a}_1, \dots, \underline{a}_m)), \\ t(p(\alpha_1, \dots, \alpha_n)) = 1 \iff \mathcal{A} &\models p(\varphi_1(\underline{a}_1, \dots, \underline{a}_m), \dots, \varphi_n(\underline{a}_1, \dots, \underline{a}_m)). \end{aligned}$$

In particular, if p is a tautology, then $\mathcal{A} \models p(\varphi_1, \dots, \varphi_n)$.

Proof. Easy induction on p. We leave the details to the reader. □

Definition. An *L-tautology* is a formula of the form $p(\varphi_1, \dots, \varphi_n)$ for some tautology $p(\alpha_1, \dots, \alpha_n) \in \text{Prop}\{\alpha_1, \dots, \alpha_n\}$ and some formulas $\varphi_1, \dots, \varphi_n$.

By Lemma 2.7.2 all L-tautologies are valid in all L-structures. The propositional axioms of L are L-tautologies, so all propositional axioms of L are valid in all L-structures. It is easy to check that all equality axioms of L are valid in all L-structures. In exercise 3 below the reader is asked to show that all quantifier axioms of L are valid in all L-structures. This finishes the proof of Proposition 2.7.1.

Next we introduce rules for deriving new formulas from given formulas.

Definition. The *logical rules of L* are the following:
(i) Modus Ponens (MP): From φ and $\varphi \to \psi$, infer ψ.
(ii) Generalization Rule (G): If the variable x does not occur free in φ, then
  (a) from $\varphi \to \psi$, infer $\varphi \to \forall x\psi$;
  (b) from $\psi \to \varphi$, infer $\exists x\psi \to \varphi$.

A key property of the logical rules is that if the hypotheses of a logical rule are valid in the L-structure $\mathcal{A}$, then so is its conclusion. Ihe reader should verify this.

Definition. A *formal proof*, or just *proof*, of φ from Σ is a sequence $\varphi_1, \dots, \varphi_n$ of formulas with $n \ge 1$ and $\varphi_n = \varphi$, such that for $k = 1, \dots, n$:
(i) either $\varphi_k \in \Sigma$,
(ii) or φ_k is a logical axiom,

(iii) or there are $i, j \in \{1, \ldots, k-1\}$ such that φ_k can be inferred from φ_i and φ_j by MP, or from φ_i by G.

If there exists a proof of φ from Σ, then we write $\Sigma \vdash \varphi$ and say Σ *proves* φ.

Proposition 2.7.3. *If $\Sigma \vdash \varphi$, then $\Sigma \models \varphi$.*

This follows easily from earlier facts that we stated and which the reader was asked to verify. The converse is more interesting, and due to Gödel (1930):

Theorem 2.7.4 (Completeness Theorem of Predicate Logic).

$$\Sigma \vdash \varphi \iff \Sigma \models \varphi$$

Remark. Our choice of proof system, and thus our notion of formal proof is somewhat arbitrary. However the equivalence of $\vdash$ and $\models$ (Completeness Theorem) justifies our choice of logical axioms and rules and shows in particular that no further logical axioms and rules are needed. Moreover, this equivalence has consequences that can be stated in terms of $\models$ alone. An example is the important Compactness Theorem.

Theorem 2.7.5 (Compactness Theorem). *If $\Sigma \models \sigma$ then there is a finite subset Σ_0 of Σ such that $\Sigma_0 \models \sigma$.*

The Compactness Theorem has many consequences. Here is one.

Corollary 2.7.6. *Suppose σ is an L_{Ri}-sentence that holds in all fields of characteristic 0. Then there exists a natural number N such that σ is true in all fields of characteristic $p > N$.*

Proof. By assumption we have $\mathrm{Fl} \cup \{n1 \neq 0 : n = 1, 2, 3, \ldots\} \models \sigma$. Then by Compactness, there is $N \in \mathbf{N}$ such that $\mathrm{Fl} \cup \{n1 \neq 0 : n = 1, \ldots, N\} \models \sigma$. It follows that σ is true in all fields of characteristic $p > N$. ☐

The converse of this proposition fails, see exercise 8 below. Note that Fl(0) is infinite. Could there be an alternative *finite* set of axioms whose models are exactly the fields of characteristic 0?

Corollary 2.7.7. *There is no finite set of L_{Ri}-sentences whose models are exactly the fields of characteristic 0.*

Proof. Suppose there is such a finite set of sentences $\{\sigma_1, \ldots, \sigma_N\}$. Let $\sigma := \sigma_1 \wedge \cdots \wedge \sigma_N$. Then the models of σ are just the fields of characteristic 0. By the previous result σ holds in some field of characteristic $p > 0$. Contradiction! ☐

Exercises. All but the last one to be done without using Theorem 2.7.4 or 2.7.5. Actually, Exercise (4) will be used in proving Theorem 2.7.4.

(1) Let $L = \{R\}$ where R is a binary relation symbol, and let $\mathcal{A} = (A; R)$ be a finite L-structure (i. e. the set A is finite). Then there exists an L-sentence σ such that the models of σ are exactly the L-structures isomorphic to $\mathcal{A}$. (In fact, for an arbitrary language L, two finite L-structures are isomorphic iff they satisfy the same L-sentences.)

(2) If t is free for y in φ and φ is valid in $\mathcal{A}$, then $\varphi(t/y)$ is valid in $\mathcal{A}$.

(3) Suppose t is free for y in $\varphi = \varphi(x_1, \dots, x_n, y)$. Then:

(i) Each $\mathcal{A}$-instance of the quantifier axiom $\varphi(t/y) \to \exists y\varphi$ has the form

$$\varphi(\underline{a}_1, \dots, \underline{a}_n, \tau) \to \exists y\varphi(\underline{a}_1, \dots, \underline{a}_n, y)$$

with $a_1, \dots, a_n \in A$ and τ a variable-free L_A-term.

(ii) The quantifier axiom $\varphi(t/y) \to \exists y\varphi$ is valid in $\mathcal{A}$. (Hint: use Lemma 2.6.2.)

(iii) The quantifier axiom $\forall y\varphi \to \varphi(t/y)$ is valid in $\mathcal{A}$.

(4) If φ is an L-tautology, then $\vdash \varphi$.

(5) $\Sigma \vdash \varphi_i$ for $i = 1, \dots, n \Longleftrightarrow \Sigma \vdash \varphi_1 \wedge \cdots \wedge \varphi_n$.

(6) If $\Sigma \vdash \varphi \to \psi$ and $\Sigma \vdash \psi \to \varphi$, then $\Sigma \vdash \varphi \leftrightarrow \psi$.

(7) $\vdash \neg\exists x\varphi \leftrightarrow \forall x\neg\varphi$ and $\vdash \neg\forall x\varphi \leftrightarrow \exists x\neg\varphi$.

(8) Indicate an L_{Ri}-sentence that is true in the field of real numbers, but false in all fields of positive characteristic.

(9) Let σ be an L_{Ab}-sentence which holds in all non-trivial torsion free abelian groups. Then there exists $N \in \mathbf{N}$ such that σ is true in all groups $\mathbf{Z}/p\mathbf{Z}$ where p is a prime number and $p > N$.

Chapter 3

The Completeness Theorem

The main aim of this chapter is to prove the Completeness Theorem. As a byproduct we also derive some more elementary facts about predicate logic. The last section contains some of the basics of *universal algebra*, which we can treat here rather efficiently using our construction of a so-called *term-model* in the proof of the Completeness Theorem.

Conventions on the use of L, $\mathcal{A}$, t, φ, ψ, θ, σ and Σ are as in the beginning of Section 2.6.

3.1 Another Form of Completeness

It is convenient to prove first a variant of the Completeness Theorem.

Definition. We say that Σ is *consistent* if $\Sigma \nvdash \bot$, and otherwise (that is, if $\Sigma \vdash \bot$), we call Σ *inconsistent*.

Theorem 3.1.1 (Completeness Theorem - second form).
Σ is consistent if and only if Σ has a model.

We first show that this second form of the Completeness Theorem implies the first form. This will be done through a series of technical lemmas, which are also useful later in this Chapter.

Lemma 3.1.2. *Suppose $\Sigma \vdash \varphi$. Then $\Sigma \vdash \forall x\varphi$.*

Proof. From $\Sigma \vdash \varphi$ and the L-tautology $\varphi \to (\neg\forall x\varphi \to \varphi)$ we obtain $\Sigma \vdash \neg\forall x\varphi \to \varphi$ by MP. Then by G we have $\Sigma \vdash \neg\forall x\varphi \to \forall x\varphi$. Using the L-tautology $(\neg\forall x\varphi \to \forall x\varphi) \to \forall x\varphi$ and MP we get $\Sigma \vdash \forall x\varphi$. $\square$

Lemma 3.1.3 (Deduction Lemma). *Suppose $\Sigma \cup \{\sigma\} \vdash \varphi$. Then $\Sigma \vdash \sigma \to \varphi$.*

Proof. By induction on the length of a proof of φ from $\Sigma \cup \{\sigma\}$.

The cases where φ is a logical axiom, or $\varphi \in \Sigma \cup \{\sigma\}$ or φ is obtained by MP are treated just as in the proof of the Deduction Lemma of Propositional Logic.

Suppose that φ is obtained by part (a) of G, so φ is $\varphi_1 \to \forall x\psi$ where x does not occur free in φ_1 and $\Sigma \cup \{\sigma\} \vdash \varphi_1 \to \psi$, and where we assume inductively that $\Sigma \vdash \sigma \to (\varphi_1 \to \psi)$. We have to argue that then $\Sigma \vdash \sigma \to (\varphi_1 \to \forall x\psi)$. From the L-tautology $\big(\sigma \to (\varphi_1 \to \psi)\big) \to \big((\sigma \wedge \varphi_1) \to \psi\big)$ and MP we get $\Sigma \vdash (\sigma \wedge \varphi_1) \to \psi$. Since x does not occur free in $\sigma \wedge \varphi_1$ this gives $\Sigma \vdash (\sigma \wedge \varphi_1) \to \forall x\psi$, by G. Using the L-tautology

$$\big((\sigma \wedge \varphi_1) \to \forall x\psi\big) \to \big(\sigma \to (\varphi_1 \to \forall x\psi)\big)$$

and MP this gives $\Sigma \vdash \sigma \to (\varphi_1 \to \forall x\psi)$.

The case that φ is obtained by part (b) of G is left to the reader. □

Corollary 3.1.4. *Suppose* $\Sigma \cup \{\sigma_1, \ldots, \sigma_n\} \vdash \varphi$. *Then* $\Sigma \vdash \sigma_1 \wedge \ldots \wedge \sigma_n \to \varphi$.

We leave the proof as an exercise.

Corollary 3.1.5. $\Sigma \vdash \sigma$ *if and only if* $\Sigma \cup \{\neg\sigma\}$ *is inconsistent.*

The proof is just like that of the corresponding fact of Propositional Logic.

Lemma 3.1.6. $\Sigma \vdash \forall y\varphi$ *if and only if* $\Sigma \vdash \varphi$.

Proof. ($\Leftarrow$) This is Lemma 3.1.2. For ($\Rightarrow$), assume $\Sigma \vdash \forall y\varphi$. We have the quantifier axiom $\forall y\varphi \to \varphi$, so by MP we get $\Sigma \vdash \varphi$. □

Corollary 3.1.7. $\Sigma \vdash \forall y_1 \ldots \forall y_n\varphi$ *if and only if* $\Sigma \vdash \varphi$.

Corollary 3.1.8. *The second form of the Completeness Theorem implies the first form (Theorem 2.7.4).*

Proof. Assume the second form of the Completeness Theorem holds, and that $\Sigma \models \varphi$. It suffices to show that then $\Sigma \vdash \varphi$. From $\Sigma \models \varphi$ we obtain $\Sigma \models \forall y_1 \ldots \forall y_n\varphi$ where $\varphi = \varphi(y_1, \ldots, y_n)$, and so $\Sigma \cup \{\neg\sigma\}$ has no model where σ is the sentence $\forall y_1 \ldots \forall y_n\varphi$. But then by the 2^{nd} form of the Completeness Theorem $\Sigma \cup \{\neg\sigma\}$ is inconsistent. Then by Corollary 3.1.5 we have $\Sigma \vdash \sigma$ and thus by Corollary 3.1.7 we get $\Sigma \vdash \varphi$. □

We finish this section with another form of the Compactness Theorem:

Theorem 3.1.9 (Compactness Theorem - second form).
If each finite subset of Σ *has a model, then* Σ *has a model.*

This follows from the second form of the Completeness Theorem.

3.2 Proof of the Completeness Theorem

We are now going to prove Theorem 3.1.1. Since ($\Leftarrow$) is clear, we focus our attention on ($\Rightarrow$), that is, given a consistent set of sentences Σ we must show that Σ has a model. This job will be done in a series of lemmas. Unless we say so, we do not assume in those lemmas that Σ is consistent.

Lemma 3.2.1. *Suppose $\Sigma \vdash \varphi$ and t is free for x in φ. Then $\Sigma \vdash \varphi(t/x)$.*

Proof. From $\Sigma \vdash \varphi$ we get $\Sigma \vdash \forall x\varphi$ by Lemma 3.1.2. Then MP together with the quantifier axiom $\forall x\varphi \to \varphi(t/x)$ gives $\Sigma \vdash \varphi(t/x)$ as required. □

Lemma 3.2.2. *Suppose $\Sigma \vdash \varphi$, and $t_1, \ldots, t_n$ are terms whose variables do not occur bound in φ. Then $\Sigma \vdash \varphi(t_1/x_1, \ldots, t_n/x_n)$.*

Proof. Take distinct variables $y_1, \ldots, y_n$ that do not occur in φ or $t_1, \ldots, t_n$ and that are distinct from $x_1, \ldots, x_n$. Use Lemma 3.2.1 n times in succession to obtain $\Sigma \vdash \psi$ where $\psi = \varphi(y_1/x_1, \ldots, y_n/x_n)$. Apply Lemma 3.2.1 again n times to get $\Sigma \vdash \psi(t_1/y_1, \ldots, t_n/y_n)$. To finish, observe that $\psi(t_1/y_1, \ldots, t_n/y_n) = \varphi(t_1/x_1, \ldots, t_n/x_n)$. □

Lemma 3.2.3.
(1) *For each L-term t we have $\vdash t = t$.*
(2) *Let t, t' be L-terms and $\Sigma \vdash t = t'$. Then $\Sigma \vdash t' = t$.*
(3) *Let t_1, t_2, t_3 be L-terms and $\Sigma \vdash t_1 = t_2$ and $\Sigma \vdash t_2 = t_3$. Then $\Sigma \vdash t_1 = t_3$.*
(4) *Let $R \in L^r$ be m-ary and let $t_1, t'_1, \ldots, t_m, t'_m$ be L-terms such that $\Sigma \vdash t_i = t'_i$ for $i = 1, \ldots, m$ and $\Sigma \vdash Rt_1 \ldots t_m$. Then $\Sigma \vdash Rt'_1 \ldots t'_m$.*
(5) *Let $F \in L^f$ be n-ary, and let $t_1, t'_1, \ldots, t_n, t'_n$ be L-terms such that $\Sigma \vdash t_i = t'_i$ for $i = 1, \ldots, n$. Then $\Sigma \vdash Ft_1 \ldots t_n = Ft'_1 \ldots t'_n$.*

Proof. For (1), use the equality axiom $x = x$ and apply Lemma 3.2.2. For (2), take an equality axiom $x = y \to y = x$ and apply Lemma 3.2.2 to get $\vdash t = t' \to t' = t$. Then MP gives $\Sigma \vdash t' = t$.
For (3), take an equality axiom $(x = y \wedge y = z) \to (x = z)$ and apply Lemma 3.2.2 to get $\vdash (t_1 = t_2 \wedge t_2 = t_3) \to t_1 = t_3$. By the assumptions and Exercise 5 in Section **??** we have $\Sigma \vdash t_1 = t_2 \wedge t_2 = t_3$. Then MP yields $\Sigma \vdash t_1 = t_3$.
To prove (4) we first use Exercise 5 to get

$$\Sigma \vdash t_1 = t'_1 \wedge \ldots \wedge t_m = t'_m \wedge Rt_1 \ldots t_m.$$

Take an equality axiom $x_1 = y_1 \wedge \ldots \wedge x_m = y_m \wedge Rx_1 \ldots x_m \to Ry_1 \ldots y_m$ and apply Lemma 3.2.2 to obtain

$$\Sigma \vdash t_1 = t'_1 \wedge \ldots \wedge t_m = t'_m \wedge Rt_1 \ldots t_m \to Rt'_1 \ldots t'_m.$$

Then MP gives $\Sigma \vdash Rt'_1 \ldots t'_m$. Part (5) is obtained in a similar way by using an equality axiom $x_1 = y_1 \wedge \ldots \wedge x_n = y_n \to Fx_1 \ldots x_n = Fy_1 \ldots y_n$. □

Definition. Let T_L be the set of variable free L-terms. We define a binary relation $\sim_\Sigma$ on T_L by

$$t_1 \sim_\Sigma t_2 \iff \Sigma \vdash t_1 = t_2.$$

Parts (1), (2) and (3) of the last lemma yield the following.

Lemma 3.2.4. *The relation $\sim_\Sigma$ is an equivalence relation on T_L.*

Definition. Suppose L has at least one constant symbol. Then T_L is nonempty. We define the L-structure $\mathcal{A}_\Sigma$ as follows:
(i) Its underlying set is $A_\Sigma := T_L/\sim_\Sigma$. Let $[t]$ denote the equivalence class of $t \in T_L$ with respect to $\sim_\Sigma$.
(ii) If $R \in L^r$ is m-ary, then $R^{\mathcal{A}_\Sigma} \subseteq A_\Sigma^m$ is given by

$$([t_1], \dots, [t_m]) \in R^{\mathcal{A}_\Sigma} \ :\Longleftrightarrow \Sigma \vdash Rt_1 \dots t_m \qquad (t_1, \dots, t_m \in T_L).$$

(iii) If $F \in L^f$ is n-ary, then $F^{\mathcal{A}_\Sigma} : A_\Sigma^n \to A_\Sigma$ is given by

$$F^{\mathcal{A}_\Sigma}([t_1], \dots, [t_n]) = [Ft_1 \dots t_n] \qquad (t_1, \dots, t_n \in T_L).$$

Remark. The reader should verify that this counts as a definition, i.e., does not introduce ambiguity. (Use parts (4) and (5) of Lemma 3.2.3.)

Corollary 3.2.5. *Suppose L has a constant symbol, and Σ is consistent. Then*
(1) *for each $t \in T_L$ we have $t^{\mathcal{A}_\Sigma} = [t]$;*
(2) *for each atomic σ we have: $\Sigma \vdash \sigma \Longleftrightarrow \mathcal{A}_\Sigma \models \sigma$.*

Proof. Part (1) follows by an easy induction. Let σ be $Rt_1 \dots t_m$ where $R \in L^r$ is m-ary and $t_1, \dots, t_m \in T_L$. Then

$$\Sigma \vdash Rt_1 \dots t_m \Leftrightarrow ([t_1], \dots, [t_m]) \in R^{\mathcal{A}_\Sigma} \Leftrightarrow \mathcal{A}_\Sigma \models Rt_1 \dots t_m,$$

where the last "$\Leftrightarrow$" follows from the definition of $\models$ together with part (1). Now suppose that σ is $t_1 = t_2$ where $t_1, t_2 \in T_L$. Then

$$\Sigma \vdash t_1 = t_2 \Leftrightarrow [t_1] = [t_2] \Leftrightarrow t_1^{\mathcal{A}_\Sigma} = t_2^{\mathcal{A}_\Sigma} \Leftrightarrow \mathcal{A}_\Sigma \models t_1 = t_2.$$

We also have $\Sigma \vdash \top \Leftrightarrow \mathcal{A}_\Sigma \models \top$. So far we haven't used the assumption that Σ is consistent, but now we do. The consistency of Σ means that $\Sigma \nvdash \bot$. We also have $\mathcal{A}_\Sigma \not\models \bot$ by definition of $\models$. Thus $\Sigma \vdash \bot \Leftrightarrow \mathcal{A}_\Sigma \models \bot$. ☐

If the equivalence in part (2) of this corollary holds for *all* σ (not only for atomic σ), then $\mathcal{A}_\Sigma \models \Sigma$, so we would have found a model of Σ, and be done. But clearly this equivalence can only hold for all σ if Σ has the property that for each σ, either $\Sigma \vdash \sigma$ or $\Sigma \vdash \neg\sigma$. This property is of interest for other reasons as well, and deserves a name:

Definition. We say that Σ is *complete* if Σ is consistent, and for each σ either $\Sigma \vdash \sigma$ or $\Sigma \vdash \neg\sigma$.

Example. Let $L = L_{\text{Ab}}$, $\Sigma :=$ Ab (the set of axioms for abelian groups), and σ the sentence $\exists x(x \neq 0)$. Then $\Sigma \nvdash \sigma$ since the trivial group doesn't satisfy σ. Also $\Sigma \nvdash \neg\sigma$, since there are non-trivial abelian groups and σ holds in such groups. Thus Σ is *not* complete.

Completeness is a strong property and it can be hard to show that a given set of axioms is complete. The set of axioms for algebraically closed fields of characteristic 0 *is* complete (see the end of Section 4.3).

A key fact about completeness needed in this chapter is that any consistent set of sentences extends to a complete set of sentences:

Lemma 3.2.6 (Lindenbaum). *Suppose Σ is consistent. Then $\Sigma \subseteq \Sigma'$ for some complete set of L-sentences Σ'.*

The proof uses Zorn's Lemma, and is just like that of the corresponding fact of Propositional Logic in Section 1.2.

Completeness of Σ does not guarantee that the equivalence of part (2) of Corollary 3.2.5 holds for *all* σ. Completeness is only a necessary condition for this equivalence to hold for all σ; another necessary condition is "to have witnesses":

Definition. A Σ-*witness* for the sentence $\exists x\varphi(x)$ is a term $t \in T_L$ such that $\Sigma \vdash \varphi(t)$. We say that Σ *has witnesses* if there is a Σ-witness for every sentence $\exists x\varphi(x)$ proved by Σ.

Theorem 3.2.7. *Let L have a constant symbol, and suppose Σ is consistent. Then the following two conditions are equivalent:*
(i) *For each σ we have: $\Sigma \vdash \sigma \Leftrightarrow \mathcal{A}_\Sigma \models \sigma$.*
(ii) *Σ is complete and has witnesses.*
In particular, if Σ is complete and has witnesses, then $\mathcal{A}_\Sigma$ is a model of Σ.

Proof. It should be clear that (i) implies (ii). For the converse, assume (ii). We use induction on the number of logical symbols in σ to obtain (i). We already know that (i) holds for atomic sentences. The cases that $\sigma = \neg\sigma_1$, $\sigma = \sigma_1 \vee \sigma_2$, and $\sigma = \sigma_1 \wedge \sigma_2$ are treated just as in the proof of the corresponding Lemma 2.2.11 for Propositional Logic. It remains to consider two cases:
Case $\sigma = \exists x\varphi(x)$:
($\Rightarrow$) Suppose that $\Sigma \vdash \sigma$. Because we are assuming that Σ has witnesses we have a $t \in T_L$ such that $\Sigma \vdash \varphi(t)$. Then by the inductive hypothesis $\mathcal{A}_\Sigma \models \varphi(t)$. So by Lemma 2.6.2 we have an $a \in A_\Sigma$ such that $\mathcal{A}_\Sigma \models \varphi(\underline{a})$. Therefore $\mathcal{A}_\Sigma \models \exists x\varphi(x)$, hence $\mathcal{A}_\Sigma \models \sigma$.
($\Leftarrow$) Assume $\mathcal{A}_\Sigma \models \sigma$. Then there is an $a \in A_\Sigma$ such that $\mathcal{A}_\Sigma \models \varphi(\underline{a})$. Choose $t \in T_L$ such that $[t] = a$. Then $t^{\mathcal{A}_\Sigma} = a$, hence $\mathcal{A}_\Sigma \models \varphi(t)$ by Lemma 2.6.2. Applying the inductive hypothesis we get $\Sigma \vdash \varphi(t)$. This yields $\Sigma \vdash \exists x\varphi(x)$ by MP and the quantifier axiom $\varphi(t) \to \exists x\varphi(x)$.
Case $\sigma = \forall x\varphi(x)$: This is similar to the previous case but we also need the result from Exercise 7 in Section 2.7 that $\vdash \neg\forall x\varphi \leftrightarrow \exists x\neg\varphi$. $\square$

We call attention to some new notation in the next lemmas: the symbol $\vdash_L$ is used to emphasize that we are dealing with formal provability within L.

Lemma 3.2.8. *Let Σ be a set of L-sentences, c a constant symbol not in L, and $L_c := L \cup \{c\}$. Let $\varphi(y)$ be an L-formula and suppose $\Sigma \vdash_{L_c} \varphi(c)$. Then $\Sigma \vdash_L \varphi(y)$.*

Proof. (Sketch) Take a proof of $\varphi(c)$ from Σ in the language L_c, and take a variable z different from all variables occurring in that proof, and also such that $z \neq y$. Replace in every formula in this proof each occurrence of c by z. Check that one obtains in this way a proof of $\varphi(z/y)$ in the language L from Σ. So $\Sigma \vdash_L \varphi(z/y)$ and hence by Lemma 3.2.1 we have $\Sigma \vdash_L \varphi(z/y)(y/z)$, that is, $\Sigma \vdash_L \varphi(y)$. $\square$

Lemma 3.2.9. *Assume Σ is consistent and $\Sigma \vdash \exists y \varphi(y)$. Let c be a constant symbol not in L. Put $L_c := L \cup \{c\}$. Then $\Sigma \cup \{\varphi(c)\}$ is a consistent set of L_c-sentences.*

Proof. Suppose not. Then $\Sigma \cup \{\varphi(c)\} \vdash_{L_c} \perp$. By the Deduction Lemma (3.1.3) $\Sigma \vdash_{L_c} \varphi(c) \to \perp$. Then by Lemma 3.2.8 we have $\Sigma \vdash_L \varphi(y) \to \perp$. By G we have $\Sigma \vdash_L \exists y \varphi(y) \to \perp$. Applying MP yields $\Sigma \vdash \perp$, contradicting the consistency of Σ. □

Lemma 3.2.10. *Suppose Σ is consistent. Let $\sigma_1 = \exists x_1 \varphi_1(x_1)$, …, $\sigma_n = \exists x_n \varphi_n(x_n)$ be such that $\Sigma \vdash \sigma_i$ for every $i = 1, \ldots, n$. Let $c_1, \ldots, c_n$ be distinct constant symbols not in L. Put $L' := L \cup \{c_1, \ldots, c_n\}$ and $\Sigma' = \Sigma \cup \{\varphi_1(c_1), \ldots, \varphi_n(c_n)\}$. Then Σ' is a consistent set of L'-sentences.*

Proof. The previous lemma covers the case $n = 1$. The general case follows by induction on n. □

In the next lemma we use a superscript "w" for "witness."

Lemma 3.2.11. *Suppose Σ is consistent. For each L-sentence $\sigma = \exists x \varphi(x)$ such that $\Sigma \vdash \sigma$, let c_σ be a constant symbol not in L such that if σ' is a different L-sentence of the form $\exists x' \varphi'(x')$ provable from Σ, then $c_\sigma \neq c_{\sigma'}$. Put*

$$\begin{aligned} L^w &:= L \cup \{c_\sigma : \sigma = \exists x \varphi(x) \text{ is an } L\text{-sentence such that } \Sigma \vdash \sigma\} \\ \Sigma^w &:= \Sigma \cup \{\varphi(c_\sigma) : \sigma = \exists x \varphi(x) \text{ is an } L\text{-sentence such that } \Sigma \vdash \sigma\} \end{aligned}$$

Then Σ^w is a consistent set of L^w-sentences.

Proof. Suppose not. Then $\Sigma^w \vdash \perp$. Take a proof of $\perp$ from Σ^w and let $c_{\sigma_1}, \ldots, c_{\sigma_n}$ be constant symbols in $L^w \smallsetminus L$ such that this proof is a proof of $\perp$ in the language $L \cup \{c_{\sigma_1}, \ldots, c_{\sigma_n}\}$ from $\Sigma \cup \{\varphi_1(c_{\sigma_1}), \ldots, \varphi_n(c_{\sigma_n})\}$, where $\sigma_i = \exists x_i \varphi_i(x_i)$ for $1 \leq i \leq n$. So $\Sigma \cup \{\varphi_1(c_{\sigma_1}), \ldots, \varphi_n(c_{\sigma_n})\}$ is an inconsistent set of $L \cup \{c_{\sigma_1}, \ldots, c_{\sigma_n}\}$-sentences. This contradicts Lemma 3.2.10. □

Lemma 3.2.12. *Let (L_n) be an increasing sequence of languages: $L_0 \subseteq L_1 \subseteq L_2 \subseteq \ldots$, and denote their union by L_∞. Let Σ_n be a consistent set of L_n-sentences, for each n, such that $\Sigma_0 \subseteq \Sigma_1 \subseteq \Sigma_2 \ldots$. Then the union $\Sigma_\infty := \bigcup_n \Sigma_n$ is a consistent set of L_∞-sentences.*

Proof. Suppose that $\Sigma_\infty \vdash \perp$. Take a proof of $\perp$ from Σ_∞. Then we can choose n so large that this is actually a proof of $\perp$ from Σ_n in L_n. This contradicts the consistency of Σ_n. □

Suppose the language L^* extends L, let $\mathcal{A}$ be an L-structure, and let $\mathcal{A}^*$ be an L^*-structure. Then $\mathcal{A}$ is said to be a *reduct* of $\mathcal{A}^*$ (and $\mathcal{A}^*$ an *expansion* of $\mathcal{A}$) if $\mathcal{A}$ and $\mathcal{A}^*$ have the same underlying set and the same interpretations of the symbols of L. For example, $(\mathbf{N};\ 0, +)$ is a reduct of $(\mathbf{N};\ <, 0, 1, +, \cdot)$. Note that any L^*-structure $\mathcal{A}^*$ has a unique reduct to an L-structure, which we indicate by $\mathcal{A}^* |_L$. A key fact (to be verified by the reader) is that if $\mathcal{A}$ is a reduct of $\mathcal{A}^*$, then $t^{\mathcal{A}} = t^{\mathcal{A}^*}$ for all variable-free L_A-terms t, and

$$\mathcal{A} \models \sigma \iff \mathcal{A}^* \models \sigma$$

for all L_A-sentences σ.

We can now prove Theorem 3.1.1.

Proof. Let Σ be a consistent set of L-sentences. We construct a sequence (L_n) of languages and a sequence (Σ_n) where each Σ_n is a consistent set of L_n-sentences. We begin by setting $L_0 = L$ and $\Sigma_0 = \Sigma$. Given the language L_n and the consistent set of L_n-sentences Σ_n, put

$$L_{n+1} := \begin{cases} L_n & \text{if } n \text{ is even,} \\ L_n^w & \text{if } n \text{ is odd,} \end{cases}$$

choose a complete set of L_n-sentences $\Sigma_n' \supseteq \Sigma_n$, and put

$$\Sigma_{n+1} := \begin{cases} \Sigma_n' & \text{if } n \text{ is even,} \\ \Sigma_n^w & \text{if } n \text{ is odd.} \end{cases}$$

Here L_n^w and Σ_n^w are obtained from L_n and Σ_n in the same way that L^w and Σ^w are obtained from L and Σ in Lemma 3.2.11. Note that $L_n \subseteq L_{n+1}$, and $\Sigma_n \subseteq \Sigma_{n+1}$.

By the previous lemma the set Σ_∞ of L_∞-sentences is consistent. It is also complete. To see this, let σ be an L_∞-sentence. Take n even and so large that σ is an L_n-sentence. Then $\Sigma_{n+1} \vdash \sigma$ or $\Sigma_{n+1} \vdash \neg\sigma$ and thus $\Sigma_\infty \vdash \sigma$ or $\Sigma_\infty \vdash \neg\sigma$.

We claim that Σ_∞ has witnesses. To see this, let $\sigma = \exists x \varphi(x)$ be an L_∞-sentence such that $\Sigma_\infty \vdash \sigma$. Now take n to be odd and so large that σ is an L_n-sentence and $\Sigma_n \vdash \sigma$. Then by construction of $\Sigma_{n+1} = \Sigma_n^w$ we have $\Sigma_{n+1} \vdash \varphi(c_\sigma)$, so $\Sigma_\infty \vdash \varphi(c_\sigma)$.

It follows from Theorem 3.2.7 that Σ_∞ has a model, namely $\mathcal{A}_{\Sigma_\infty}$. Put $\mathcal{A} := \mathcal{A}_{\Sigma_\infty} |_L$. Then $\mathcal{A} \models \Sigma$. This concludes the proof of the Completeness Theorem (second form). $\square$

Exercises.

(1) Suppose Σ is consistent. Then Σ is complete if and only if every two models of Σ satisfy the same sentences.

(2) Let L have just a constant symbol c, a unary relation symbol U and a unary function symbol f, and suppose that $\Sigma \vdash Ufc$, and that f does not occur in the sentences of Σ. Then $\Sigma \vdash \forall x Ux$.

3.3 Some Elementary Results of Predicate Logic

Here we obtain some generalities of predicate logic: Equivalence and Equality Theorems, Variants, and Prenex Form. In some proofs we shall take advantage of the fact that the Completeness Theorem is now available.

Lemma 3.3.1 (Distribution Rule).

(i) *Suppose* $\Sigma \vdash \varphi \to \psi$. *Then* $\Sigma \vdash \exists x\varphi \to \exists x\psi$ *and* $\Sigma \vdash \forall x\varphi \to \forall x\psi$.

(ii) *Suppose* $\Sigma \vdash \varphi \leftrightarrow \psi$. *Then* $\Sigma \vdash \exists x\varphi \leftrightarrow \exists x\psi$ *and* $\Sigma \vdash \forall x\varphi \leftrightarrow \forall x\psi$.

Proof. We only do (i), since (ii) then follows easily. Let $\mathcal{A}$ be a model of Σ. By the Completeness Theorem it suffices to show that then $\mathcal{A} \models \exists x\varphi \to \exists x\psi$ and $\mathcal{A} \models \forall x\varphi \to \forall x\psi$. We shall prove $\mathcal{A} \models \exists x\varphi \to \exists x\psi$ and leave the other part to the reader. We have $\mathcal{A} \models \varphi \to \psi$. Choose variables $y_1, \dots, y_n$ such that $\varphi = \varphi(x, y_1, \dots, y_n)$ and $\psi = \psi(x, y_1, \dots, y_n)$. We need only show that then for all $a_1, \dots, a_n \in A$

$$\mathcal{A} \models \exists x\varphi(x, \underline{a}_1, \dots, \underline{a}_n) \to \exists x\psi(x, \underline{a}_1, \dots, \underline{a}_n)$$

Suppose $\mathcal{A} \models \exists x\varphi(x, \underline{a}_1, \dots, \underline{a}_n)$. Then $\mathcal{A} \models \varphi(\underline{a}_0, \underline{a}_1, \dots, \underline{a}_n)$ for some $a_0 \in A$. From $\mathcal{A} \models \varphi \to \psi$ we obtain $\mathcal{A} \models \varphi(\underline{a}_0, \dots, \underline{a}_n) \to \psi(\underline{a}_0, \dots, \underline{a}_n)$, hence $\mathcal{A} \models \psi(\underline{a}_0, \dots, \underline{a}_n)$, and thus $\mathcal{A} \models \exists x\psi(x, \underline{a}_1, \dots, \underline{a}_n)$. □

Theorem 3.3.2 (Equivalence Theorem). *Suppose* $\Sigma \vdash \varphi \leftrightarrow \varphi'$, *and let* ψ' *be obtained from* ψ *by replacing some occurrence of* φ *as a subformula of* ψ *by* φ'. *Then* ψ' *is again a formula and* $\Sigma \vdash \psi \leftrightarrow \psi'$.

Proof. By induction on the number of logical symbols in ψ. If ψ is atomic, then necessarily $\psi = \varphi$ and $\psi' = \varphi'$ and the desired result holds trivially.
Suppose that $\psi = \neg\theta$. Then either $\psi = \varphi$ and $\psi' = \varphi'$, and the desired result holds trivially, or the occurrence of φ we are replacing is an occurrence in θ. Then the inductive hypothesis gives $\Sigma \vdash \theta \leftrightarrow \theta'$, where θ' is obtained by replacing that occurrence (of φ) by φ'. Then $\psi' = \neg\theta'$ and the desired result follows easily. The cases $\psi = \psi_1 \vee \psi_2$ and $\psi = \psi_1 \wedge \psi_2$ are left as exercises.
Suppose that $\psi = \exists x\theta$. The case $\psi = \varphi$ (and thus $\psi' = \varphi'$) is trivial. Suppose $\psi \neq \varphi$. Then the occurrence of φ we are replacing is an occurrence inside θ. So by inductive hypothesis we have $\Sigma \vdash \theta \leftrightarrow \theta'$. Then by the distribution rule $\Sigma \vdash \exists x\theta \leftrightarrow \exists x\theta'$. The proof is similar if $\psi = \forall x\theta$. □

Definition. We say φ_1 and φ_2 are Σ-*equivalent* if $\Sigma \vdash \varphi_1 \leftrightarrow \varphi_2$. (In case $\Sigma = \emptyset$, we just say *equivalent*.) One verifies easily that Σ-equivalence is an equivalence relation on the set of L-formulas.

Given a family $(\varphi)_{i \in I}$ of formulas with finite index set I we choose a bijection $k \mapsto i(k) : \{1, \dots, n\} \to I$ and set

$$\bigvee_{i \in I} \varphi_i := \varphi_{i(1)} \vee \cdots \vee \varphi_{i(n)}, \qquad \bigwedge_{i \in I} \varphi_i := \varphi_{i(1)} \wedge \cdots \wedge \varphi_{i(n)}.$$

If I is clear from context we just write $\bigvee_i \varphi_i$ and $\bigwedge_i \varphi_i$ instead. Of course, these notations $\bigvee_{i \in I} \varphi_i$ and $\bigwedge_{i \in I} \varphi_i$ can only be used when the particular choice of bijection of $\{1, \dots, n\}$ with I does not matter; this is usually the case because the equivalence class of $\varphi_{i(1)} \vee \cdots \vee \varphi_{i(n)}$ does not depend on this choice, and the same is true with "$\wedge$" instead of "$\vee$".

Definition. A *variant* of a formula is obtained by successive replacements of the following type:
(i) replace an occurrence of a subformula $\exists x\varphi$ by $\exists y\varphi(y/x)$.
(ii) replace an occurrence of a subformula $\forall x\varphi$ by $\forall y\varphi(y/x)$.
where y is free for x in φ and y does not occur free in φ.

Lemma 3.3.3. *A formula is equivalent to any of its variants.*

Proof. By the Equivalence Theorem (3.3.2) it suffices to show $\vdash \exists x\varphi \leftrightarrow \exists y\varphi(y/x)$ and $\vdash \forall x\varphi \leftrightarrow \forall y\varphi(y/x)$ where y is free for x in φ and does not occur free in φ. We prove the first equivalence, leaving the second as an exercise. Applying G to the quantifier axiom $\varphi(y/x) \rightarrow \exists x\varphi$ gives $\vdash \exists y\varphi(y/x) \rightarrow \exists x\varphi$. Similarly we get $\vdash \exists x\varphi \rightarrow \exists y\varphi(y/x)$ (use that $\varphi = \varphi(y/x)(x/y)$ by the assumption on y). An application of Exercise 6 finishes the proof. □

Definition. A formula in *prenex form* is a formula $Q_1x_1 \ldots Q_nx_n\varphi$ where $x_1, \ldots, x_n$ are distinct variables, each $Q_i \in \{\exists, \forall\}$ and φ is quantifier free. We call $Q_1x_1 \ldots Q_nx_n$ the *prefix*, and φ the *matrix* of the formula. Note that a quantifier-free formula is in prenex form; this is the case $n = 0$.

We leave the proof of the next lemma as an exercise. Instead of "occurrence of ... as a subformula" we say "part ...". In this lemma Q denotes a quantifier, that is, $Q \in \{\exists, \forall\}$, and Q' denotes the other quantifier: $\exists' = \forall$ and $\forall' = \exists$.

Lemma 3.3.4. *The following* prenex transformations *always change a formula into an equivalent formula.*
(1) *Replace the formula by one of its variants.*
(2) *Replace a part $\neg Qx\psi$ by $Q'x\neg\psi$.*
(3) *Replace a part $(Qx\psi) \vee \theta$ by $Qx(\psi \vee \theta)$ where x is not free in θ.*
(4) *Replace a part $\psi \vee Qx\theta$ by $Qx(\psi \vee \theta)$ where x is not free in ψ.*
(5) *Replace a part $(Qx\psi) \wedge \theta$ by $Qx(\psi \wedge \theta)$ where x is not free in θ.*
(6) *Replace a part $\psi \wedge Qx\theta$ by $Qx(\psi \wedge \theta)$ where x is not free in ψ.*

Remark. Note that the free variables of a formula (those that occur free in the formula) do not change under prenex transformations.

Theorem 3.3.5 (Prenex Form). *Every formula can be changed into one in prenex form by a finite sequence of prenex transformations. In particular, each formula is equivalent to one in prenex form.*

Proof. By induction on the number of logical symbols. Atomic formulas are already in prenex form. To simplify notation, write $\varphi \Longrightarrow_{pr} \psi$ to indicate that ψ can be obtained from φ by a finite sequence of prenex transformations. Assume inductively that

$$\begin{aligned} \varphi_1 &\Longrightarrow_{pr} Q_1x_1 \ldots Q_mx_m\psi_1 \\ \varphi_2 &\Longrightarrow_{pr} Q_{m+1}y_1 \ldots Q_{m+n}y_n\psi_2, \end{aligned}$$

where $Q_1, \ldots, Q_m, \ldots, Q_{m+n} \in \{\exists, \forall\}$, $x_1, \ldots, x_m$ are distinct, $y_1, \ldots, y_n$ are distinct, and ψ_1 and ψ_2 are quantifier-free.

Then for $\varphi := \neg\varphi_1$, we have

$$\varphi \Longrightarrow_{pr} \neg Q_1 x_1 \dots Q_m x_m \psi_1.$$

Applying m prenex transformations of type (2) we get

$$\neg Q_1 x_1 \dots Q_m x_m \psi_1 \Longrightarrow_{pr} {Q_1}' x_1 \dots {Q_m}' x_m \neg\psi_1,$$

hence $\varphi \Longrightarrow_{pr} {Q_1}' x_1 \dots {Q_m}' x_m \neg\psi_1$.

Next, let $\varphi := \varphi_1 \vee \varphi_2$. The assumptions above yield

$$\varphi \Longrightarrow_{pr} (Q_1 x_1 \dots Q_m x_m \psi_1) \vee (Q_{m+1} y_1 \dots Q_{m+n} y_n \psi_2).$$

Replacing here the RHS (righthand side) by a variant we may assume that $\{x_1, \dots, x_m\} \cap \{y_1, \dots, y_n\} = \emptyset$, and also that no x_i occurs free in ψ_2 and no y_j occurs free in ψ_1. Applying $m+n$ times prenex transformation of types (3) and (4) we obtain

$$\begin{aligned}(Q_1 x_1 \dots Q_m x_m \psi_1) \vee (Q_{m+1} y_1 \dots Q_{m+n} y_n \psi_2) \Longrightarrow_{pr}& \\ Q_1 x_1 \dots Q_m x_m Q_{m+1} y_1 \dots Q_{m+n} y_n (\psi_1 \vee \psi_2).&\end{aligned}$$

Hence $\varphi \Longrightarrow_{pr} Q_1 x_1 \dots Q_m x_m Q_{m+1} y_1 \dots Q_{m+n} y_n (\psi_1 \vee \psi_2)$. Likewise, to deal with $\varphi_1 \wedge \varphi_2$, we apply prenex transformations of types (5) and (6).

Next, let $\varphi := \exists x \varphi_1$. Applying prenex transformations of type (1) we can assume $x_1, \dots, x_m$ are different from x. Then $\varphi \Longrightarrow_{pr} \exists x Q_1 x_1 \dots Q_m x_n \psi_1$, and the RHS is in prenex form. The case $\varphi := \forall x \varphi_1$ is similar. $\square$

We finish this section with a result on equalities. Note that by Corollary 2.1.7, the result of replacing an occurrence of an L-term τ in t by an L-term τ' is an L-term t'. An occurrence of an L-term τ in φ is said to be *outside quantifier scope* if

Proposition 3.3.6. *Let τ and τ' be L-terms such that $\Sigma \vdash \tau = \tau'$, let t' be obtained from t by replacing an occurrence of τ in t by τ', and let φ' be obtained from φ by replacing an occurrence of τ in φ outside quantifier scope by τ'. Then $\Sigma \vdash t = t'$ and $\Sigma \vdash \varphi \leftrightarrow \varphi'$.*

Proof. We shall obtain $\Sigma \vdash t = t'$ by induction on t. If t is a variable, then necessarily $t = \tau$, so $t' = \tau'$, and we are done. Suppose $t = F t_1 \dots t_n$ where $F \in L^f$ is n-ary and $t_1, \dots, t_n$ are L-terms. Using the facts on admissible words at the end of Section 2.1, including exercise 5, we see that $t' = F t_1' \dots t_n'$ where for some $i \in \{1, \dots, n\}$ we have $t_j = t_j'$ for all $j \neq i$, $j \in \{1, \dots, n\}$, and t_i' is obtained from t_i by replacing an occurrence of τ in t_i by τ'. Inductively we can assume that $\Sigma \vdash t_1 = t_1', \dots, \Sigma \vdash t_n = t_n'$, so by part (5) of Lemma 3.2.3 we have $\Sigma \vdash t = t'$. To get $\Sigma \vdash \varphi \leftrightarrow \varphi'$, first take care of the case that φ is atomic by a similar argument as for terms, and then proceed by the usual kind of induction on formulas. $\square$

Exercises.

(1) Let $(\varphi_i)_{i\in I}$ be a family of formulas with finite index set I. Then

$$\vdash \exists x \bigvee_i \varphi_i \longleftrightarrow \bigvee_i \exists x\varphi_i, \quad \vdash \forall x \bigwedge_i \varphi_i \longleftrightarrow \bigwedge_i \forall x\varphi_i.$$

(2) A formula is said to be *unnested* if each atomic subformula has the form $Rx_1 \dots x_m$ with m-ary $R \in L^r \cup \{\top, \bot, =\}$ and distinct variables $x_1, \dots, x_m$, or the form $fx_1 \dots x_n = x_{n+1}$ with n-ary $f \in L^f$ and distinct variables $x_1, \dots, x_{n+1}$. (This allows $\top$ and $\bot$ as atomic subformulas of unnested formulas.) Then:
(i) each atomic formula $\varphi(y_1, \dots, y_n)$ is equivalent to an unnested existential formula $\varphi_1(y_1, \dots, y_n)$, and also to an unnested universal formula $\varphi_2(y_1, \dots, y_n)$.
(ii) each formula $\varphi(y_1, \dots, y_n)$ is equivalent to an unnested formula $\varphi^u(y_1, \dots, y_n)$.

(3) Let P be a unary relation symbol, Q be a binary relation symbol, and x, y distinct variables. Use prenex transformations to put

$$\forall x \exists y (P(x) \wedge Q(x,y)) \to \exists x \forall y (Q(x,y) \to P(y))$$

into prenex form.

3.4 Equational Classes and Universal Algebra

The term-structure $\mathcal{A}_\Sigma$ introduced in the proof of the Completeness Theorem also plays a role in what is called *universal algebra*. This is a general setting for constructing mathematical objects by *generators and relations*. Free groups, tensor products of various kinds, polynomial rings, and so on, are all special cases of a single construction in universal algebra.

In this section we fix a language L that has only function symbols, including at least one constant symbol. So L has no relation symbols. Instead of "L-structure" we say "L-algebra", and $\mathcal{A}$, $\mathcal{B}$ denote L-algebras. A substructure of $\mathcal{A}$ is also called a *subalgebra* of $\mathcal{A}$, and a *quotient algebra* of $\mathcal{A}$ is an L-algebra $\mathcal{A}/\sim$ where $\sim$ is a congruence on $\mathcal{A}$. We call $\mathcal{A}$ *trivial* if $|A| = 1$. There is up to isomorphism exactly one trivial L-algebra.

An L-*identity* is an L-sentence

$$\forall \vec{x}\big(s_1(\vec{x}) = t_1(\vec{x}) \wedge \dots \wedge s_n(\vec{x}) = t_n(\vec{x})\big), \qquad \vec{x} = (x_1, \dots, x_m)$$

where $x_1, \dots, x_m$ are distinct variables and $\forall \vec{x}$ abbreviates $\forall x_1 \dots \forall x_m$, and where $s_1, t_1, \dots, s_n, t_n$ are L-terms.

Given a set Σ of L-identities we define a Σ-*algebra* to be an L-algebra that satisfies all identities in Σ, in other words, a Σ-algebra is the same as a model of Σ. To such a Σ we associate the class $\mathrm{Mod}(\Sigma)$ of all Σ-algebras. A class $\mathcal{C}$ of L-algebras is said to be *equational* if there is a set Σ of L-identities such that $\mathcal{C} = \mathrm{Mod}(\Sigma)$.

Examples. With $L = L_{\mathrm{Gr}}$, Gr is a set of L-identities, and Mod(Gr), the class of groups, is the corresponding equational class of L-algebras. With $L = L_{\mathrm{Ri}}$, Ri is a set of L-identities, and Mod(Ri), the class of rings, is the corresponding equational class of L-algebras. If one adds to Ri the identity $\forall x \forall y(xy = yx)$ expressing the commutative law, then the corresponding class is the class of commutative rings.

Theorem 3.4.1. (G.Birkhoff) *Let $\mathcal{C}$ be a class of L-algebras. Then the class $\mathcal{C}$ is equational if and only if the following conditions are satisfied:*

(1) closure under isomorphism: if $\mathcal{A} \in \mathcal{C}$ and $\mathcal{A} \cong \mathcal{B}$, then $\mathcal{B} \in \mathcal{C}$.

(2) the trivial L-algebra belongs to $\mathcal{C}$;

(3) every subalgebra of any algebra in $\mathcal{C}$ belongs to $\mathcal{C}$;

(4) every quotient algebra of any algebra in $\mathcal{C}$ belongs to $\mathcal{C}$;

(5) the product of any family $(\mathcal{A}_i)$ of algebras in $\mathcal{C}$ belongs to $\mathcal{C}$.

It is easy to see that if $\mathcal{C}$ is equational, then conditions (1)–(5) are satisfied. (For (3) and (4) one can also appeal to the Exercises 6 and 7 of section 2.5) Towards a proof of the converse, we need some universal-algebraic considerations that are of interest beyond the connection to Birkhoff's theorem.
For the rest of this section we fix a set Σ of L-identities. Associated to Σ is the term algebra $\mathcal{A}_\Sigma$ whose elements are the equivalence classes $[t]$ of variable-free L-terms t, where two such terms s and t are equivalent iff $\Sigma \vdash s = t$.

Lemma 3.4.2. *$\mathcal{A}_\Sigma$ is a Σ-algebra.*

Proof. Consider an identity

$$\forall \vec{x}\big(s_1(\vec{x}) = t_1(\vec{x}) \wedge \cdots \wedge s_n(\vec{x}) = t_n(\vec{x})\big), \qquad \vec{x} = (x_1, \ldots, x_m)$$

in Σ, let $j \in \{1, \ldots, n\}$ and put $s = s_j$ and $t = t_j$. Let $a_1, \ldots, a_m \in A_\Sigma$ and put $\mathcal{A} = \mathcal{A}_\Sigma$. It suffices to show that then $s(\underline{a}_1, \ldots, \underline{a}_m)^{\mathcal{A}} = t(\underline{a}_1, \ldots, \underline{a}_m)^{\mathcal{A}}$. Take variable-free L-terms $\alpha_1, \ldots, \alpha_m$ such that $a_1 = [\alpha_1], \ldots, a_m = [\alpha_m]$. Then by part (1) of Corollary 3.2.5 we have $a_1 = \alpha_1^{\mathcal{A}}, \ldots, a_m = \alpha_m^{\mathcal{A}}$, so

$$s(\underline{a}_1, \ldots, \underline{a}_m)^{\mathcal{A}} = s(\alpha_1, \ldots, \alpha_m)^{\mathcal{A}}, \qquad t(\underline{a}_1, \ldots, \underline{a}_m)^{\mathcal{A}} = t(\alpha_1, \ldots, \alpha_m)^{\mathcal{A}}$$

by Lemma 2.6.1. Also, by part (1) of Corollary 3.2.5,

$$s(\alpha_1, \ldots, \alpha_m)^{\mathcal{A}} = [s(\alpha_1, \ldots, \alpha_m)], \qquad t(\alpha_1, \ldots, \alpha_m)^{\mathcal{A}} = [t(\alpha_1, \ldots, \alpha_m)].$$

Now $\Sigma \vdash s(\alpha_1, \ldots, \alpha_m) = t(\alpha_1, \ldots, \alpha_m)$, so $[s(\alpha_1, \ldots, \alpha_m)] = [t(\alpha_1, \ldots, \alpha_m)]$, and thus $s(\underline{a}_1, \ldots, \underline{a}_m)^{\mathcal{A}} = t(\underline{a}_1, \ldots, \underline{a}_m)^{\mathcal{A}}$, as desired. $\square$

Actually, we are going to show that $\mathcal{A}_\Sigma$ is a so-called *initial* Σ-algebra.

An **initial** Σ-algebra is a Σ-algebra $\mathcal{A}$ such that for any Σ-algebra $\mathcal{B}$ there is a unique homomorphism $\mathcal{A} \to \mathcal{B}$.

For example, the trivial group is an initial Gr-algebra, and the ring of integers is an initial Ri-algebra.

Suppose $\mathcal{A}$ and $\mathcal{B}$ are both initial Σ-algebras. Then there is a unique isomorphism $\mathcal{A} \to \mathcal{B}$. To see this, let i and j be the unique homomorphisms $\mathcal{A} \to \mathcal{B}$ and $\mathcal{B} \to \mathcal{A}$, respectively. Then we have homomorphisms $j \circ i : \mathcal{A} \to \mathcal{A}$ and $i \circ j : \mathcal{B} \to \mathcal{B}$, respectively, so necessarily $j \circ i = \mathrm{id}_A$ and $i \circ j = \mathrm{id}_B$, so i and j are isomorphisms. So if there is an initial Σ-algebra, it is unique up-to-unique-isomorphism.

Lemma 3.4.3. *$\mathcal{A}_\Sigma$ is an initial Σ-algebra.*

Proof. Let $\mathcal{B}$ be any Σ-algebra. Note that if $s, t \in T_L$ and $[s] = [t]$, then $\Sigma \vdash s = t$, so $s^{\mathcal{B}} = t^{\mathcal{B}}$. Thus we have a map

$$A_\Sigma \to B, \quad [t] \mapsto t^{\mathcal{B}}.$$

It is easy to check that this map is a homomorphism $\mathcal{A}_\Sigma \to \mathcal{B}$. By Exercise 3 in Section 2.4 it is the only such homomorphism. $\square$

Free algebras. Let I be an index set in what follows. Let $\mathcal{A}$ be a Σ-algebra and $(a_i)_{i \in I}$ an I-indexed family of elements of A. Then $\mathcal{A}$ is said to be a *free Σ-algebra on* (a_i) if for every Σ-algebra $\mathcal{B}$ and I-indexed family (b_i) of elements of B there is exactly one homomorphism $h : \mathcal{A} \to \mathcal{B}$ such that $h(a_i) = b_i$ for all $i \in I$. We also express this by "$\big(\mathcal{A}, (a_i)\big)$ is a free Σ-algebra". Finally, $\mathcal{A}$ itself is sometimes referred to as a free Σ-algebra if there is a family $(a_j)_{j \in J}$ in A such that $\big(\mathcal{A}, (a_j)\big)$ is a free Σ-algebra.

As an example, take $L = L_{\mathrm{Ri}}$ and $\mathrm{cRi} := \mathrm{Ri} \cup \{\forall x \forall y \; xy = yz\}$, where x, y are distinct variables. So the cRi-algebras are just the commutative rings. Let $\mathbb{Z}[X_1, \dots, X_n]$ be the ring of polynomials in distinct indeterminates $X_1, \dots, X_n$ over $\mathbb{Z}$. For any commutative ring R and elements $b_1, \dots, b_n \in R$ we have a unique ring homomorphism $\mathbb{Z}[X_1, \dots, X_n] \to R$ that sends X_i to b_i for $i = 1, \dots, n$, namely the *evaluation map* (or *substitution map*)

$$\mathbb{Z}[X_1, \dots, X_n] \to R, \quad f(X_1, \dots, X_n) \mapsto f(b_1, \dots, b_n).$$

Thus $\mathbb{Z}[X_1, \dots, X_n]$ is a free commutative ring on $(X_i)_{1 \le i \le n}$.

For a simpler example, let $L = L_{\mathrm{Mo}} := \{1, \cdot\} \subseteq L_{\mathrm{Gr}}$ be the language of monoids, and consider

$$\mathrm{Mo} := \{\forall x (1 \cdot x = x \wedge x \cdot 1 = x), \; \forall x \forall y \forall z \big((xy)z = x(yz)\big)\},$$

where x, y, z are distinct variables. A *monoid*, or *semigroup with identity*, is a model $\mathcal{A} = (A; 1, \cdot)$ of Mo, and we call $1 \in A$ the identity of the monoid $\mathcal{A}$, and $\cdot$ its product operation.

Let E^* be the set of words on an alphabet E, and consider E^* as a monoid by taking the empty word as its identity and the concatenation operation $(v, w) \mapsto vw$ as its product operation. Then E^* is a free monoid on the family $(e)_{e \in E}$ of words of length 1, because for any monoid $\mathcal{B}$ and elements $b_e \in B$ (for $e \in E$)

we have a unique monoid homomorphism $E^* \to \mathcal{B}$ that sends each $e \in E$ to b_e, namely,

$$e_1 \dots e_n \mapsto b_{e_1} \cdots b_{e_n}.$$

Remark. If $\mathcal{A}$ and $\mathcal{B}$ are both free Σ-algebras on (a_i) and (b_i) respectively, with same index set I, and $g : \mathcal{A} \to \mathcal{B}$ and $h : \mathcal{B} \to \mathcal{A}$ are the unique homomorphisms such that $g(a_i) = b_i$ and $h(b_i) = a_i$ for all i, then $(h \circ g)(a_i) = a_i$ for all i, so $h \circ g = \mathrm{id}_A$, and likewise $g \circ h = \mathrm{id}_B$, so g is an isomorphism with inverse h. Thus, given I, there is, up to unique isomorphism preserving I-indexed families, at most one free Σ-algebra on an I-indexed family of its elements.

We shall now construct free Σ-algebras as initial algebras by working in an extended language. Let $L_I := L \cup \{c_i : i \in I\}$ be the language L augmented by new constant symbols c_i for $i \in I$, where *new* means that $c_i \notin L$ for $i \in I$ and $c_i \neq c_j$ for distinct $i, j \in I$. So an L_I-algebra $\big(\mathcal{B}, (b_i)\big)$ is just an L-algebra $\mathcal{B}$ equipped with an I-indexed family (b_i) of elements of B. Let Σ_I be Σ viewed as a set of L_I-identities. Then a free Σ-algebra on an I-indexed family of its elements is just an initial Σ_I-algebra. In particular, the Σ_I-algebra $\mathcal{A}_{\Sigma_I}$ is a free Σ-algebra on $([c_i])$. Thus, up to unique isomorphism of Σ_I-algebras, there is a unique free Σ-algebra on an I-indexed family of its elements.

Let $(\mathcal{A}, (a_i)_{i\in I})$ be a free Σ-algebra. Then $\mathcal{A}$ is generated by (a_i). To see why, let $\mathcal{B}$ be the subalgebra of $\mathcal{A}$ generated by (a_i). Then we have a unique homomorphism $h : \mathcal{A} \to \mathcal{B}$ such that $h(a_i) = a_i$ for all $i \in I$, and then the composition

$$\mathcal{A} \to \mathcal{B} \to \mathcal{A}$$

is necessarily $\mathrm{id}_{\mathcal{A}}$, so $\mathcal{B} = \mathcal{A}$.

Let $\mathcal{B}$ be any Σ-algebra, and take any family $(b_j)_{j\in J}$ in B that generates $\mathcal{B}$. Take a free Σ-algebra $\big(\mathcal{A}, (a_j)_{j\in J}\big)$, and take the unique homomorphism $h : \big(\mathcal{A}, (a_j)\big) \to \big(\mathcal{B}, (b_j)\big)$. Then $h(t^{\mathcal{A}}(a_{j_1}, \dots, a_{j_n}) = t^{\mathcal{B}}(b_{j_1}, \dots, b_{j_n})$ for all L-term $t(x_1, \dots, x_n)$ and $j_1, \dots, j_n \in J$, so $h(A) = B$, and thus h induces an isomorphism $\mathcal{A}/{\sim_h} \cong \mathcal{B}$. We have shown:

Every Σ-algebra is isomorphic to a quotient of a free Σ-algebra.

This fact can sometimes be used to reduce problems on Σ-algebras to the case of free Σ-algebras; see the next subsection for an example.

Proof of Birkhoff's theorem. Let us say that a class $\mathcal{C}$ of L-algebras is *closed* if it has properties (1)–(5) listed in Theorem 3.4.1. Assume $\mathcal{C}$ is closed; we have to show that then $\mathcal{C}$ is equational. Indeed, let $\Sigma(\mathcal{C})$ be the set of L-identities

$$\forall \vec{x}\big(s(\vec{x}) = t(\vec{x})\big)$$

that are true in all algebras of $\mathcal{C}$. It is clear that each algebra in $\mathcal{C}$ is a $\Sigma(\mathcal{C})$-algebra, and it remains to show that every $\Sigma(\mathcal{C})$-algebra belongs to $\mathcal{C}$. Here is the key fact from which this will follow:

Claim. If $\mathcal{A}$ is an initial $\Sigma(\mathcal{C})$-algebra, then $\mathcal{A} \in \mathcal{C}$.

To prove this claim we take $\mathcal{A} := \mathcal{A}_{\Sigma(\mathcal{C})}$. For $s, t \in T_L$ such that $s = t$ does not belong to $\Sigma(\mathcal{C})$ we pick $\mathcal{B}_{s,t} \in \mathcal{C}$ such that $\mathcal{B}_{s,t} \models s \neq t$, and we let $h_{s,t} : \mathcal{A} \to \mathcal{B}_{s,t}$ be the unique homomorphism, so $h_{s,t}([s]) \neq h_{s,t}([t])$. Let $\mathcal{B} := \prod \mathcal{B}_{s,t}$ where the product is over all pairs (s,t) as above, and let $h : \mathcal{A} \to \mathcal{B}$ be the homomorphism given by $h(a) = \big(h_{s,t}(a)\big)$. Note that $\mathcal{B} \in \mathcal{C}$. Then h is injective. To see why, let $s, t \in T_L$ be such that $[s] \neq [t]$ in $A = A_{\Sigma(\mathcal{C})}$. Then $s = t$ does not belong to $\Sigma(\mathcal{C})$, so $h_{s,t}([s]) \neq h_{s,t}([t])$, and thus $h([s]) \neq h([t])$. This injectivity gives $\mathcal{A} \cong h(\mathcal{A}) \subseteq \mathcal{B}$, so $\mathcal{A} \in \mathcal{C}$. This finishes the proof of the claim.

Now, every $\Sigma(\mathcal{C})$-algebra is isomorphic to a quotient of a free $\Sigma(\mathcal{C})$-algebra, so it remains to show that free $\Sigma(\mathcal{C})$-algebras belong to $\mathcal{C}$. Let $\mathcal{A}$ be a free $\Sigma(\mathcal{C})$-algebra on $(a_i)_{i \in I}$. Let $\mathcal{C}_I$ be the class of all L_I-algebras $\big(\mathcal{B}, (b_i)\big)$ with $\mathcal{B} \in \mathcal{C}$. It is clear that $\mathcal{C}_I$ is closed as a class of L_I-algebras. Now, $\big(\mathcal{A}, (a_i)\big)$ is easily seen to be an initial $\Sigma(\mathcal{C}_I)$-algebra. By the claim above, applied to $\mathcal{C}_E$ in place of $\mathcal{C}$, we obtain $\big(\mathcal{A}, (a_i)\big) \in \mathcal{C}_I$, and thus $\mathcal{A} \in \mathcal{C}$.

Generators and relations. Let G be any set. Then we have a Σ-algebra $\mathcal{A}$ with a map $\iota : G \to A$ such that for any Σ-algebra $\mathcal{B}$ and any map $j : G \to B$ there is a unique homomorphism $h : \mathcal{A} \to \mathcal{B}$ such that $h \circ \iota = j$; in other words, $\mathcal{A}$ is a free as a Σ-algebra on $(\iota g)_{g \in G}$. Note that if $(\mathcal{A}', \iota')$ (with $\iota' : G \to A'$) has the same universal property as $(\mathcal{A}, \iota)$, then the unique homomorphism $h : \mathcal{A} \to \mathcal{A}'$ such that $h \circ \iota = \iota'$ is an isomorphism, so this universal property determines the pair $(\mathcal{A}, \iota)$ up-to-unique-isomorphism. So there is no harm in calling $(\mathcal{A}, \iota)$ *the free Σ-algebra on G*. Note that $\mathcal{A}$ is generated as an L-algebra by ιG.

Here is a particular way of constructing the free Σ-algebra on G. Take the language $L_G := L \cup G$ (disjoint union) with the elements of G as constant symbols. Let $\Sigma(G)$ be Σ considered as a set of L_G-identities. Then $\mathcal{A} := \mathcal{A}_{\Sigma(G)}$ as a Σ-algebra with the map $g \mapsto [g] : G \to A_{\Sigma(G)}$ is the free Σ-algebra on G.

Next, let R be a set of sentences $s(\vec{g}) = t(\vec{g})$ where $s(x_1, \ldots, x_n)$ and $t(x_1, \ldots, x_n)$ are L-terms and $\vec{g} = (g_1, \ldots, g_n) \in G^n$ (with n depending on the sentence). We wish to construct the *Σ-algebra generated by G with R as set of relations.*[1] This object is described up-to-isomorphism in the next lemma.

Lemma 3.4.4. *There is a Σ-algebra $\mathcal{A}(G,R)$ with a map $i : G \to A(G,R)$ such that:*

(1) $\mathcal{A}(G,R) \models s(i\vec{g}) = t(i\vec{g})$ for all $s(\vec{g}) = t(\vec{g})$ in R;

(2) for any Σ-algebra $\mathcal{B}$ and map $j : G \to B$ with $\mathcal{B} \models s(j\vec{g}) = t(j\vec{g})$ for all $s(\vec{g}) = t(\vec{g})$ in R, there is a unique homomorphism $h : \mathcal{A}(G,R) \to \mathcal{B}$ such that $h \circ i = j$.

Proof. Let $\Sigma(R) := \Sigma \cup R$, viewed as a set of L_G-sentences, let $\mathcal{A}(G,R) := \mathcal{A}_{\Sigma(R)}$, and define $i : G \to A(G,R)$ by $i(g) = [g]$. As before one sees that the universal property of the lemma is satisfied. □

[1]The use of the term "relations" here has nothing to do with n-ary relations on sets.

Chapter 4

Some Model Theory

In this chapter we first derive the Löwenheim-Skolem Theorem. Next we develop some basic methods related to proving completeness of a given set of axioms: Vaught's Test, back-and-forth, quantifier elimination. Each of these methods, when applicable, achieves a lot more than just establishing completeness.

4.1 Löwenheim-Skolem; Vaught's Test

Below, the cardinality of a structure is defined to be the cardinality of its underlying set. In this section we have the same conventions concerning L, $\mathcal{A}$, t, φ, ψ, θ, σ and Σ as in the beginning of Section 2.6, unless specified otherwise.

Theorem 4.1.1 (Countable Löwenheim-Skolem Theorem).
Suppose L is countable and Σ has a model. Then Σ has a countable model.

Proof. Since Var is countable, the hypothesis that L is countable yields that the set of L-sentences is countable. Hence the language

$$L \cup \{c_\sigma : \Sigma \vdash \sigma \text{ where } \sigma \text{ is an } L\text{-sentence of the form } \exists x \varphi(x)\}$$

is countable, that is, adding witnesses keeps the language countable. The union of countably many countable sets is countable, hence the set L_∞ constructed in the proof of the Completeness Theorem is countable. It follows that there are only countably many variable-free L_∞-terms, hence $\mathcal{A}_{\Sigma_\infty}$ is countable, and thus its reduct $\mathcal{A}_{\Sigma_\infty}|_L$ is a countable model of Σ. ☐

Remark. The above proof is the first time that we used the countability of $\text{Var} = \{\mathsf{v}_0, \mathsf{v}_1, \mathsf{v}_2, \dots\}$. As promised in Section 2.4, we shall now indicate why the Downward Löwenheim-Skolem Theorem goes through without assuming that Var is countable.

Suppose that Var is uncountable. Take a countably infinite subset $\text{Var}' \subseteq \text{Var}$. Then each sentence is equivalent to one whose variables are all from Var'. By replacing each sentence in Σ by an equivalent one all whose variables are from Var$'$, we obtain a countable set Σ' of sentences such that Σ and Σ' have the same models. As in the proof above, we obtain a countable model of Σ' working throughout in the setting where only variables from Var$'$ are used in terms and formulas. This model is a countable model of Σ.

The following test can be useful in showing that a set of axioms Σ is complete.

Proposition 4.1.2 (Vaught's Test). *Let L be countable, and suppose Σ has a model, and that all countable models of Σ are isomorphic. Then Σ is complete.*

Proof. Suppose Σ is not complete. Then there is σ such that $\Sigma \nvdash \sigma$ and $\Sigma \nvdash \neg\sigma$. Hence by the Löwenheim-Skolem Theorem there is a countable model $\mathcal{A}$ of Σ such that $\mathcal{A} \not\models \sigma$, and there is a countable model $\mathcal{B}$ of Σ such that $\mathcal{B} \not\models \neg\sigma$. We have $\mathcal{A} \cong \mathcal{B}$, $\mathcal{A} \models \neg\sigma$ and $\mathcal{B} \models \sigma$, contradiction. $\square$

Example. Let $L = \emptyset$, so the L-structures are just the non-empty sets. Let $\Sigma = \{\sigma_1, \sigma_2, \ldots\}$ where

$$\sigma_n = \exists x_1 \ldots \exists x_n \bigwedge_{1 \le i < j \le n} x_i \neq x_j$$

The models of Σ are exactly the infinite sets. All countable models of Σ are countably infinite and hence isomorphic to $\mathbf{N}$. Thus by Vaught's Test Σ is complete.

In this example the hypothesis of Vaught's Test is trivially satisfied. In other cases it may require work to check this hypothesis. One general method in model theory, *Back-and-Forth*, is often used to verify the hypothesis of Vaught's Test. The proof of the next theorem shows Back-and-Forth in action. To formulate that theorem we define a *totally ordered set* to be a structure $(A;\, <)$ for the language L_{O} that satisfies the following axioms (where x, y, z are distinct variables):

$$\forall x (x \not< x), \quad \forall x \forall y \forall z \big((x < y \wedge y < z) \rightarrow x < z \big), \quad \forall x \forall y (x < y \vee x = y \vee y < x).$$

Such a totally ordered set is said to be *dense* if it satisfies in addition the axiom $\forall x \forall y (x < y \rightarrow \exists z (x < z \wedge z < y))$, and it is said to *have no endpoints* if it satisfies the axiom $\forall x \exists y \exists z (y < x \wedge x < z)$. For example, $(\mathbf{Q};\, <)$ and $(\mathbf{R};\, <)$ are dense totally ordered sets without endpoints.

Theorem 4.1.3 (Cantor). *Any two countable dense totally ordered sets without endpoints are isomorphic.*

Proof. Let $(A;\, <)$ and $(B;\, <)$ be countable dense totally ordered sets without endpoints. So $A = \{a_n : n \in \mathbf{N}\}$ and $B = \{b_n : n \in \mathbf{N}\}$. We define by recursion a sequence (α_n) in A and a sequence (β_n) in B as follows: put $\alpha_0 := a_0$ and $\beta_0 := b_0$. Let $n > 0$, and suppose we have distinct $\alpha_0, \ldots, \alpha_{n-1}$ in A and distinct $\beta_0, \ldots, \beta_{n-1}$ in B such that for all $i, j < n$ we have $\alpha_i < \alpha_j \iff \beta_i < \beta_j$. Then we define $\alpha_n \in A$ and $\beta_n \in B$ as follows:

Case 1: n is even. (Here we go **forth**.) First take $k \in \mathbf{N}$ minimal such that $a_k \notin \{\alpha_0, \dots, \alpha_{n-1}\}$; then take $l \in \mathbf{N}$ minimal such that b_l *is situated with respect to* $\beta_0, \dots, \beta_{n-1}$ *as* a_k *is situated with respect to* $\alpha_0, \dots, \alpha_{n-1}$, that is, l is minimal such that for $i = 0, \dots, n-1$ we have: $\alpha_i < a_k \iff \beta_i < b_l$, and $\alpha_i > a_k \iff \beta_i > b_l$. (The reader should check that such an l exists: that is where density and "no endpoints" come in); put $\alpha_n := a_k$ and $\beta_n := b_l$.

Case 2: n is odd. (Here we go **back**.) First take $l \in \mathbf{N}$ minimal such that $b_l \notin \{\beta_0, \dots, \beta_{n-1}\}$; next take $k \in \mathbf{N}$ minimal such that a_k *is situated with respect to* $\alpha_0, \dots, \alpha_{n-1}$ *as* b_l *is situated with respect to* $\beta_0, \dots, \beta_{n-1}$, that is, k is minimal such that for $i = 0, \dots, n-1$ we have: $\alpha_i < a_k \iff \beta_i < b_l$, and $\alpha_i > a_k \iff \beta_i > b_l$. Put $\beta_n := b_l$ and $\alpha_n := a_k$.

One proves easily by induction on n that then $a_n \in \{\alpha_0, \dots, \alpha_{2n}\}$ and $b_n \in \{\beta_0, \dots, \beta_{2n}\}$. Thus we have a bijection $\alpha_n \mapsto \beta_n : A \to B$, and this bijection is an isomorphism $(A;\ <) \to (B;\ <)$. ☐

In combination with Vaught's Test this gives

Corollary 4.1.4. *The set of axioms defining dense totally ordered sets without endpoints is complete.*

In the results below we let κ denote an infinite cardinal. Recall that the set of ordinals $\lambda < \kappa$ has cardinality κ. We have the following generalization of the Löwenheim-Skolem theorem.

Theorem 4.1.5 (Generalized Löwenheim-Skolem Theorem). *Suppose L has size at most κ and Σ has an infinite model. Then Σ has a model of cardinality κ.*

Proof. Let $\{c_\lambda\}_{\lambda<\kappa}$ be a family of κ new constant symbols that are not in L and are pairwise distinct (that is, $c_\lambda \neq c_\mu$ for $\lambda < \mu < \kappa$). Let $L' = L \cup \{c_\lambda : \lambda < \kappa\}$ and let $\Sigma' = \Sigma \cup \{c_\lambda \neq c_\mu : \lambda < \mu < \kappa\}$. We claim that Σ' is consistent. To see this it suffices to show that, given any finite set $\Lambda \subseteq \kappa$, the set of L'-sentences

$$\Sigma_\Lambda := \Sigma \cup \{c_\lambda \neq c_\mu : \lambda, \mu \in \Lambda, \lambda \neq \mu\}$$

has a model. Take an infinite model $\mathcal{A}$ of Σ. We make an L'-expansion $\mathcal{A}_\Lambda$ of $\mathcal{A}$ by interpreting distinct c_λ's with $\lambda \in \Lambda$ by distinct elements of A, and interpreting the c_λ's with $\lambda \notin \Lambda$ arbitrarily. Then $\mathcal{A}_\Lambda$ is a model of Σ_Λ, which establishes the claim.
Note that L' also has size at most κ. The same arguments we used in proving the countable version of the Löwenheim-Skolem Theorem show that then Σ' has a model $\mathcal{B}' = (\mathcal{B}, (b_\lambda)_{\lambda<\kappa})$ of cardinality at most κ. We have $b_\lambda \neq b_\mu$ for $\lambda < \mu < \kappa$, hence $\mathcal{B}$ is a model of Σ of cardinality κ. ☐

The next proposition is Vaught's Test for arbitrary languages and cardinalities.

Proposition 4.1.6. *Suppose L has size at most κ, Σ has a model and all models of Σ are infinite. Suppose also that any two models of Σ of cardinality κ are isomorphic. Then Σ is complete.*

Proof. Let σ be an L-sentence and suppose that $\Sigma \nvDash \sigma$ and $\Sigma \nvDash \neg\sigma$. We will derive a contradiction. First $\Sigma \nvDash \sigma$ means that $\Sigma \cup \{\neg\sigma\}$ has a model. Similarly $\Sigma \nvDash \neg\sigma$ means that $\Sigma \cup \{\sigma\}$ has a model. These models must be infinite since they are models of Σ, so by the Generalized Löwenheim-Skolem Theorem $\Sigma \cup \{\neg\sigma\}$ has a model $\mathcal{A}$ of cardinality κ, and $\Sigma \cup \{\sigma\}$ has a model $\mathcal{B}$ of cardinality κ. By assumption $\mathcal{A} \cong \mathcal{B}$, contradicting that $\mathcal{A} \models \neg\sigma$ and $\mathcal{B} \models \sigma$. □

We now discuss in detail one particular application of this generalized Vaught Test. Fix a field F. A *vector space over* F is an abelian (additively written) group V equipped with a scalar multiplication operation

$$F \times V \longrightarrow V : (\lambda, v) \longmapsto \lambda v$$

such that for all $\lambda, \mu \in F$ and all $v, w \in V$ we have
(i) $(\lambda + \mu)v = \lambda v + \mu v$
(ii) $\lambda(v + w) = \lambda v + \lambda w$
(iii) $1v = v$
(iv) $(\lambda\mu)v = \lambda(\mu v)$.

Let L_F be the language of vector spaces over F: it extends the language $L_{\mathrm{Ab}} = \{0, -, +\}$ of abelian groups with unary function symbols f_λ, one for each $\lambda \in F$; a vector space over F is viewed as an L_F-structure by interpreting each f_λ as the function $v \longmapsto \lambda v$. One easily specifies a set Σ_F of sentences whose models are exactly the vector spaces over F. Note that Σ_F is not complete since the trivial vector space satisfies $\forall x(x = 0)$ but F viewed as vector space over F does not. Moreover, if F is finite, then we have both non-trivial finite vector spaces and non-trivial infinite vector spaces; to avoid these special cases we are going to restrict attention to infinite vector spaces over F. Let $v_1, v_2, \ldots$ be a sequence of distinct variables and put

$$\Sigma_F^\infty := \Sigma_F \cup \{\exists v_1 \ldots \exists v_n \bigwedge_{1 \le i < j \le n} v_i \neq v_j \ : \ n > 0\}$$

So the models of Σ_F^∞ are exactly the infinite vector spaces over F. Note that if F itself is infinite then each non-trivial vector space over F is infinite.

We will need the following facts about vector spaces V and W over F. (Proofs can be found in various texts.)

Fact.
(a) *V has a* basis *B, that is, $B \subseteq V$, and for each vector $v \in V$ there is a unique family $(\lambda_b)_{b \in B}$ of scalars (elements of F) such that $\{b \in B : \lambda_b \neq 0\}$ is finite and $v = \Sigma_{b \in B} \lambda_b b$.*
(b) *Any two bases B and C of V have the same cardinality.*
(c) *If V has basis B and W has basis C, then any bijection $B \to C$ extends uniquely to an isomorphism $V \to W$.*
(d) *Let B be a basis of V. Then $|V| = |B| \cdot |F|$ provided either $|F|$ or $|B|$ is infinite. If both are finite then $|V| = |F|^{|B|}$.*

Theorem 4.1.7. Σ_F^∞ *is complete.*

Proof. Take a $\kappa > |F|$. In particular L_F has size at most κ. Let V be a vector space over F of cardinality κ. Then a basis of V must also have size κ by property (d) above. Thus any two vector spaces over F of cardinality κ have bases of cardinality κ and thus are isomorphic. It follows by the Generalized Vaught Test that Σ_F^∞ is complete. □

Remark. Theorem 4.1.7 and Exercise 1 imply for instance that if $F = \mathbf{R}$ then all non-trivial vector spaces over F satisfy exactly the same sentences in L_F.

Using the generalized Vaught Test one can also prove that ACF(0) (whose models are the algebraically closed fields of characteristic 0) is complete. The proof is similar except that you need to work with transcendence bases and be familiar with the notion *algebraic closure of a field*.

If the hypothesis of Vaught's Test (or its generalization) is satisfied, then many things follow of which completeness is only one; it goes beyond the scope of these notes to develop this large chapter of model theory, which goes under the name of "categoricity in power".

Exercises.

(1) Let $L = \{U\}$ where U is a unary relation symbol. Consider the L-structure $(\mathbf{Z};\ \mathbf{N})$. Give an *informative* description of a complete set of L-sentences true in $(\mathbf{Z};\ \mathbf{N})$. (A description like $\{\sigma : (\mathbf{Z};\ \mathbf{N}) \models \sigma\}$ is not informative. An explicit, possibly infinite, list of axioms is required. Hint: Make an educated guess and try to verify it using Vaught's Test or one of its variants.)

(2) Let Σ_1 and Σ_2 be sets of L-sentences such that no symbol of L occurs in both Σ_1 and Σ_2. Suppose Σ_1 and Σ_2 have infinite models. Then $\Sigma_1 \cup \Sigma_2$ has a model.

(3) Let $L = \{S\}$ where S is a unary function symbol. Consider the L-structure $(\mathbf{Z};\ S)$ where $S(a) = a + 1$ for $a \in \mathbf{Z}$. Give an *informative* description of a complete set of L-sentences true in $(\mathbf{Z};\ S)$.

4.2 Elementary Equivalence and Back-and-Forth

In the rest of this chapter we relax notation, and just write $\varphi(a_1, \dots, a_n)$ for an L_A-formula $\varphi(\underline{a}_1, \dots, \underline{a}_n)$, where $\mathcal{A} = (A; \dots)$ is an L-structure, $\varphi(x_1, \dots, x_n)$ an L_A-formula, and $(a_1, \dots, a_n) \in A^n$.

In this section $\mathcal{A}$ and $\mathcal{B}$ denote L-structures. We say that $\mathcal{A}$ and $\mathcal{B}$ are *elementarily equivalent* (notation: $\mathcal{A} \equiv \mathcal{B}$) if they satisfy the same L-sentences. Thus by the previous section $(\mathbf{Q};\ <) \equiv (\mathbf{R};\ <)$, and any two infinite vector spaces over a given field F are elementarily equivalent.

A *partial isomorphism* from $\mathcal{A}$ to $\mathcal{B}$ is a bijection $\gamma : X \to Y$ with $X \subseteq A$ and $Y \subseteq B$ such that

(i) for each m-ary $R \in L^r$ and $a_1, \dots, a_m \in X$

$$(a_1, \dots, a_m) \in R^{\mathcal{A}} \iff (\gamma a_1, \dots, \gamma a_m) \in R^{\mathcal{B}}.$$

(ii) for each n-ary $f \in L^f$ and $a_1, \ldots, a_n, a_{n+1} \in X$

$$f^{\mathcal{A}}(a_1, \ldots, a_n) = a_{n+1} \Longleftrightarrow f^{\mathcal{B}}(\gamma a_1, \ldots, \gamma a_n) = \gamma(a_{n+1}).$$

Example: suppose $\mathcal{A} = (A;\ <)$ and $\mathcal{B} = (B;\ <)$ are totally ordered sets, and $a_1, \ldots, a_N \in A$ and $b_1, \ldots, b_N \in B$ are such that $a_1 < a_2 < \cdots < a_N$ and $b_1 < b_2 < \cdots < b_N$; then the map $a_i \mapsto b_i : \{a_1, \ldots, a_N\} \to \{b_1, \ldots, b_N\}$ is a partial isomorphism from $\mathcal{A}$ to $\mathcal{B}$.

A *back-and-forth system* from $\mathcal{A}$ to $\mathcal{B}$ is a nonempty collection Γ of partial isomorphisms from $\mathcal{A}$ to $\mathcal{B}$ such that

(i) ("Forth") for each $\gamma \in \Gamma$ and $a \in A$ there is a $\gamma' \in \Gamma$ such that γ' extends γ and $a \in \text{domain}(\gamma')$;

(ii) ("Back") for each $\gamma \in \Gamma$ and $b \in B$ there is a $\gamma' \in \Gamma$ such that γ' extends γ and $b \in \text{codomain}(\gamma')$.

We say that $\mathcal{A}$ and $\mathcal{B}$ are *back-and-forth equivalent* (notation: $\mathcal{A} \equiv_{bf} \mathcal{B}$) if there exists a back-and-forth system from $\mathcal{A}$ to $\mathcal{B}$.

Cantor's proof in the previous section generalizes as follows:

Proposition 4.2.1. *Suppose $\mathcal{A}$ and $\mathcal{B}$ are countable and $\mathcal{A} \equiv_{bf} \mathcal{B}$. Then $\mathcal{A} \cong \mathcal{B}$.*

Proof. Let Γ be a back-and-forth system from $\mathcal{A}$ to $\mathcal{B}$. We proceed as in the proof of Cantor's theorem, and construct a sequence (γ_n) of partial isomorphisms from $\mathcal{A}$ to $\mathcal{B}$ such that each γ_{n+1} extends γ_n, $A = \bigcup_n \text{domain}(\gamma_n)$ and $B = \bigcup_n \text{codomain}(\gamma_n)$. Then the map $A \to B$ that extends each γ_n is an isomorphism $\mathcal{A} \to \mathcal{B}$. $\square$

In applying this proposition and the next one in a concrete situation, the key is to guess a back-and-forth system. That is where insight and imagination (and experience) come in. In the following result we do not need to assume countability.

Proposition 4.2.2. *If $\mathcal{A} \equiv_{bf} \mathcal{B}$, then $\mathcal{A} \equiv \mathcal{B}$.*

Proof. Suppose Γ is a back-and-forth system from $\mathcal{A}$ to $\mathcal{B}$. By induction on the number of logical symbols in *unnested* formulas $\varphi(y_1, \ldots, y_n)$ one shows that for each $\gamma \in \Gamma$ and $a_1, \ldots, a_n \in \text{domain}(\gamma)$ we have

$$\mathcal{A} \models \varphi(a_1, \ldots, a_n) \Longleftrightarrow \mathcal{B} \models \varphi(\gamma a_1, \ldots, \gamma a_n).$$

For $n = 0$ and using the result of Exercise 2 this yields $\mathcal{A} \equiv \mathcal{B}$. $\square$

4.3 Quantifier Elimination

In this section $x = (x_1, \ldots, x_n)$ is a tuple of distinct variables and y is a single variable distinct from $x_1, \ldots, x_n$.

Definition. Σ has *quantifier elimination* (QE) if every L-formula $\varphi(x)$ is Σ-equivalent to a quantifier free (short: q-free) L-formula $\varphi^{\text{qf}}(x)$.

In particular, if Σ has QE, then every L-sentence is Σ-equivalent to a q-free L-sentence. (Take $n = 0$ in the above definition.)

Remark. Suppose Σ has QE. Let L' be a language such that $L' \supseteq L$ and all symbols of $L' \smallsetminus L$ are constant symbols. (This includes the case $L' = L$.) Then every set Σ' of L'-sentences with $\Sigma' \supseteq \Sigma$ has QE. (To see this, check first that each L'-formula $\varphi'(x)$ has the form $\varphi(c, x)$ for some L-formula $\varphi(u, x)$, where $u = (u_1, \ldots, u_m)$ is a tuple of distinct variables distinct from $x_1, \ldots, x_n$, and $c = (c_1, \ldots, c_m)$ is a tuple of constant symbols in $L' \smallsetminus L$.)

A *basic conjunction in* L is by definition a conjunction of finitely many atomic and negated atomic L-formulas. Each q-free L-formula $\varphi(x)$ is equivalent to a disjunction $\varphi_1(x) \vee \cdots \vee \varphi_k(x)$ of basic conjunctions $\varphi_i(x)$ in L ("disjunctive normal form").

Lemma 4.3.1. *Suppose that for every basic conjunction $\theta(x, y)$ in L there is a q-free L-formula $\theta^{\text{qf}}(x)$ such that*

$$\Sigma \vdash \exists y \theta(x, y) \leftrightarrow \theta^{\text{qf}}(x).$$

Then Σ has QE.

Proof. Let us say that an L-formula $\varphi(x)$ has Σ-QE if it is Σ-equivalent to a q-free L-formula $\varphi^{\text{qf}}(x)$. Note that if the L-formulas $\varphi_1(x)$ and $\varphi_2(x)$ have Σ-QE, then $\neg\varphi_1(x)$, $(\varphi_1 \vee \varphi_2)(x)$, and $(\varphi_1 \wedge \varphi_2)(x)$ have Σ-QE.

Next, let $\varphi(x) = \exists y \psi(x, y)$, and suppose inductively that the L-formula $\psi(x, y)$ has Σ-QE. Hence $\psi(x, y)$ is Σ-equivalent to a disjunction $\bigvee_i \psi_i(x, y)$ of basic conjunctions $\psi_i(x, y)$ in L, with i ranging over some finite index set. In view of the equivalence of $\exists y \bigvee_i \psi_i(x, y)$ with $\bigvee_i \exists y \psi_i(x, y)$ we obtain

$$\Sigma \vdash \varphi(x) \longleftrightarrow \bigvee_i \exists y \psi_i(x, y).$$

Each $\exists y \psi_i(x, y)$ has Σ-QE, by hypothesis, so $\varphi(x)$ has Σ-QE.

Finally, let $\varphi(x) = \forall y \psi(x, y)$, and suppose inductively that the L-formula $\psi(x, y)$ has Σ-QE. This case reduces to the previous case since $\varphi(x)$ is equivalent to $\neg \exists y \neg \psi(x, y)$. $\square$

Remark. In the structure $(\mathbf{R};\ <, 0, 1, +, -, \cdot)$ the formula

$$\varphi(a, b, c) := \exists y (a y^2 + b y + c = 0)$$

is equivalent to the q-free formula

$$(b^2 - 4ac \geq 0 \wedge a \neq 0) \vee (a = 0 \wedge b \neq 0) \vee (a = 0 \wedge b = 0 \wedge c = 0).$$

Note that this equivalence gives an effective test for the existence of a y with a certain property, which avoids in particular having to check an infinite number of values of y (even uncountably many in the case above). This illustrates the kind of property QE is.

Lemma 4.3.2. *Suppose Σ has* QE *and $\mathcal{B}$ and $\mathcal{C}$ are models of Σ with a common substructure $\mathcal{A}$ (we do not assume $\mathcal{A} \models \Sigma$). Then $\mathcal{B}$ and $\mathcal{C}$ satisfy the same L_A-sentences.*

Proof. Let σ be an L_A-sentence. We have to show $\mathcal{B} \models \sigma \Leftrightarrow \mathcal{C} \models \sigma$. Write σ as $\varphi(a)$ with $\varphi(x)$ an L-formula and $a \in A^n$. Take a q-free L-formula $\varphi^{qf}(x)$ that is Σ-equivalent to $\varphi(x)$. Then $\mathcal{B} \models \sigma$ iff $\mathcal{B} \models \varphi^{qf}(a)$ iff $\mathcal{A} \models \varphi^{qf}(a)$ (by Exercise 6) iff $\mathcal{C} \models \varphi^{qf}(a)$ (by the same exercise) iff $\mathcal{C} \models \sigma$. $\square$

Corollary 4.3.3. *Suppose Σ has a model, has* QE*, and there exists an L-structure that can be embedded into every model of Σ. Then Σ is complete.*

Proof. Take an L-structure $\mathcal{A}$ that can be embedded into every model of Σ. Let $\mathcal{B}$ and $\mathcal{C}$ be any two models of Σ. So $\mathcal{A}$ is isomorphic to a substructure of $\mathcal{B}$ and of $\mathcal{C}$. Then by a slight rewording of the proof of Lemma 4.3.2 (considering only L-sentences), we see that $\mathcal{B}$ and $\mathcal{C}$ satisfy the same L-sentences. It follows that Σ is complete. $\square$

Remark. We have seen that Vaught's test can be used to prove completeness. The above corollary gives another way of establishing completeness, and is often applicable when the hypothesis of Vaught's Test is not satisfied. Completeness is only one of the nice consequences of QE, and the easiest one to explain at this stage. The main impact of QE is rather that it gives access to the structural properties of definable sets. This will be reflected in exercises at the end of this section. Applications of model theory to other areas of mathematics often involve QE as a key step.

We mention without proof two examples of QE, and give a complete proof for a third example in the next section. The following theorem is due to Tarski and (independently) to Chevalley. It dates from around 1950.

Theorem 4.3.4. ACF *has* QE*.*

It is clear that ACF is not complete, since it says nothing about the characteristic: it doesn't prove $1 + 1 = 0$, nor does it prove $1 + 1 \neq 0$. However, ACF(0), which contains additional axioms forcing the characteristic to be 0, is complete by 4.3.3 and the fact that the ring of integers embeds in every algebraically closed field of characteristic 0. Tarski also established the following more difficult theorem, which is one of the key results in real algebraic geometry. (His original proof is rather long; there is a shorter one due to A. Seidenberg, and a very elegant short proof by A. Robinson based elementary model theoretic results.)

Definition. RCF is a set of axioms true in the ordered field $(\mathbf{R};\ <, 0, 1, -, +, \cdot)$ of real numbers. In addition to the ordered field axioms, it has the axiom $\forall x\,(x > 0 \rightarrow \exists y\,(x = y^2))$ (x, y distinct variables) and for each odd $n > 1$ the axiom

$$\forall x_1 \ldots \forall x_n \exists y\,(y^n + x_1 y^{n-1} + \cdots + x_n = 0)$$

where $x_1, \ldots, x_n, y$ are distinct variables.

Theorem 4.3.5. RCF *admits* QE *and is complete.*

Exercises. In (5) and (6), an *L-theory* is a set T of L-sentences such that for all L-sentences σ, if $T \vdash \sigma$, then $\sigma \in T$. An *axiomatization* of an L-theory T is a set Σ of L-sentences such that $T = \{\sigma : \sigma \text{ is an } L\text{-sentence and } \Sigma \vdash \sigma\}$.

(1) The subsets of $\mathbf{C}$ definable in $(\mathbf{C};\ 0, 1, -, +, \cdot)$ are exactly the finite subsets of $\mathbf{C}$ and their complements in $\mathbf{C}$. (Hint: use the fact that ACF has QE.)

(2) The subsets of $\mathbf{R}$ definable in the model $(\mathbf{R};\ <, 0, 1, -, +, \cdot)$ of RCF are exactly the finite unions of intervals of all kinds (including degenerate intervals with just one point) (Hint: use the fact that RCF has QE.)

(3) Let Eq_∞ be a set of axioms in the language $\{\sim\}$ (where $\sim$ is a binary relation symbol) that say:
(i) $\sim$ is an equivalence relation;
(ii) every equivalence class is infinite;
(iii) there are infinitely many equivalence classes.
Then Eq_∞ admits QE and is complete. (It is also possible to use Vaught's test to prove completeness.)

(4) Suppose that a set Σ of L-sentences has QE. Let the language L' extend L by new symbols of arity 0, and let $\Sigma' \supseteq \Sigma$ be a set of L'-sentences. Then Σ' (as a set of L'-sentences) also has QE.

(5) Suppose the L-theory T has QE. Then T has an axiomatization consisting of sentences $\forall x \exists y \varphi(x, y)$ and $\forall x \psi(x)$ where $\varphi(x, y)$ and $\psi(x)$ are q-free. (Hint: let Σ be the set of L-sentences provable from T that have the indicated form; show that Σ has QE, and is an axiomatization of T.)

(6) Assume the L-theory T has built-in Skolem functions, that is, for each basic conjunction $\varphi(x, y)$ there are L-terms $t_1(x), \dots, t_k(x)$ such that

$$\Sigma \vdash \exists y \varphi(x, y) \rightarrow \varphi(x, t_1 x)) \vee \varphi(x, t_k(x)).$$

Then T has QE, for every $\varphi(x, y)$ there are L-terms $t_1(x), \dots, t_k(x)$ such that $\Sigma \vdash \exists y \varphi(x, y) \rightarrow \varphi(x, t_1 x)) \vee \varphi(x, t_k(x))$, and T has an axiomatization consisting of sentences $\forall x \psi(x)$ where $\psi(x)$ is q-free.

4.4 Presburger Arithmetic

In this section we consider in some detail one example of a set of axioms that has QE, namely "Presburger Arithmetic." Essentially, this is a complete set of axioms for ordinary arithmetic of integers without multiplication, that is, the axioms are true in $(\mathbf{Z};\ 0, 1, +, -, <)$, and prove every sentence true in this structure. There is a mild complication in trying to obtain this completeness via QE: one can show (exercise) that for any q-free formula $\varphi(x)$ in the language $\{0, 1, +, -, <\}$ there is an $N \in \mathbf{N}$ such that either $(\mathbf{Z};\ 0, 1, +, -, <) \models \varphi(n)$ for all $n > N$ or $(\mathbf{Z};\ 0, 1, +, -, <) \models \neg\varphi(n)$ for all $n > N$. In particular, formulas such as $\exists y (x = y + y)$ or $\exists y (x = y + y + y)$ are not Σ-equivalent to any q-free formula in this language, for any set Σ of axioms true in $(\mathbf{Z};\ 0, 1, +, -, <)$.

To overcome this obstacle to QE we augment the language $\{0, 1, +, -, <\}$ by new unary relation symbols $P_1, P_2, P_3, P_4, \ldots$ to obtain the language L_{PrA} of *Presburger Arithmetic* (named after the Polish logician Presburger who was a student of Tarski). We expand $(\mathbf{Z};\ 0, 1, +, -, <)$ to the L_{PrA}-structure

$$\tilde{\mathbf{Z}} = (\mathbf{Z};\ 0, 1, +, -, <, \mathbf{Z}, 2\mathbf{Z}, 3\mathbf{Z}, 4\mathbf{Z}, \ldots)$$

that is, P_n is interpreted as the set $n\mathbf{Z}$. This structure satisfies the set PrA of *Presburger Axioms* which consists of the following sentences:

(i) the axioms of Ab for abelian groups;
(ii) the axioms expressing that $<$ is a total order;
(iii) $\forall x \forall y \forall z (x < y \to x + z < y + z)$ (*translation invariance of* $<$);
(iv) $0 < 1 \wedge \neg \exists y (0 < y < 1)$ (*discreteness axiom*);
(v) $\forall x \exists y \bigvee_{0 \le r < n} x = ny + r1, \quad n = 1, 2, 3, \ldots$ (*division with remainder*);
(vi) $\forall x (P_n x \leftrightarrow \exists y\, x = ny), \quad n = 1, 2, 3, \ldots$ (*defining axioms for* $P_1, P_2, \ldots$).

Here we have fixed distinct variables x, y, z for definiteness. In (v) and in the rest of this section r ranges over integers. Note that (v) and (vi) are infinite lists of axioms. Here are some elementary facts about models of PrA:

Proposition 4.4.1. *Let* $\mathcal{A} = (A;\ 0, 1, +, -, <, P_1^{\mathcal{A}}, P_2^{\mathcal{A}}, P_3^{\mathcal{A}}, \ldots) \models \mathrm{PrA}$. *Then*
(1) *There is a unique embedding* $\tilde{\mathbf{Z}} \longrightarrow \mathcal{A}$*; it sends* $k \in \mathbf{Z}$ *to* $k1 \in A$.
(2) *Given any* $n > 0$ *we have* $P_n^{\mathcal{A}} = n\mathcal{A}$*, where we regard* $\mathcal{A}$ *as an abelian group, and* $\mathcal{A}/n\mathcal{A}$ *has exactly* n *elements, namely* $0 + n\mathcal{A}, \ldots, (n-1)1 + n\mathcal{A}$.
(3) *For any* $n > 0$ *and* $a \in A$*, exactly one of the* $a, a+1, \ldots, a + (n-1)1$ *lies in* $n\mathcal{A}$*;*
(4) $\mathcal{A}$ *is torsion-free as an abelian group.*

Theorem 4.4.2. PrA *admits QE.*

Proof. Let $(x, y) = (x_1, \ldots, x_n, y)$ be a tuple of $n + 1$ distinct variables, and consider a basic conjunction $\varphi(x, y)$ in L_{PrA}. By Lemma 4.3.1 it suffices to show that $\exists y \varphi(x, y)$ is PrA-equivalent to a q-free formula $\psi(x)$. We may assume that each conjunct of φ is of one of the following types, for some integer $m \ge 1$ and L_{PrA}-term $t(x)$:

$$my = t(x), \qquad my < t(x), \qquad t(x) < my, \qquad P_n(my + t(x)).$$

To justify this assumption observe that if we had instead a conjunct $my \ne t(x)$ then we could replace it by $(my < t(x)) \vee (t(x) < my)$ and use the fact that $\exists y (\varphi_1(x, y) \vee \varphi_2(x, y))$ is equivalent to $\exists y \varphi_1(x, y) \vee \exists y \varphi_2(x, y)$. Similarly a negation $\neg P_n(my + t(x))$ can be replaced by the disjunction

$$P_n(my + t(x) + 1) \vee \ldots \vee P_n(my + t(x) + (n-1)1)$$

Also conjuncts in which y does not appear can be eliminated because

$$\vdash \exists y (\psi(x) \wedge \theta(x, y)) \longleftrightarrow \psi(x) \wedge \exists y \theta(x, y).$$

Since PrA $\vdash P_n(z) \leftrightarrow P_{rn}(rz)$ for $r > 0$ we can replace $P_n(my + t(x))$ by $P_{rn}(rmy + rt(x))$. Also, for $r > 0$ we can replace $my = t(x)$ by $rmy = rt(x)$, and likewise with $my < t(x)$ and $t(x) < my$. We can therefore assume that all conjuncts have the same "coefficient" m in front of the variable y. After all these reductions, and after rearranging conjuncts, $\varphi(x, y)$ has the form

$$\bigwedge_{h\in H} my = t_h(x) \wedge \bigwedge_{i\in I} t_i(x) < my \wedge \bigwedge_{j\in J} my < t_j(x) \wedge \bigwedge_{k\in K} P_{n(k)}(my + t_k(x))$$

where $m > 0$ and H, I, J, K are disjoint finite index sets. We allow some of these index sets to be empty in which case the corresponding conjunction can be left out.

Suppose that $H \neq \emptyset$, say $h' \in H$. Then the formula $\exists y\varphi(x, y)$ is PrA-equivalent to

$$P_m(t_{h'}(x)) \wedge \bigwedge_{h\in H} t_h(x) = t_{h'}(x) \wedge \bigwedge_{i\in I} t_i(x) < t_{h'}(x) \wedge \bigwedge_{j\in J} t_{h'}(x) < t_j(x)$$
$$\wedge \bigwedge_{k\in K} P_{n(k)}(t_{h'}(x) + t_k(x))$$

For the rest of the proof we assume that $H = \emptyset$.

To understand what follows, it may help to focus on the model $\tilde{\mathbf{Z}}$, although the arguments go through for arbitrary models of PrA. Fix any value $a \in \mathbf{Z}^n$ of x. Consider the system of linear congruences (with "unknown" y)

$$P_{n(k)}(my + t_k(a)), \qquad (k \in K),$$

which in more familar notation would be written as

$$my + t_k(a) \equiv 0 \mod n(k), \qquad (k \in K).$$

The solutions in $\mathbf{Z}$ of this system form a union of congruence classes modulo $N := \prod_{k\in K} n(k)$, where as usual we put $N = 1$ for $K = \emptyset$. This suggests replacing y successively by Nz, $1 + Nz$,. . . ,$(N - 1)1 + Nz$. Our precise claim is that $\exists y\varphi(x, y)$ is PrA-equivalent to the formula $\theta(x)$ given by

$$\bigvee_{r=0}^{N-1} \left(\bigwedge_{k\in K} P_{n(k)}((mr)1 + t_k(x)) \wedge \exists z \left(\bigwedge_{i\in I} t_i(x) < m(r1 + Nz) \right.\right.$$
$$\left.\left. \wedge \bigwedge_{j\in J} m(r1 + Nz) < t_j(x) \right)\right)$$

We prove this equivalence with $\theta(x)$ as follows. Suppose

$$\mathcal{A} = (A, \ldots) \models \text{PrA}, \quad a = (a_1, \ldots, a_n) \in A^n.$$

We have to show that $\mathcal{A} \models \exists y\varphi(a, y)$ if and only if $\mathcal{A} \models \theta(a)$. So let $b \in A$ be such that $\mathcal{A} \models \varphi(a, b)$. Division with remainder yields a $c \in A$ and an r such that $b = r1 + Nc$ and $0 \le r \le N - 1$. Note that then for $k \in K$,

$$mb + t_k(a) = m(r1 + Nc) + t_k(a) = (mr)1 + (mN)c + t_k(a) \in n(k)\mathcal{A}$$

and so $\mathcal{A} \models P_{n(k)}((mr)1 + t_k(a))$. Also,

$$\begin{aligned} t_i(a) &< m(r1 + Nc) \quad \text{for every } i \in I, \\ m(r1 + Nc) &< t_j(a) \quad \text{for every } j \in J. \end{aligned}$$

Therefore $\mathcal{A} \models \theta(a)$ with $\exists z$ witnessed by c. For the converse, suppose that the disjunct of $\theta(a)$ indexed by a certain $r \in \{0, \ldots, N - 1\}$ is true in $\mathcal{A}$, with $\exists z$ witnessed by $c \in A$. Then put $b = r1 + Nc$ and we get $\mathcal{A} \models \varphi(a, b)$. This proves the claimed equivalence.

Now that we have proved the claim we have reduced to the situation (after changing notation) where $H = K = \emptyset$ (i. e. no equations and no congruences). So $\varphi(x, y)$ now has the form

$$\bigwedge_{i \in I} t_i(x) < my \wedge \bigwedge_{j \in J} my < t_j(x)$$

If $J = \emptyset$ or $I = \emptyset$ then $\mathrm{PrA} \vdash \exists y\varphi(x, y) \leftrightarrow \top$. This leaves the case where both I and J are nonempty. So suppose $\mathcal{A} \models \mathrm{PrA}$ and that A is the underlying set of $\mathcal{A}$. For each value $a \in A^n$ of x there is $i_0 \in I$ such that $t_{i_0}(a)$ is maximal among the $t_i(a)$ with $i \in I$, and a $j_0 \in J$ such that $t_{j_0}(a)$ is minimal among the $t_j(a)$ with $j \in J$. Moreover each interval of m successive elements of A contains a multiple of m. Therefore $\exists y\varphi(x, y)$ is equivalent in $\mathcal{A}$ to the disjunction over all pairs $(i_0, j_0) \in I \times J$ of the q-free formula

$$\bigwedge_{i \in I} t_i(x) \le t_{i_0}(x) \wedge \bigwedge_{j \in J} t_{j_0}(x) \le t_j(x)$$
$$\wedge \bigvee_{r=1}^{m} \Big(P_m(t_{i_0}(x) + r1) \wedge (t_{i_0}(x) + r1 < t_{j_0}(x))\Big)$$

This completes the proof. Note that L_{PrA} does not contain the relation symbol $\le$; we just write $t \le t'$ to abbreviate $(t < t') \vee (t = t')$. $\square$

Remark. It now follows from Corollary 4.3.3 that PrA is complete: it has QE and $\tilde{\mathbf{Z}}$ can be embedded in every model.

Discussion. The careful reader will have noticed that the elimination procedure in the proof above is constructive: it describes an algorithm that, given any basic conjunction $\varphi(x, y)$ in L_{PrA} as input, constructs a q-free formula $\psi(x)$ of L_{PrA} such that $\mathrm{PrA} \vdash \exists y\varphi(x, y) \leftrightarrow \psi(x)$. In view of the equally constructive proof of Lemma 4.3.1 this yields an algorithm that, given any L_{PrA}-formula $\varphi(x)$ as input, constructs a q-free L_{PrA}-formula $\varphi^{\mathrm{qf}}(x)$ such that $\mathrm{PrA} \vdash \varphi(x) \leftrightarrow \varphi^{\mathrm{qf}}(x)$. (Thus PrA has *effective* QE.)

In particular, this last algorithm constructs for any L_{PrA}-sentence σ a q-free L_{PrA}-sentence σ^{qf} such that $\mathrm{PrA} \vdash \sigma \leftrightarrow \sigma^{\mathrm{qf}}$. Since we also have an obvious algorithm that, given any q-free L_{PrA}-sentence σ^{qf}, checks whether σ^{qf} is true in $\tilde{\mathbf{Z}}$, this yields an algorithm that, given any L_{PrA}-sentence σ, checks whether σ is true in $\tilde{\mathbf{Z}}$. Thus the structure $\tilde{\mathbf{Z}}$ is *decidable*. (A precise definition of decidability will be given in the next Chapter.) The algorithms above can easily be implemented by computer programs.

Let some L-structure $\mathcal{A}$ be given, and suppose we have an algorithm for deciding whether any given L-sentence is true in $\mathcal{A}$. Even if this algorithm can be implemented by a computer program, it does not guarantee that the program is of practical use, or *feasible*: on some moderately small inputs it might have to run for 10^{100} years before producing an output. This bad behaviour is not at all unusual: no (classical, sequential) algorithm for deciding the truth of L_{PrA}-sentences in $\tilde{\mathbf{Z}}$ is feasible in a precise technical sense. Results of this kind belong to *complexity theory*; this is an area where mathematics (logic, number theory,...) and computer science interact.

There do exist feasible *integer linear programming* algorithms that decide the truth in $\tilde{\mathbf{Z}}$ of sentences of a special form, and this shows another (very practical) side of complexity theory.

A *positive* impact of QE is that it yields structural properties of definable sets, as we discuss next for $\tilde{\mathbf{Z}}$.

Definition. Let d be a positive integer. An *arithmetic progression of modulus* d is a set of the form

$$\{k \in \mathbf{Z} : k \equiv r \mod d,\ \alpha < k < \beta\},$$

where $r \in \{0, \dots, d-1\}$, $\alpha, \beta \in \mathbf{Z} \cup \{-\infty, +\infty\}$, $\alpha < \beta$.

We leave the proof of the next lemma to the reader.

Lemma 4.4.3. *Arithmetic progressions have the following properties.*

(1) *If $P, Q \subseteq \mathbf{Z}$ are arithmetic progressions of moduli d and e respectively, then $P \cap Q$ is an arithmetic progression of modulus* $\mathrm{lcm}(d, e)$.
(2) *If $P \subseteq \mathbf{Z}$ is an arithmetic progression, then $\mathbf{Z} \setminus P$ is a finite union of arithmetic progressions.*
(3) *Let $\mathcal{P}$ be the collection of all finite unions of arithmetic progressions. Then $\mathcal{P}$ contains with any two sets X, Y also $X \cup Y$, $X \cap Y$, $X \setminus Y$.*

Corollary 4.4.4. *Let $S \subseteq \mathbf{Z}$. Then*

$$S \text{ is definable in } \tilde{\mathbf{Z}} \iff S \text{ is a finite union of arithmetic progressions.}$$

Proof. ($\Leftarrow$) It suffices to show that each arithmetic progression is definable in $\tilde{\mathbf{Z}}$; this is straightforward and left to the reader. ($\Rightarrow$) By QE and Lemma 4.4.3 it suffices to show that each atomic L_{PrA}-formula $\varphi(x)$ defines in $\tilde{\mathbf{Z}}$ a finite union of arithmetic progressions. Every atomic formula $\varphi(x)$ is of the form $t_1(x) < t_2(x)$ or $t_1(x) = t_2(x)$ or $P_d(t(x))$, where $t_1(x)$, $t_2(x)$ and $t(x)$ are L_{PrA}-terms. The first two kinds reduce to $t(x) > 0$ and $t(x) = 0$ respectively (by subtraction). It follows that we may assume that $\varphi(x)$ has the form $kx + l1 > 0$, or the form $kx + l1 = 0$, or the form $P_d(kx + l1)$, where $k, l \in \mathbf{Z}$. Considering cases ($k = 0$, $k \neq 0$ and $k \equiv 0 \mod d$, and so on), we see that such a $\varphi(x)$ defines an arithmetic progression. □

Exercises.

(1) Prove that $2\mathbf{Z}$ cannot be defined in the structure $(\mathbf{Z};\ 0, 1, +, -, <)$ by a q-free formula of the language $\{0, 1, +, -, <\}$.

4.5 Skolemization and Extension by Definition

In this section L is a sublanguage of L', Σ a set of L-sentences, and Σ' a set of L'-sentences with $\Sigma \subseteq \Sigma'$.

Definition. Σ' is said to be *conservative over* Σ (or a *conservative extension of* Σ) if for every L-sentence σ

$$\Sigma' \vdash_{L'} \sigma \iff \Sigma \vdash_L \sigma.$$

Here ($\Longrightarrow$) is the significant direction, ($\Longleftarrow$) is automatic.

Remarks

(1) Suppose Σ' conservative over Σ. Then Σ is consistent if and only if Σ' is consistent.

(2) If each model of Σ has an L'-expansion to a model of Σ', then Σ' is conservative over Σ. (This follows easily from the Completeness Theorem.)

Proposition 4.5.1. *Let $\varphi(x_1, \ldots, x_n, y)$ be an L-formula. Let f_φ be an n-ary function symbol not in L, and put $L' := L \cup \{f_\varphi\}$ and*

$$\Sigma' := \Sigma \cup \{\forall x_1 \ldots \forall x_n (\exists y \varphi(x_1, \ldots, x_n, y) \to \varphi(x_1, \ldots, x_n, f_\varphi(x_1, \ldots, x_n)))\}$$

Then Σ' is conservative over Σ.

Proof. Let $\mathcal{A}$ be any model of Σ. By remark (2) it suffices to obtain an L'-expansion $\mathcal{A}'$ of $\mathcal{A}$ that makes the new axiom about f_φ true. We choose a function $f_\varphi^{\mathcal{A}'} : A^n \longrightarrow A$ as follows. For any $(a_1, \ldots, a_n) \in A^n$, if there is a $b \in A$ such that $\mathcal{A} \models \varphi(\underline{a}_1, \ldots, \underline{a}_n, \underline{b})$ then we let $f_\varphi^{\mathcal{A}'}(a_1, \ldots, a_n)$ be such an element b, and if no such b exists, we let $f_\varphi^{\mathcal{A}'}(a_1, \ldots, a_n)$ be an arbitrary element of A. Thus

$$\mathcal{A}' \models \forall x_1 \ldots \forall x_n (\exists y \varphi(x_1, \ldots, x_n, y) \to \varphi(x_1, \ldots, x_n, f_\varphi(x_1, \ldots, x_n)))$$

as desired. □

Remark. A function $f_\varphi^{\mathcal{A}'}$ as in the proof is called a *Skolem function in* $\mathcal{A}$ for the formula $\varphi(x_1, \ldots, x_n, y)$. It yields a "witness" for each relevant n-tuple.

Definition. Given an L-formula $\varphi(x_1, \dots, x_m)$, let R_φ be an m-ary relation symbol not in L, and put $L_\varphi := L \cup \{R_\varphi\}$ and $\Sigma_\varphi := \Sigma \cup \{\rho_\varphi\}$ where ρ_φ is

$$\forall x_1 \dots \forall x_m (\varphi(x_1, \dots, x_m) \leftrightarrow R_\varphi(x_1, \dots, x_m)).$$

The sentence ρ_φ is called the *defining axiom for* R_φ. We call Σ_φ an *extension of* Σ *by a definition for the relation symbol* R_φ.

Remark. Each model $\mathcal{A}$ of Σ expands uniquely to a model of Σ_φ. We denote this expansion by $\mathcal{A}_\varphi$. Every model of Σ_φ is of the form $\mathcal{A}_\varphi$ for a unique model $\mathcal{A}$ of Σ.

Proposition 4.5.2. *Let $\varphi = \varphi(x_1, \dots, x_m)$ be as above. Then we have:*
(1) *Σ_φ is conservative over Σ.*
(2) *For each L_φ-formula $\psi(y)$ where $y = (y_1, \dots, y_n)$ there is an L-formula $\psi^*(y)$, called a* translation *of $\psi(y)$, such that $\Sigma_\varphi \vdash \psi(y) \leftrightarrow \psi^*(y)$.*
(3) *Suppose $\mathcal{A} \models \Sigma$ and $S \subseteq A^m$. Then S is 0-definable in $\mathcal{A}$ if and only if S is 0-definable in $\mathcal{A}_\varphi$, and the same with* definable *instead of* 0-definable.

Proof. (1) is clear from the remark preceding the proposition, and (3) is immediate from (2). To prove (2) we observe that by the Equivalence Theorem (3.3.2) it suffices to prove it for formulas $\psi(y) = R_\varphi t_1(y) \dots t_m(y)$ where the t_i are L-terms. In this case we can take

$$\exists u_1 \dots \exists u_m (u_1 = t_1(y) \wedge \dots \wedge u_m = t_m(y) \wedge \varphi(u_1/x_1, \dots, u_n/x_n))$$

as $\psi^*(y)$ where the variables $u_1, \dots, u_m$ do not appear in φ and are not among $y_1, \dots, y_n$. $\square$

Definition. Suppose $\varphi(x, y)$ is an L-formula where $(x, y) = (x_1, \dots, x_m, y)$ is a tuple of $m + 1$ distinct variables, such that $\Sigma \vdash \forall x_1 \dots \forall x_m \exists^! y \varphi(x, y)$, where $\exists^! y \varphi(x, y)$ abbreviates $\exists y \big(\varphi(x, y) \wedge \forall z (\varphi(x, z) \to y = z)\big)$, with z a variable not occurring in φ and not among $x_1, \dots, x_m, y$. Let f_φ be an m-ary function symbol not in L and put $L' := L \cup \{f_\varphi\}$ and $\Sigma' := \Sigma \cup \{\gamma_\varphi\}$ where γ_φ is

$$\forall x_1 \dots \forall x_m \varphi(x, f_\varphi(x))$$

The sentence γ_φ is called the *defining axiom for* f_φ. We call Σ' an *extension of* Σ *by a definition of the function symbol* f_φ.

Remark. Each model $\mathcal{A}$ of Σ expands uniquely to a model of Σ'. We denote this expansion by $\mathcal{A}'$. Every model of Σ' is of the form $\mathcal{A}'$ for a unique model $\mathcal{A}$ of Σ. Proposition 4.5.2 goes through when L_φ, Σ_φ, and $\mathcal{A}$ are replaced by L', Σ', and $\mathcal{A}'$, respectively.

In the next definition we use the following notation and terminology. Let X, Y be sets, $f : X \to Y$ a map, and $S \subseteq X^n$. Then the f-image of S is the subset

$$f(S) := \{(f(x_1), \dots, f(x_n)) : (x_1, \dots, x_n) \in S\}$$

of Y^n. Also, given $k \in \mathbf{N}$, we use the bijection

$$((y_{11}, \ldots, y_{1k}), \ldots, (y_{n1}, \ldots, y_{nk})) \mapsto (y_{11}, \ldots, y_{1k}, \ldots, y_{n1}, \ldots, y_{nk})$$

from $(Y^k)^n$ to Y^{nk} to identify these two sets.

Definition. A *definition* of an L-structure $\mathcal{A}$ in a structure $\mathcal{B}$ is an injective map $\delta : A \to B^k$, with $k \in \mathbf{N}$, such that
(i) $\delta(A) \subseteq B^k$ is definable in $\mathcal{B}$.
(ii) For each m-ary $R \in L^r$ the set $\delta(R^{\mathcal{A}}) \subseteq (B^k)^m = B^{mk}$ is definable in $\mathcal{B}$.
(iii) For each n-ary $f \in L^f$ the set $\delta\big(\text{graph of } f^{\mathcal{A}}\big) \subseteq (B^k)^{n+1} = B^{(n+1)k}$ is definable in $\mathcal{B}$.

Remark. Here $\mathcal{B}$ is a structure for a language L^* that may have nothing to do with the language L. Replacing everywhere "definable" by "0-definable", we get the notion of a 0-definition of $\mathcal{A}$ in $\mathcal{B}$.

A more general way of viewing a structure $\mathcal{A}$ as in some sense living inside a structure $\mathcal{B}$ is to allow δ to be an injective map from A into B^k/E for some equivalence relation E on B^k that is definable in $\mathcal{B}$, and imposing suitable conditions. Our special case corresponds to E = equality on B^k. (We do not develop this idea here further: the right setting for it would be many-sorted structures, rather than our one-sorted structures.)

Recall that by Lagrange's "four squares" theorem we have

$$\mathbf{N} = \{a^2 + b^2 + c^2 + d^2 : a, b, c, d \in \mathbf{Z}\}.$$

It follows that the inclusion map $\mathbf{N} \to \mathbf{Z}$ is a 0-definition of $(\mathbf{N};\ 0, +, \cdot, <)$ in $(\mathbf{Z};\ 0, 1, +, -, \cdot)$. The bijection

$$a + bi \mapsto (a, b) : \mathbf{C} \to \mathbf{R}^2 \qquad (a, b \in \mathbf{R})$$

is a 0-definition of the field $(\mathbf{C};\ 0, 1, +, -, \cdot)$ of complex numbers in the field $(\mathbf{R};\ 0, 1, +, -, \cdot)$ of real numbers.

There is no definition of the field of real numbers in the field of complex numbers, but the proof of this fact is somewhat beyond the scope of these notes. (A special case says that $\mathbf{R}$, considered as a subset of $\mathbf{C}$, is not definable in the field of complex numbers; this follows easily from the fact that ACF admits QE, see exercise 1.) Indeed, the only fields that can be defined in the field of complex numbers are finite fields and fields isomorphic to the field of complex numbers.

Proposition 4.5.3. *Let $\delta : A \to B^k$ be a 0-definition of the L-structure $\mathcal{A}$ in the L^*-structure $\mathcal{B}$. Let $x_1, \ldots, x_n$ be distinct variables* (viewed as ranging over A)*, and let $x_{11}, \ldots, x_{1k}, \ldots, x_{n1}, \ldots, x_{nk}$ be nk distinct variables* (viewed as ranging over B)*. Then we have a map that assigns to each L-formula $\varphi(x_1, \ldots, x_n)$ an L^*-formula $\delta\varphi(x_{11}, \ldots, x_{1k}, \ldots, x_{n1}, \ldots, x_{nk})$ such that*

$$\delta(\varphi^{\mathcal{A}}) = (\delta\varphi)^{\mathcal{B}} \subseteq B^{nk}.$$

In particular, for $n = 0$ the map above assigns to each L-sentence σ an L^*-sentence $\delta\sigma$ such that $\mathcal{A} \models \sigma \Longleftrightarrow \mathcal{B} \models \delta\sigma$.

Exercises.

(1) Prove the version of Proposition 4.5.2 for an extension of Σ by a definition of a function symbol.

Chapter 5

Computability, Decidability, and Incompleteness

In this chapter we prove Gödel's famous Incompleteness Theorem. Consider the structure $\mathfrak{N} := (\mathbf{N};\ 0, S, +, \cdot, <)$, where $S : \mathbf{N} \to \mathbf{N}$ is the successor function. A simple form of the incompleteness theorem is as follows.

Let Σ be a computable set of sentences in the language of $\mathfrak{N}$ and true in $\mathfrak{N}$. Then there exists a sentence σ in that language such that $\mathfrak{N} \models \sigma$, but $\Sigma \nvdash \sigma$.

In other words, no computable set of axioms in the language of $\mathfrak{N}$ and true in $\mathfrak{N}$ can be complete, hence the name *Incompleteness Theorem*. The only unexplained terminology here is "computable." Intuitively, "Σ is computable" means that there is an algorithm to recognize whether any given sentence in the language of $\mathfrak{N}$ belongs to Σ. (It seems reasonable to require this of an axiom system for $\mathfrak{N}$.) Thus we begin this chapter with developing the notion of *computability*. The interest of this notion is tied to the Church-Turing Thesis as explained in Section 5.2, and goes far beyond incompleteness. For example, computability plays a role in combinatorial group theory (Higman's Theorem) and in certain diophantine questions (Hilbert's 10th problem), not to mention its role in the ideological underpinnings of computer science.

5.1 Computable Functions

First some notation. We let $\mu x(..x..)$ denote the least $x \in \mathbf{N}$ for which $..x..$ holds. Here $..x..$ is some condition on natural numbers x. For example $\mu x(x^2 > 7) = 3$. We will only use this notation when the meaning of $..x..$ is clear, and the set $\{x \in \mathbf{N} : ..x..\}$ is non-empty. For $a \in \mathbf{N}$ we also let $\mu x_{<a}(..x..)$ be the least $x < a$ in $\mathbf{N}$ such that $..x..$ holds if there is such an x, and if there is no such x we put $\mu x_{<a}(..x..) := a$. For example, $\mu x_{<4}(x^2 > 3) = 2$ and $\mu x_{<2}(x > 5) = 2$.

Definition. For $R \subseteq \mathbf{N}^n$, we define $\chi_R : \mathbf{N}^n \to \mathbf{N}$ by $\chi_R(a) = \begin{cases} 1 & \text{if } a \in R, \\ 0 & \text{if } a \notin R. \end{cases}$

Think of such R as an n-ary relation on $\mathbf{N}$. We call χ_R the characteristic function of R, and often write $R(a_1, \dots, a_n)$ instead of $(a_1, \dots, a_n) \in R$.

Example. $\chi_<(m, n) = 1$ iff $m < n$, and $\chi_<(m, n) = 0$ iff $m \geq n$.

Definition. For $i = 1, \dots, n$ we define $I_i^n : \mathbf{N}^n \to \mathbf{N}$ by $I_i^n(a_1, \dots, a_n) = a_i$. These functions are called *coordinate functions*.

Definition. The *computable functions* (or *recursive functions*) are the functions from $\mathbf{N}^n$ to $\mathbf{N}$ (for $n = 0, 1, 2, \dots$) obtained by inductively applying the following rules:

(R1) $+ : \mathbf{N}^2 \to \mathbf{N}$, $\cdot : \mathbf{N}^2 \to \mathbf{N}$, $\chi_\leq : \mathbf{N}^2 \to \mathbf{N}$, and the coordinate functions I_i^n (for each n and $i = 1, \dots, n$) are computable.

(R2) If $G : \mathbf{N}^k \to \mathbf{N}$ is computable and $H_1, \dots, H_k : \mathbf{N}^t \to \mathbf{N}$ are computable, then so is the function $F = G(H_1, \dots, H_k) : \mathbf{N}^t \to \mathbf{N}$ defined by

$$F(a) = G(H_1(a), \dots, H_k(a)).$$

(R3) If $G : \mathbf{N}^{n+1} \to \mathbf{N}$ is computable, and for all $a \in \mathbf{N}^n$ there exists $x \in \mathbf{N}$ such that $G(a, x) = 0$, then the function $F : \mathbf{N}^n \to \mathbf{N}$ given by

$$F(a) = \mu x (G(a, x) = 0)$$

is computable.

A relation $R \subseteq \mathbf{N}^n$ is said to be *computable* (or *recursive*) if its characteristic function $\chi_R : \mathbf{N}^n \longrightarrow \mathbf{N}$ is computable.

Example. If $F : \mathbf{N}^3 \to \mathbf{N}$ and $G : \mathbf{N}^2 \to \mathbf{N}$ are computable, then so is the function $H : \mathbf{N}^4 \to \mathbf{N}$ defined by $H(x_1, x_2, x_3, x_4) = F(G(x_1, x_4), x_2, x_4)$. This follows from (R2) by noting that $H(x) = F(G(I_1^4(x), I_4^4(x)), I_2^4(x), I_4^4(x))$ where $x = (x_1, x_2, x_3, x_4)$. We shall use this device from now on in many proofs, but only tacitly. (The reader should of course notice when we do so.)

From (R1), (R2) and (R3) we derive further rules for obtaining computable functions. This is mostly an exercise in programming.

Lemma 5.1.1. *Let $H_1, \dots, H_k : \mathbf{N}^n \to \mathbf{N}$ and $R \subseteq \mathbf{N}^k$ be computable. Then $R(H_1, \dots, H_k) \subseteq \mathbf{N}^n$ is computable, where for $a \in \mathbf{N}^n$ we put*

$$R(H_1, \dots, H_k)(a) \Longleftrightarrow R(H_1(a), \dots, H_k(a)).$$

Proof. Observe that $\chi_{R(H_1, \dots, H_k)} = \chi_R(H_1, \dots, H_k)$. Now apply (R2). $\square$

Lemma 5.1.2. *The functions $\chi_\geq$ and $\chi_=$ on $\mathbf{N}^2$ are computable, as is the constant function $c_0^n : \mathbf{N}^n \to \mathbf{N}$ where $c_0^n(a) = 0$ for all n and $a \in \mathbf{N}^n$.*

Proof. The function $\chi_{\geq}$ is computable because

$$\chi_{\geq}(m,n) = \chi_{\leq}(n,m) = \chi_{\leq}(I_2^2(m,n), I_1^2(m,n))$$

which enables us to apply (R1) and (R2). Similarly, $\chi_{=}$ is computable:

$$\chi_{=}(m,n) = \chi_{\leq}(m,n) \cdot \chi_{\geq}(m,n).$$

To handle c_0^n we observe

$$c_0^n(a) = \mu x(I_{n+1}^{n+1}(a,x) = 0).$$

$\square$

For $k \in \mathbf{N}$ we define the constant function $c_k^n : \mathbf{N}^n \to \mathbf{N}$ by $c_k^n(a) = k$.

Lemma 5.1.3. *Every constant function c_k^n is computable.*

Proof. This is true for $k = 0$ (and all n) by Lemma 5.1.2. Next, observe that

$$c_{k+1}^n(a) = \mu x(c_k^n(a) < x) = \mu x\big(\chi_{\geq}(c_k^{n+1}(a,x), I_{n+1}^{n+1}(a,x)) = 0\big)$$

for $a \in \mathbf{N}^n$, so the general result follows by induction on k. $\square$

Let P, Q be n-ary relations on $\mathbf{N}$. Then we can form the n-ary relations $\neg P := \mathbf{N}^n \smallsetminus P$, $P \vee Q := P \cup Q$, $P \wedge Q := P \cap Q$, $P \to Q := (\neg P) \vee Q$ and $P \leftrightarrow Q := (P \to Q) \wedge (Q \to P)$ on $\mathbf{N}$.

Lemma 5.1.4. *Suppose P, Q are computable. Then $\neg P$, $P \vee Q$, $P \wedge Q$, $P \to Q$ and $P \leftrightarrow Q$ are also computable.*

Proof. Let $a \in \mathbf{N}^n$. Then $\neg P(a)$ iff $\chi_P(a) = 0$ iff $\chi_P(a) = c_0^n(a)$, so $\chi_{\neg P}(a) = \chi_{=}(\chi_P(a), c_0^n(a))$. Hence $\neg P$ is computable by Lemma 5.1.1. Next, $P \wedge Q$ is computable since $\chi_{P\cap Q} = \chi_P \cdot \chi_Q$. By De Morgan's Law, $P \vee Q = \neg(\neg P \wedge \neg Q)$. Thus $P \vee Q$ is computable. The rest is clear. $\square$

Lemma 5.1.5. *The binary relations $<, \leq, =, >, \geq, \neq$ on $\mathbf{N}$ are computable.*

Proof. The relations $\geq$, $\leq$ and $=$ have already been taken care of by Lemma 5.1.2 and (R1). The remaining relations are complements of these three, so by Lemma 5.1.4 they are also computable. $\square$

Lemma 5.1.6. (Definition by Cases) *Let $R_1, \dots, R_k \subseteq \mathbf{N}^n$ be computable such that for each $a \in \mathbf{N}^n$ exactly one of $R_1(a), \dots, R_k(a)$ holds, and suppose that $G_1, \dots, G_k : \mathbf{N}^n \to \mathbf{N}$ are computable. Then $G : \mathbf{N}^n \to \mathbf{N}$ given by*

$$G(a) = \begin{cases} G_1(a) & \text{if } R_1(a) \\ \vdots & \vdots \\ G_k(a) & \text{if } R_k(a) \end{cases}$$

is computable.

Proof. This follows from $G = G_1 \cdot \chi_{R_1} + \cdots + G_k \cdot \chi_{R_k}$. □

Lemma 5.1.7. (Definition by Cases) *Let $R_1, \ldots, R_k \subseteq \mathbf{N}^n$ be computable such that for each $a \in \mathbf{N}^n$ exactly one of $R_1(a), \ldots, R_k(a)$ holds. Let $P_1, \ldots, P_k \subseteq \mathbf{N}^n$ be computable. Then the relation $P \subseteq \mathbf{N}^n$ defined by*

$$P(a) \iff \begin{cases} P_1(a) & \text{if } R_1(a) \\ \vdots & \vdots \\ P_k(a) & \text{if } R_k(a) \end{cases}$$

is computable.

Proof. Use that $P = (P_1 \wedge R_1) \vee \cdots \vee (P_k \wedge R_k)$. □

Lemma 5.1.8. *Let $R \subseteq \mathbf{N}^{n+1}$ be computable such that for all $a \in \mathbf{N}^n$ there exists $x \in \mathbf{N}$ with $(a, x) \in R$. Then the function $F : \mathbf{N}^n \to \mathbf{N}$ given by*

$$F(a) = \mu x R(a, x)$$

is computable.

Proof. Note that

$$F(a) = \mu x(\chi_R(a, x) \neq 0) = \mu x(\chi_=(\chi_R(a, x), c_0^{n+1}(a, x)) = 0)$$

and apply (R3). □

Here is a nice consequence of 5.1.5 and 5.1.8.

Lemma 5.1.9. *Let $F : \mathbf{N}^n \to \mathbf{N}$. Then F is computable if and only if its graph (a subset of $\mathbf{N}^{n+1}$) is computable.*

Proof. Let $R \subseteq \mathbf{N}^{n+1}$ be the graph of F. Then for all $a \in \mathbf{N}^n$ and $b \in \mathbf{N}$,

$$R(a, b) \iff F(a) = b, \qquad F(a) = \mu x R(a, x),$$

from which the lemma follows immediately. □

Lemma 5.1.10. *If $R \subseteq \mathbf{N}^{n+1}$ is computable, then the function $F_R : \mathbf{N}^{n+1} \to \mathbf{N}$ defined by $F_R(a, y) = \mu x_{<y} R(a, x)$ is computable.*

Proof. Use that $F_R(a, y) = \mu x(R(a, x) \text{ or } x = y)$. □

Some notation: below we use the bold symbol $\boldsymbol{\exists}$ as shorthand for "there exists a natural number"; likewise, we use the bold symbol $\boldsymbol{\forall}$ to abbreviate "for all natural numbers." These abbreviation symbols should not be confused with the logical symbols $\exists$ and $\forall$.

Lemma 5.1.11. *Suppose $R \subseteq \mathbf{N}^{n+1}$ is computable. Let $P, Q \subseteq \mathbf{N}^{n+1}$ be the relations defined by*

$$\begin{aligned} P(a,y) &\iff \exists x_{<y}\, R(a,x) \\ Q(a,y) &\iff \forall x_{<y}\, R(a,x), \end{aligned}$$

for $(a,y) = (a_1,\dots,a_n,y) \in \mathbf{N}^{n+1}$. Then P and Q are computable.

Proof. Using the notation and results from Lemma 5.1.10 we note that $P(a,y)$ iff $F_R(a,y) < y$. Hence $\chi_P(a,y) = \chi_<(F_R(a,y),y)$. For Q, note that $\neg Q(a,y)$ iff $\exists x_{<y} \neg R(a,x)$. □

Lemma 5.1.12. *The function $\dot{-} : \mathbf{N}^2 \to \mathbf{N}$ defined by $a \dot{-} b = \begin{cases} a-b & \text{if } a \geq b, \\ 0 & \text{if } a < b \end{cases}$ is computable.*

Proof. Use that $a \dot{-} b = \mu x(b + x = a \text{ or } a < b)$. □

The results above imply easily that many familiar functions are computable. But is the exponential function $n \mapsto 2^n$ computable? It certainly is in the intuitive sense: we know how to compute (in principle) its value at any given argument. It is not that obvious from what we have proved so far that it is computable in our precise sense. We now develop some coding tricks due to Gödel that enable us to prove routinely that functions like 2^x are computable according to our definition of "computable function".

Definition. Define the function Pair : $\mathbf{N}^2 \to \mathbf{N}$ by

$$\mathrm{Pair}(x,y) := \frac{(x+y)(x+y+1)}{2} \;+\; x$$

We call Pair the *pairing function*.

Lemma 5.1.13. *The function* Pair *is bijective and computable.*

Proof. Exercise. □

Definition. Since Pair is a bijection we can define functions

$$\mathrm{Left}, \mathrm{Right} : \mathbf{N} \to \mathbf{N}$$

by

$$\mathrm{Pair}(x,y) = a \iff \mathrm{Left}(a) = x \text{ and } \mathrm{Right}(a) = y.$$

The reader should check that $\mathrm{Left}(a), \mathrm{Right}(a) \leq a$ for $a \in \mathbf{N}$, and $\mathrm{Left}(a) < a$ if $0 < a \in \mathbf{N}$.

Lemma 5.1.14. *The functions* Left *and* Right *are computable.*

Proof. Use 5.1.9 in combination with

$$\mathrm{Left}(a) = \mu x\big(\exists y_{<a+1}\,\mathrm{Pair}(x,y) = a\big) \text{ and } \mathrm{Right}(a) = \mu y\big(\exists x_{<a+1}\,\mathrm{Pair}(x,y) = a\big).$$

□

For $a, b, c \in \mathbf{Z}$ we have (by definition): $a \equiv b \mod c \iff a - b \in c\mathbf{Z}$.

Lemma 5.1.15. *The ternary relation $a \equiv b \mod c$ on $\mathbf{N}$ is computable.*

Proof. Use that for $a, b, c \in \mathbf{N}$ we have
$a \equiv b \mod c \iff \big(\exists x_{<a+1}\, a = x \cdot c + b \text{ or } \exists x_{<b+1}\, b = x \cdot c + a\big)$. □

We can now introduce Gödel's function $\beta : \mathbf{N}^2 \to \mathbf{N}$.

Definition. For $a, i \in \mathbf{N}$ we let $\beta(a,i)$ be the remainder of $\mathrm{Left}(a)$ upon division by $1 + (i+1)\,\mathrm{Right}(a)$, that is,

$$\beta(a,i) := \mu x\big(x \equiv \mathrm{Left}(a) \mod 1 + (i+1)\,\mathrm{Right}(a)\big).$$

Proposition 5.1.16. *The function β is computable, $\beta(a,i) \le a \dot{-} 1$ for all $a, i \in \mathbf{N}$. For any sequence $(a_0, \ldots, a_{n-1})$ of natural numbers there exists $a \in \mathbf{N}$ such that $\beta(a,i) = a_i$ for $i < n$.*

Proof. The computability of β is clear from earlier results. We have $\beta(a,i) \le \mathrm{Left}(a) \le a \dot{-} 1$.

Let $a_0, \ldots, a_{n-1}$ be natural numbers. Then we take an $N \in \mathbf{N}$ such that $a_i \le N$ for all $i < n$ and N is a multiple of every prime number less than n. We claim that then $1 + N,\ 1 + 2N, \ldots, 1 + nN$ are relatively prime. To see this, suppose p is a prime number such that $p \mid 1 + iN$ and $p \mid 1 + jN$ $(1 \le i < j \le n)$; then p divides their difference $(j - i)N$, but p does not divide N, hence $p \mid j - i < n$, a contradiction.

By the Chinese Remainder Theorem there exists an M such that

$$\begin{aligned} M &\equiv a_0 \mod 1 + N \\ M &\equiv a_1 \mod 1 + 2N \\ &\vdots \\ M &\equiv a_{n-1} \mod 1 + nN. \end{aligned}$$

Put $a := \mathrm{Pair}(M, N)$; then $\mathrm{Left}(a) = M$ and $\mathrm{Right}(a) = N$, and thus $\beta(a,i) = a_i$ as required. □

Remark. Proposition 5.1.16 shows that we can use β to encode a sequence of numbers $a_0, \ldots, a_{n-1}$ in terms of a single number a. We use this as follows to show that the function $n \mapsto 2^n$ is computable.

If $a_0, \ldots, a_n$ are natural numbers such that $a_0 = 1$, and $a_{i+1} = 2a_i$ for all $i < n$, then necessarily $a_n = 2^n$. Hence by Proposition 5.1.16 we have $\beta(a,n) = 2^n$ where

$$a := \mu x(\beta(x,0) = 1 \text{ and } \forall i_{<n} \beta(x,i+1) = 2\beta(x,i)),$$

that is,

$$2^n = \beta(a,n) = \beta(\mu x(\beta(x,0) = 1 \text{ and } \forall i_{<n} \beta(x, i+1) = 2\beta(x,i)), n)$$

It follows that $n \mapsto 2^n$ is computable.

The above suggests a general method, which we develop next. To each sequence $(a_1, \dots, a_n)$ of natural numbers we assign a *sequence number*, denoted $\langle a_1, \dots, a_n \rangle$, and defined to be the least natural number a such that $\beta(a, 0) = n$ (the length of the sequence) and $\beta(a, i) = a_i$ for $i = 1, \dots, n$. For $n = 0$ this gives $\langle\rangle = 0$, where $\langle\rangle$ is the sequence number of the empty sequence. We define the *length function* $\text{lh} : \mathbf{N} \longrightarrow \mathbf{N}$ by $\text{lh}(a) = \beta(a, 0)$, so lh is computable. Observe that $\text{lh}(\langle a_1, \dots, a_n \rangle) = n$.

Put $(a)_i := \beta(a, i+1)$. The function $(a, i) \mapsto (a)_i : \mathbf{N}^2 \longrightarrow \mathbf{N}$ is computable, and $(\langle a_1, \dots, a_n \rangle)_i = a_{i+1}$ for $i < n$. Finally, let $\text{Seq} \subseteq \mathbf{N}$ denote the set of sequence numbers. The set Seq is computable since

$$a \in \text{Seq} \iff \forall x_{<a} (\text{lh}(x) \neq \text{lh}(a) \text{ or } \exists i_{<\text{lh}(a)} (x)_i \neq (a)_i)$$

Lemma 5.1.17. *For any n, the function $(a_1, \dots, a_n) \mapsto \langle a_1, \dots, a_n \rangle : \mathbf{N}^n \to \mathbf{N}$ is computable, and $a_i < \langle a_1, \dots, a_n \rangle$ for $(a_1, \dots, a_n) \in \mathbf{N}^n$ and $i = 1, \dots, n$.*

Proof. Use $\langle a_1, \dots, a_n \rangle = \mu a(\beta(a, 0) = n, \beta(a, 1) = a_1, \dots, \beta(a, n) = a_n)$, and apply Lemmas 5.1.8, 5.1.4 and 5.1.16. ☐

Lemma 5.1.18.

(1) *The function* $\text{In} : \mathbf{N}^2 \to \mathbf{N}$ *defined by*

$$\text{In}(a, i) = \mu x(\text{lh}(x) = i \ \textit{ and} \forall j_{<i} (x)_j = (a)_j)$$

is computable and $\text{In}(\langle a_1, \dots, a_n \rangle, i) = \langle a_1, \dots, a_i \rangle$ *for all* $a_1, \dots, a_n \in \mathbf{N}$ *and* $i \leq n$.

(2) *The function* $* : \mathbf{N}^2 \to \mathbf{N}$ *defined by*

$$a * b = \mu x(\text{lh}(x) = \text{lh}(a) + \text{lh}(b) \ \textit{and}\ \forall i_{<\text{lh}(a)} (x)_i = (a)_i \\ \textit{and}\ \forall j_{<\text{lh}(b)} (x)_{\text{lh}(a)+j} = (b)_j)$$

is computable and $\langle a_1, \dots, a_m \rangle * \langle b_1, \dots, b_n \rangle = \langle a_1, \dots, a_m, b_1, \dots, b_n \rangle$ *for all* $a_1, \dots, a_m, b_1, \dots, b_n \in \mathbf{N}$.

Definition. For $F : \mathbf{N}^{n+1} \to \mathbf{N}$, let $\bar{F} : \mathbf{N}^{n+1} \to \mathbf{N}$ be given by

$$\bar{F}(a, b) = \langle F(a, 0), \dots, F(a, b-1) \rangle \qquad (a \in \mathbf{N}^n).$$

Note that $\bar{F}(a, 0) = \langle\rangle = 0$.

Lemma 5.1.19. *Let $F : \mathbf{N}^{n+1} \to \mathbf{N}$. Then F is computable if and only if $\bar{F}$ is computable.*

Proof. Suppose F is computable. Then $\bar{F}$ is computable since

$$\bar{F}(a,b) = \mu x(\mathrm{lh}(x) = b \text{ and } \forall i_{<b}\, (x)_i = F(a,i)).$$

In the other direction, suppose $\bar{F}$ is computable. Then F is computable since $F(a,b) = (\bar{F}(a,b+1))_b$. □

Given $G : \mathbf{N}^{n+2} \to \mathbf{N}$ there is a unique function $F : \mathbf{N}^{n+1} \to \mathbf{N}$ such that

$$F(a,b) = G(a,b,\bar{F}(a,b)) \qquad (a \in \mathbf{N}^n).$$

This will be clear if we express the requirement on F as follows:

$$F(a,0) = G(a,0,0), \qquad F(a,b+1) = G(a,b+1,\langle F(a,0),\dots,F(a,b)\rangle).$$

The next result is important because it allows us to introduce computable functions by recursion on its values at smaller arguments.

Proposition 5.1.20. *Let G and F be as above and suppose G is computable. Then F is computable.*

Proof. Note that

$$\bar{F}(a,b) = \mu x(\mathrm{Seq}(x) \text{ and } \mathrm{lh}(x) = b \text{ and } \forall i_{<b}(x)_i = G(a,i,\mathrm{In}(x,i)))$$

for all $a \in \mathbf{N}^n$ and $b \in \mathbf{N}$. It follows that $\bar{F}$ is computable, and thus by the previous lemma F is computable. □

Definition. Let $A : \mathbf{N}^n \to \mathbf{N}$ and $B : \mathbf{N}^{n+2} \to \mathbf{N}$ be given. Let a range over $\mathbf{N}^n$, and define the function $F : \mathbf{N}^{n+1} \to \mathbf{N}$ by

$$\begin{aligned} F(a,0) &= A(a), \\ F(a,b+1) &= B(a,b,F(a,b)). \end{aligned}$$

We say that F is obtained from A and B by *primitive recursion*.

Proposition 5.1.21. *Suppose A, B, and F are as above, and A and B are computable. Then F is computable.*

Proof. Define $G : \mathbf{N}^{n+2} \to \mathbf{N}$ by

$$G(a,b,c) = \begin{cases} A(a) & \text{if } c = 0, \\ B(a, b \dot{-} 1, (c)_{b \dot{-} 1}) & \text{if } c > 0. \end{cases}$$

Clearly, G is computable. We claim that

$$F(a,b) = G(a,b,\bar{F}(a,b)).$$

This claim yields the computability of F, by Proposition 5.1.20. We have $F(a,0) = A(a) = G(a,0,0) = G(a,0,\bar{F}(a,0))$ and

$$F(a,b+1) = B(a,b,F(a,b)) = B(a,b,(\bar{F}(a,b+1))_b) = G(a,b+1,\bar{F}(a,b+1)).$$

The claim follows. □

Proposition 5.1.20 will be applied over and over again in the later section on Gödel numbering, but in combination with definitions by cases. As a simple example of such an application, let $G : \mathbf{N} \to \mathbf{N}$ and $H : \mathbf{N}^2 \to \mathbf{N}$ be computable. There is clearly a unique function $F : \mathbf{N}^2 \to \mathbf{N}$ such that for all $a, b \in \mathbf{N}$

$$F(a,b) = \begin{cases} F(a, G(b)) & \text{if } G(b) < b, \\ H(a,b) & \text{otherwise.} \end{cases}$$

In particular $F(a,0) = H(a,0)$. We claim that F is computable.

According to Proposition 5.1.20 this claim will follow if we can specify a computable function $K : \mathbf{N}^3 \to \mathbf{N}$ such that $F(a,b) = K(a,b,\bar{F}(a,b))$ for all $a, b \in \mathbf{N}$. Such a function K is given by

$$K(a,b,c) = \begin{cases} (c)_{G(b)} & \text{if } G(b) < b, \\ H(a,b) & \text{otherwise.} \end{cases}$$

Exercises.

(1) The set of prime numbers is computable.

(2) The Fibonacci numbers are the natural numbers F_n defined recursively by $F_0 = 0$, $F_1 = 1$, and $F_{n+2} = F_{n+1} + F_n$. The function $n \mapsto F_n : \mathbf{N} \to \mathbf{N}$ is computable.

(3) If $f_1, \dots, f_n : \mathbf{N}^m \to \mathbf{N}$ are computable and $X \subseteq \mathbf{N}^n$ is computable, then $f^{-1}(X) \subseteq \mathbf{N}^m$ is computable, where $f := (f_1, \dots, f_n) : \mathbf{N}^m \to \mathbf{N}^n$.

(4) If $f : \mathbf{N} \to \mathbf{N}$ is computable and surjective, then there is a computable function $g : \mathbf{N} \to \mathbf{N}$ such that $f \circ g = \mathrm{id}_{\mathbf{N}}$.

(5) If $f : \mathbf{N} \to \mathbf{N}$ is computable and strictly increasing, then $f(\mathbf{N}) \subseteq \mathbf{N}$ is computable.

(6) All computable functions and relations are definable in $\mathfrak{N}$.

(7) Let $F : \mathbf{N}^n \to \mathbf{N}$, and define

$$\langle F \rangle : \mathbf{N} \to \mathbf{N}, \qquad \langle F \rangle(a) := F\big((a)_0, \dots, (a)_{n-1}\big).$$

Note that then $F(a_1, \dots, a_n) = \langle F \rangle(\langle a_1, \dots, a_n \rangle)$ for all $a_1, \dots, a_n \in \mathbf{N}$, and show that F is computable iff $\langle F \rangle$ is computable. (Hence n-variable computability reduces to 1-variable computability.)

Let $\mathcal{F}$ be a collection of functions $F : \mathbf{N}^m \to \mathbf{N}$ for various m. We say that $\mathcal{F}$ is *closed under composition* if for all $G : \mathbf{N}^k \to \mathbf{N}$ in $\mathcal{F}$ and all $H_1, \dots, H_k : \mathbf{N}^t \to \mathbf{N}$ in $\mathcal{F}$, the function $F = G(H_1, \dots, H_k) : \mathbf{N}^t \to \mathbf{N}$ is in $\mathcal{F}$. We say that $\mathcal{F}$ is *closed under minimalization* if for every $G : \mathbf{N}^{n+1} \to \mathbf{N}$ in $\mathcal{F}$ such that for all $a \in \mathbf{N}^n$ there exists $x \in \mathbf{N}$ with $G(a,x) = 0$, the function $F : \mathbf{N}^n \to \mathbf{N}$ given by $F(a) = \mu x(G(a,x) = 0)$ is in $\mathcal{F}$. We say that a relation $R \subseteq \mathbf{N}^n$ is *in* $\mathcal{F}$ if its characteristic function χ_R is in $\mathcal{F}$.

(8) Suppose $\mathcal{F}$ contains the functions mentioned in (R1), and is closed under composition and minimalization. Show that all lemmas and propositions of this Section go through with *computable* replaced by *in* $\mathcal{F}$.

5.2 The Church-Turing Thesis

The computable functions as defined in the last section are also computable in the informal sense that for each such function $F : \mathbf{N}^n \to \mathbf{N}$ there is an algorithm that on any input $a \in \mathbf{N}^n$ stops after a finite number of steps and produces an output $F(a)$. An *algorithm* is given by a finite list of instructions, a computer program, say. These instructions should be *deterministic* (leave nothing to chance or choice). We deliberately neglect physical constraints of space and time: imagine that the program that implements the algorithm has unlimited access to time and memory to do its work on any given input.

Let us write "calculable" for this intuitive, informal, idealized notion of computable. The **Church-Turing Thesis** asserts

each calculable function $F : \mathbf{N} \to \mathbf{N}$ *is computable.*

The corresponding assertion for functions $\mathbf{N}^n \to \mathbf{N}$ follows, because the result of Exercise 7 is clearly also valid for "calculable" instead of "computable." Call a set $P \subseteq \mathbf{N}$ *calculable* if its characteristic function is calculable.

While the Church-Turing Thesis is not a precise mathematical statement, it is an important guiding principle, and has never failed in practice: any function that any competent person has ever recognized as being calculable, has turned out to be computable, and the informal grounds for calculability have always translated routinely into an actual proof of computability. Here is a heuristic (informal) argument that might make the Thesis plausible.

Let an algorithm be given for computing $F : \mathbf{N} \to \mathbf{N}$. We can assume that on any input $a \in \mathbf{N}$ this algorithm consists of a finite sequence of steps, numbered from 0 to n, say, where at each step i it produces a natural number a_i, with $a_0 = a$ as starting number. It stops after step n with $a_n = F(a)$. We assume that for each $i < n$ the number a_{i+1} is calculated by some fixed procedure from the earlier numbers $a_0, \dots, a_i$, that is, we have a calculable function $G : \mathbf{N} \to \mathbf{N}$ such that $a_{i+1} = G(\langle a_0, \dots, a_i \rangle)$ for all $i < n$. The algorithm should also tell us when to stop, that is, we should have a calculable $P \subseteq \mathbf{N}$ such that $\neg P(\langle a_0, \dots, a_i \rangle)$ for $i < n$ and $P(\langle a_0, \dots, a_n \rangle)$. Since G and P describe only single steps in the algorithm for F it is reasonable to assume that they at least are computable. Once this is agreed to, one can show easily that F is computable as well, see the exercise below.

A skeptical reader may find this argument dubious, but Turing gave in 1936 a compelling informal analysis of what functions $F : \mathbf{N} \to \mathbf{N}$ are calculable in principle, and this has led to general acceptance of the Thesis. In addition, various alternative formalizations of the informal notion of calculable function have been proposed, using various kinds of machines, formal systems, and so on. They all have turned out to be equivalent in the sense of defining the same class of functions on $\mathbf{N}$, namely the computable functions.

The above is only a rather narrow version of the Church-Turing Thesis, but it suffices for our purpose. There are various refinements and more ambitious versions. Also, our Church-Turing Thesis does not characterize mathematically the intuitive notion of *algorithm*, only the notion *function computable by an* *algorithm* (producing for each input from $\mathbf{N}$ an output in $\mathbf{N}$).

Exercises.

(1) Let $G : \mathbf{N} \to \mathbf{N}$ and $P \subseteq \mathbf{N}$ be given. Then there is for each $a \in \mathbf{N}$ at most one finite sequence $a_0, \ldots, a_n$ of natural numbers such that $a_0 = a$, for all $i < n$ we have $a_{i+1} = G(\langle a_0, \ldots, a_i \rangle)$ and $\neg P(\langle a_0, \ldots, a_i \rangle)$, and $P(\langle a_0, \ldots, a_n \rangle)$. Suppose that for each $a \in \mathbf{N}$ there is such a finite sequence $a_0, \ldots, a_n$, and put $F(a) := a_n$, thus defining a function $F : \mathbf{N} \to \mathbf{N}$. Show that if G and P are computable, so is F.

5.3 Primitive Recursive Functions

This section is not really needed in the rest of this chapter, but it may throw light on some issues relating to computability. One such issue is the condition, in Rule (R3) for generating computable functions, that for all $a \in \mathbf{N}^n$ there exists $y \in \mathbf{N}$ such that $G(a, y) = 0$. This condition is not constructive: it could be satisfied for a certain G without us ever finding out. We shall now argue informally that it is impossible to generate in a fully constructive way exactly the computable functions. Such a constructive generation process would presumably enable us to enumerate effectively a sequence of algorithms $\alpha_0, \alpha_1, \alpha_2, \ldots$ such that each α_n computes a (computable) function $f_n : \mathbf{N} \to \mathbf{N}$, and such that every computable function $f : \mathbf{N} \to \mathbf{N}$ occurs in the sequence $f_0, f_1, f_2, \ldots$, possibly more than once. Now consider the function $f_{\text{diag}} : \mathbf{N} \to \mathbf{N}$ defined by

$$f_{\text{diag}}(n) = f_n(n) + 1.$$

Then f_{diag} is clearly computable in the intuitive sense, but $f_{\text{diag}} \neq f_n$ for all n, in violation of the Church-Turing Thesis.

This way of producing a new function f_{diag} from a sequence (f_n) is called *diagonalization.*[1] The same basic idea applies in other cases, and is used in a more sophisticated form in the proof of Gödel's incompleteness theorem.

Here is a class of computable functions that *can* be generated constructively: The *primitive recursive functions* are the functions $f : \mathbf{N}^n \to \mathbf{N}$ obtained inductively as follows:

(PR1) The nullary function $\mathbf{N}^0 \to \mathbf{N}$ with value 0, the unary successor function S, and all coordinate functions I_i^n are primitive recursive.

(PR2) If $G : \mathbf{N}^k \to \mathbf{N}$ is primitive recursive and $H_1, \ldots, H_k : \mathbf{N}^t \to \mathbf{N}$ are primitive recursive, then $G(H_1, \ldots, H_k)$ is primitive recursive.

(PR3) If $F : \mathbf{N}^{n+1} \to \mathbf{N}$ is obtained by primitive recursion from primitive recursive functions $G : \mathbf{N}^n \to \mathbf{N}$ and $H : \mathbf{N}^{n+2} \to \mathbf{N}$, then F is primitive recursive.

[1] Perhaps *antidiagonalization* would be a more appropriate term.

A relation $R \subseteq \mathbf{N}^n$ is said to be primitive recursive if its characteristic function χ_R is primitive recursive. As the next two lemmas show, the computable functions that one ordinarily meets with are primitive recursive. In the rest of this section x ranges over $\mathbf{N}^m$ with m depending on the context, and y over $\mathbf{N}$.

Lemma 5.3.1. *The following functions and relations are primitive recursive:*

(i) *each constant function* c^n_m;

(ii) *the binary operations* $+$, $\cdot$, *and* $(x, y) \mapsto x^y$ *on* $\mathbf{N}$;

(iii) *the predecessor function* $Pd : \mathbf{N} \to \mathbf{N}$ *given by* $Pd(x) = x \dot{-} 1$, *the unary relation* $\{x \in \mathbf{N} : x > 0\}$, *the function* $\dot{-} : \mathbf{N}^2 \to \mathbf{N}$;

(iv) *the binary relations* $\geq$, $\leq$ *and* $=$ *on* $\mathbf{N}$.

Proof. The function c^0_m is obtained from c^0_0 by applying (PR2) m times with $G = S$. Next, c^n_m is obtained by applying (PR2) with $G = c^0_m$ (with $k = 0$ and $t = n$). The functions in (ii) are obtained by the usual primitive recursions. It is also easy to write down primitive recursions for the functions in (iii), in the order they are listed. For (iv), note that $\chi_{\geq}(x, y + 1) = \chi_{>0}(x) \cdot \chi_{\geq}(Pd(x), y)$. □

Lemma 5.3.2. *With the possible exceptions of Lemmas 4.1.8 and 4.1.9, all Lemmas and Propositions in Section 4.1 go through with "computable" replaced by "primitive recursive."*

Proof. To obtain the primitive recursive version of Lemma 4.1.10, note that

$$F_R(a, 0) = 0, \quad F_R(a, y+1) = F_R(a, y) \cdot \chi_R(a, F_R(a, y)) + (y+1) \cdot \chi_{\neg R}(a, F_R(a, y)).$$

A consequence of the primitive recursive version of Lemma 4.1.10 is the following "restricted minimalization scheme" for primitive recursive functions:

if $R \subseteq \mathbf{N}^{n+1}$ *and* $H : \mathbf{N}^n \to \mathbf{N}$ *are primitive recursive, and for all* $a \in \mathbf{N}^n$ *there exists* $x < H(a)$ *such that* $R(a, x)$, *then the function* $F : \mathbf{N}^n \to \mathbf{N}$ *given by* $F(a) = \mu x R(a, x)$ *is primitive recursive.*

The primitive recursive versions of Lemmas 4.1.11–4.1.16 now follow easily. In particular, the function β is primitive recursive. Also, the proof of Lemma 4.1.16 yields:

There is a primitive recursive function $B : \mathbf{N} \to \mathbf{N}$ *such that, whenever*

$$n < N, a_0 < N, \ldots, a_{n-1} < N, \quad (n, a_0, \ldots, a_{n-1}, N \in \mathbf{N})$$

then for some $a < B(N)$ *we have* $\beta(a, i) = a_i$ *for* $i = 0, \ldots, n - 1$.

Using this fact and restricted minimalization, it follows that the unary relation Seq, the unary function lh, and the binary functions $(a, i) \mapsto (a)_i$, In and $*$ are primitive recursive.

Let a function $F : \mathbf{N}^{n+1} \to \mathbf{N}$ be given. Then $\bar{F} : \mathbf{N}^{n+1} \to \mathbf{N}$ satisfies the primitive recursion $\bar{F}(a, 0) = 0$ and $\bar{F}(a, b + 1) = \bar{F}(a, b) * \langle F(a, b) \rangle$. It follows that if F is primitive recursive, so is $\bar{F}$. The converse is obvious. Suppose also that $G : \mathbf{N}^{n+2} \to \mathbf{N}$ is primitive recursive, and $F(a, b) = G(a, b, \bar{F}(a, b))$ for all $(a, b) \in \mathbf{N}^{n+1}$; then $\bar{F}$ satisfies the primitive recursion

$$\bar{F}(A, 0) = G(a, 0, 0), \quad \bar{F}(a, b+1) = \bar{F}(a, b) * \langle G(a, b, \bar{F}(a, b))\rangle.$$

so $\bar{F}$ (and hence F) is primitive recursive. □

The Ackermann Function. By diagonalization we can produce a computable function that is not primitive recursive, but the so-called Ackermann function does more, and plays a role in several contexts. First we define inductively a sequence $A_0, A_1, A_2, \ldots$ of primitive recursive functions $A_n : \mathbf{N} \to \mathbf{N}$:

$$A_0(y) = y + 1, \qquad A_{n+1}(0) = A_n(1),$$
$$A_{n+1}(y+1) = A_n(A_{n+1}(y)).$$

Thus $A_0 = S$ and $A_{n+1} \circ A_0 = A_n \circ A_{n+1}$. One verifies easily that $A_1(y) = y+2$ and $A_2(y) = 2y + 3$ for all y. We define the Ackermann function $A : \mathbf{N}^2 \to \mathbf{N}$ by $A(n, y) := A_n(y)$.

Lemma 5.3.3. *The function A is computable, and strictly increasing in each variable. Also, for all n and x, y:*

(i) $A_n(x + y) \geq A_n(x) + y$;

(ii) $n \geq 1 \Longrightarrow A_{n+1}(y) > A_n(y) + y$;

(iii) $A_{n+1}(y) \geq A_n(y + 1)$;

(iv) $2A_n(y) < A_{n+2}(y)$;

(v) $x < y \Longrightarrow A_n(x + y) \leq A_{n+2}(y)$.

Proof. We leave it to the reader to verify that A is computable. Assume inductively that $A_0, \ldots, A_n$ are strictly increasing and $A_0(y) < A_1(y) < \cdots < A_n(y)$ for all y. Then

$$A_{n+1}(y + 1) = A_n(A_{n+1}(y)) \geq A_0(A_{n+1}(y)) > A_{n+1}(y),$$

so A_{n+1} is strictly increasing. Next we show that $A_{n+1}(y) > A_n(y)$ for all y: $A_{n+1}(0) = A_n(1)$, so $A_{n+1}(0) > A_n(0)$ and $A_{n+1}(0) > 1$, so $A_{n+1}(y) > y + 1$ for all y. Hence $A_{n+1}(y + 1) = A_n(A_{n+1}(y)) > A_n(y + 1)$.

Inequality (i) follows easily by induction on n, and a second induction on y.

For inequality (ii), we proceed again by induction on (n, y): Using $A_1(y) = y+2$ and $A_2(y) = 2y+3$, we obtain $A_2(y) > A_1(y)+y$. Let $n > 1$, and assume inductively that $A_n(y) > A_{n-1}(y) + y$. Then $A_{n+1}(0) = A_n(1) > A_n(0) + 0$, and

$$\begin{aligned} A_{n+1}(y + 1) = A_n(A_{n+1}(y)) &\geq A_n(y + 1 + A_n(y)) \\ &\geq A_n(y + 1) + A_n(y) > A_n(y + 1) + y + 1. \end{aligned}$$

In (iii) we proceed by induction on y. We have equality for $y = 0$. Assuming inductively that (iii) holds for a certain y we obtain

$$A_{n+1}(y+1) = A_n(A_{n+1}(y)) \geq A_n(A_n(y+1)) \geq A_n(y+2).$$

Note that (iv) holds for $n = 0$. For $n > 0$ we have by (i), (ii) and (iii):

$$A_n(y) + A_n(y) \leq A_n(y + A_n(y)) < A_n(A_{n+1}(y)) = A_{n+1}(y+1) \leq A_{n+2}(y).$$

Note that (v) holds for $n = 0$. Assume (v) holds for a certain n. Let $x < y+1$. We can assume inductively that if $x < y$, then $A_{n+1}(x+y) \leq A_{n+3}(y)$, and we want to show that

$$A_{n+1}(x+y+1) \leq A_{n+3}(y+1).$$

Case 1. $x = y$. Then

$$\begin{aligned} A_{n+1}(x+y+1) &= A_{n+1}(2x+1) = A_n(A_{n+1}(2x)) \\ &\leq A_{n+2}(2x) < A_{n+2}(A_{n+3}(x)) = A_{n+3}(y+1). \end{aligned}$$

Case 2. $x < y$. Then

$$A_{n+1}(x+y+1) = A_n(A_{n+1}(x+y)) \leq A_{n+2}(A_{n+3}(y)) = A_{n+3}(y+1).$$

□

Below we put $|x| := x_1 + \cdots + x_m$ for $x = (x_1, \ldots, x_m) \in \mathbf{N}^m$.

Proposition 5.3.4. *Given any primitive recursive function $F : \mathbf{N}^m \to \mathbf{N}$ there is an $n = n(F)$ such that $F(x) \leq A_n(|x|)$ for all $x \in \mathbf{N}^m$.*

Proof. Call an $n = n(F)$ with the property above a *bound for F*. The nullary constant function with value 0, the successor function S, and each coordinate function I_i^m, $(1 \leq i \leq m)$, has bound 0. Next, assume $F = G(H_1, \ldots, H_k)$ where $G : \mathbf{N}^k \to \mathbf{N}$ and $H_1, \ldots, H_k : \mathbf{N}^m \to \mathbf{N}$ are primitive recursive, and assume inductively that $n(G)$ and $n(H_1), \ldots, n(H_k)$ are bounds for G and $H_1, \ldots, H_k$. By part (iv) of the previous lemma we can take $N \in \mathbf{N}$ such that $n(G) \leq N$, and $\sum_i H_i(x) \leq A_{N+1}(|x|)$ for all x. Then

$$F(x) = G(H_1(x), \ldots, H_k(x)) \leq A_N(\sum_i H_i(x)) \leq A_N(A_{N+1}(|x|)) \leq A_{N+2}(|x|).$$

Finally, assume that $F : \mathbf{N}^{m+1} \to \mathbf{N}$ is obtained by primitive recursion from the primitive recursive functions $G : \mathbf{N}^m \to \mathbf{N}$ and $H : \mathbf{N}^{m+2} \to \mathbf{N}$, and assume inductively that $n(G)$ and $n(H)$ are bounds for G and H. Take $N \in \mathbf{N}$ such that $n(G) \leq N + 3$ and $n(H) \leq N$. We claim that $N + 3$ is a bound for F: $F(x, 0) = G(x) \leq A_{N+3}(|x|)$, and by part (v) of the lemma above,

$$\begin{aligned} F(x, y+1) = H(x, y, F(x,y)) &\leq A_N\{|x| + y + A_{N+3}(|x| + y)\} \\ &\leq A_{N+2}\{A_{N+3}(|x| + y)\} = A_{N+3}(|x| + y + 1). \end{aligned}$$

□

Consider the function $A^* : \mathbf{N} \to \mathbf{N}$ defined by $A^*(n) = A(n,n)$. Then A^* is computable, and for any primitive recursive function $F : \mathbf{N} \to \mathbf{N}$ we have $F(y) < A^*(y)$ for all $y > n(F)$, where $n(F)$ is a bound for F. In particular, A^* is not primitive recursive. Hence A is computable but not primitive recursive.

The recursion in "primitive recursion" involves only one variable; the other variables just act as parameters. The Ackermann function is defined by a recursion involving *both* variables:

$$A(0,y) = y+1, \quad A(x+1,0) = A(x,1), \quad A(x+1,y+1) = A(x, A(x+1,y)).$$

This kind of double recursion is therefore more powerful in some ways than what can be done in terms of primitive recursion and composition.

5.4 Representability

Let L be a numerical language, that is, L contains the constant symbol 0 and the unary function symbol S. We let S^n0 denote the term $S \dots S0$ in which S appears exactly n times. So S^00 is the term 0, S^10 is the term $S0$, and so on. Our key example of a numerical language is

$$L(\underline{\mathrm{N}}) := \{0, S, +, \cdot, <\} \quad \text{(the language of } \mathfrak{N}\text{)}.$$

Here $\underline{\mathrm{N}}$ is the following set of nine axioms, where we fix two distinct variables x and y for the sake of definiteness:

$\underline{\mathrm{N1}}$ $\quad \forall x\,(Sx \neq 0)$
$\underline{\mathrm{N2}}$ $\quad \forall x \forall y\,(Sx = Sy \to x = y)$
$\underline{\mathrm{N3}}$ $\quad \forall x\,(x + 0 = x)$
$\underline{\mathrm{N4}}$ $\quad \forall x \forall y\,\big(x + Sy = S(x+y)\big)$
$\underline{\mathrm{N5}}$ $\quad \forall x\,(x \cdot 0 = 0)$
$\underline{\mathrm{N6}}$ $\quad \forall x \forall y\,(x \cdot Sy = x \cdot y + x)$
$\underline{\mathrm{N7}}$ $\quad \forall x\,(x \not< 0)$
$\underline{\mathrm{N8}}$ $\quad \forall x \forall y\,(x < Sy \leftrightarrow x < y \vee x = y)$
$\underline{\mathrm{N9}}$ $\quad \forall x \forall y\,(x < y \vee x = y \vee y < x)$

These axioms are clearly true in $\mathfrak{N}$. The fact that $\underline{\mathrm{N}}$ is finite will play a role later. It is a very weak set of axioms, it doesn't even prove

$$\forall x \forall y (x + y = y + x),$$

but it is strong enough to prove numerical facts such as

$$SS0 + SSS0 \neq SSSSSS0, \quad \text{and } \forall x \big(x < SS0 \to (x = 0 \vee x = S0)\big).$$

Some models of $\underline{\mathrm{N}}$.

(1) We usually refer to $\mathfrak{N}$ as the *standard model* of $\underline{\mathrm{N}}$. Another model of $\underline{\mathrm{N}}$ is $\mathfrak{N}[x] := (\mathbf{N}[x];\ \dots)$, where $0, S, +, \cdot$ are interpreted as the zero polynomial, as the unary operation of adding 1 to a polynomial, and as addition and multiplication of polynomials in $\mathbf{N}[x]$, and where $<$ is interpreted as follows: $f(x) < g(x)$ iff $f(n) < g(n)$ for all large enough n.
(2) A more bizarre model of $\underline{\mathrm{N}}$: $(\mathbf{R}^{\geq 0};\ \dots)$ with the usual interpretations of $0, S, +, \cdot$, in particular $S(r) := r+1$, and with $<$ interpreted as the binary relation $<_{\mathbf{N}}$ on $\mathbf{R}^{\geq 0}$: $r <_{\mathbf{N}} s \Leftrightarrow (r, s \in \mathbf{N}$ and $r < s)$ or $s \notin \mathbf{N}$.

The first example shows that $\underline{\mathrm{N}} \nvdash \forall x \exists y\, (x = 2y \vee x = 2y + S0)$, since in $\mathfrak{N}[x]$ the element x is not in $2\mathfrak{N}[x] \cup 2\mathfrak{N}[x]+1$; in other words, $\underline{\mathrm{N}}$ cannot prove "every element is even or odd." In the second example $<_{\mathbf{N}}$ is not even a total order on the underlying set of the model. About the only useful fact about models of $\underline{\mathrm{N}}$ is that they all contain the so-called *standard model* $\mathfrak{N}$ in a unique way:

Lemma 5.4.1. *Suppose $\mathcal{A} \models \underline{\mathrm{N}}$. Then there is a unique homomorphism*

$$\iota : \mathfrak{N} \to \mathcal{A}.$$

This homomorphism ι is an embedding, and for all $a \in A$ and n,
(i) *if $a <^{\mathcal{A}} \iota(n)$, then $a = \iota(m)$ for some $m < n$;*
(ii) *if $a \notin \iota(\mathbf{N})$, then $\iota(n) <^{\mathcal{A}} a$.*

As to the proof, note that for any homomorphism $\iota : \mathfrak{N} \to \mathcal{A}$ and all n we must have $\iota(n) = (S^n 0)^{\mathcal{A}}$. Hence there is at most one such homomorphism. It remains to show that the map $n \mapsto (S^n 0)^{\mathcal{A}} : \mathbf{N} \to A$ is an embedding $\iota : \mathfrak{N} \to \mathcal{A}$ with properties (i) and (ii). We leave this as an exercise to the reader.

Definition. Let L be a numerical language, and Σ a set of L-sentences. A relation $R \subseteq \mathbf{N}^m$ is said to be Σ-*representable*, if there is an L-formula $\varphi(x_1, \dots, x_m)$ such that for all $(a_1, \dots, a_m) \in \mathbf{N}^m$ we have
(i) $R(a_1, \dots, a_m) \Longrightarrow \Sigma \vdash \varphi(S^{a_1}0, \dots, S^{a_m}0)$
(ii) $\neg R(a_1, \dots, a_m) \Longrightarrow \Sigma \vdash \neg\varphi(S^{a_1}0, \dots, S^{a_m}0)$
Such a $\varphi(x_1, \dots, x_m)$ is said to *represent* R *in* Σ or to Σ-represent R. Note that if $\varphi(x_1, \dots, x_m)$ Σ-represents R and Σ is consistent, then for all $(a_1, \dots, a_m) \in \mathbf{N}^m$

$$R(a_1, \dots, a_m) \Longleftrightarrow \Sigma \vdash \varphi(S^{a_1}0, \dots, S^{a_m}0),$$
$$\neg R(a_1, \dots, a_m) \Longleftrightarrow \Sigma \vdash \neg\varphi(S^{a_1}0, \dots, S^{a_m}0).$$

A function $F : \mathbf{N}^m \to \mathbf{N}$ is Σ-*representable* if there is a formula $\varphi(x_1, \dots, x_m, y)$ of L such that for all $(a_1, \dots, a_m) \in \mathbf{N}^m$ we have

$$\Sigma \vdash \varphi(S^{a_1}0, \dots, S^{a_m}0, y) \leftrightarrow y = S^{F(a_1, \dots, a_m)}0.$$

Such a $\varphi(x_1, \dots, x_m, y)$ is said to *represent* F *in* Σ or to Σ-represent F.

An L-term $t(x_1, \dots, x_m)$ is said to *represent* the function $F : \mathbf{N}^m \to \mathbf{N}$ in Σ if $\Sigma \vdash t(S^{a_1}0, \dots, S^{a_m}0) = S^{F(a)}0$ for all $a = (a_1, \dots, a_m) \in \mathbf{N}^m$. Note that then the function F is Σ-represented by the formula $t(x_1, \dots, x_m) = y$.

Proposition 5.4.2. *Let L be a numerical language, Σ a set of L-sentences such that $\Sigma \vdash S0 \neq 0$, and $R \subseteq \mathbf{N}^m$ a relation. Then*

$$R \text{ is } \Sigma\text{-representable} \iff \chi_R \text{ is } \Sigma\text{-representable.}$$

Proof. ($\Leftarrow$) Assume χ_R is Σ-representable and let $\varphi(x_1, \dots, x_m, y)$ be an L-formula Σ-representing it. We show that $\psi(x_1, \dots, x_m) := \varphi(x_1, \dots, x_m, S0)$ Σ-represents R. Let $(a_1, \dots, a_m) \in R$; then $\chi_R(a_1, \dots, a_m) = 1$. Hence

$$\Sigma \vdash \varphi(S^{a_1}0, \dots, S^{a_m}0, y) \leftrightarrow y = S0,$$

so $\Sigma \vdash \varphi(S^{a_1}0, \dots, S^{a_m}0, S0)$, that is, $\Sigma \vdash \psi(S^{a_1}0, \dots, S^{a_m}0)$.
Similarly, $(a_1, \dots, a_m) \notin R$ implies $\Sigma \vdash \neg\varphi(S^{a_1}0, \dots, S^{a_m}0, S0)$. (Here we need $\Sigma \vdash S0 \neq 0$.)
($\Rightarrow$) Conversely, assume R is Σ-representable and let $\varphi(x_1, \dots, x_m)$ be an L-formula Σ-representing it. We show that

$$\psi(x_1, \dots, x_m, y) := (\varphi(x_1, \dots, x_m) \wedge y = S0) \vee (\neg\varphi(x_1, \dots, x_m) \wedge y = 0)$$

Σ-represents χ_R.
Let $(a_1, \dots, a_m) \in R$; then $\Sigma \vdash \varphi(S^{a_1}0, \dots, S^{a_m}0)$. Hence,

$$\Sigma \vdash [(\varphi(S^{a_1}0, \dots, S^{a_m}0) \wedge y = S0) \vee (\neg\varphi(S^{a_1}0, \dots, S^{a_m}0) \wedge y = 0)] \leftrightarrow y = S0$$

i.e. $\Sigma \vdash \psi(S^{a_1}0, \dots, S^{a_m}0, y) \leftrightarrow y = S0$.
And similarly for $(a_1, \dots, a_m) \notin R$, $\Sigma \vdash \psi(S^{a_1}0, \dots, S^{a_m}0, y) \leftrightarrow y = 0$. $\square$

Lemma 5.4.3. *For each n,*

$$\underline{\mathrm{N}} \vdash x < S^{n+1}0 \leftrightarrow (x = 0 \vee \dots \vee x = S^n0).$$

Proof. By induction on n. For $n = 0$, $\underline{\mathrm{N}} \vdash x < S0 \leftrightarrow x = 0$ by axioms $\underline{\mathrm{N8}}$ and $\underline{\mathrm{N7}}$. Assume $n > 0$ and $\underline{\mathrm{N}} \vdash x < S^n0 \leftrightarrow (x = 0 \vee \dots \vee x = S^{n-1}0)$. Use axiom $\underline{\mathrm{N8}}$ to conclude that $\underline{\mathrm{N}} \vdash x < S^{n+1}0 \leftrightarrow (x = 0 \vee \dots \vee x = S^n0)$. $\square$

Theorem 5.4.4 (Representability). *Each computable function $F : \mathbf{N}^n \to \mathbf{N}$ is $\underline{\mathrm{N}}$-representable. Each computable relation $R \subseteq \mathbf{N}^m$ is $\underline{\mathrm{N}}$-representable.*

Proof. By Proposition 5.4.2 we need only consider the case of functions. We make the following three claims:

(R1)′ $+ : \mathbf{N}^2 \to \mathbf{N}$, $\cdot : \mathbf{N}^2 \to \mathbf{N}$, $\chi_{\leq} : \mathbf{N}^2 \to \mathbf{N}$, and the coordinate function I^n_i (for each n and $i = 1, \dots, n$) are $\underline{\mathrm{N}}$-representable.

(R2)′ If $G : \mathbf{N}^k \to \mathbf{N}$ and $H_1, \dots, H_k : \mathbf{N}^t \to \mathbf{N}$ are $\underline{\mathrm{N}}$-representable, then so is $F = G(H_1, \dots, H_k) : \mathbf{N}^t \to \mathbf{N}$ defined by

$$F(a) = G(H_1(a), \dots, H_k(a)).$$

(R3)′ If $G : \mathbf{N}^{n+1} \to \mathbf{N}$ is $\underline{\mathrm{N}}$-representable, and for all $a \in \mathbf{N}^n$ there exists $x \in \mathbf{N}$ such that $G(a, x) = 0$, then the function $F : \mathbf{N}^n \to \mathbf{N}$ given by

$$F(a) = \mu x(G(a, x) = 0)$$

is $\underline{\mathrm{N}}$-representable.

(R1)′ : The proof of this claim has six parts.

(i) The formula $x_1 = x_2$ represents $\{(a,b) \in \mathbf{N}^2 : a = b\}$ in $\underline{\mathrm{N}}$:
Let $a, b \in \mathbf{N}$. If $a = b$ then obviously $\underline{\mathrm{N}} \vdash S^a0 = S^b0$. Suppose that $a \neq b$. Then for every model $\mathcal{A}$ of $\underline{\mathrm{N}}$ we have $\mathcal{A} \models S^a0 \neq S^b0$, by Lemma 5.4.1 and its proof. Hence $\underline{\mathrm{N}} \vdash S^a0 \neq S^b0$.

(ii) The term $x_1 + x_2$ represents $+ : \mathbf{N}^2 \to \mathbf{N}$ in $\underline{\mathrm{N}}$:
Let $a + b = c$ where $a, b, c \in \mathbf{N}$. By Lemma 5.4.1 and its proof we have $\mathcal{A} \models S^a0 + S^b0 = S^c0$ for each model $\mathcal{A}$ of $\underline{\mathrm{N}}$. It follows that $\underline{\mathrm{N}} \vdash S^a0 + S^b0 = S^c0$.

(iii) The term $x_1 \cdot x_2$ represents $\cdot : \mathbf{N}^2 \to \mathbf{N}$ in $\underline{\mathrm{N}}$:
The proof is similar to that of (ii).

(iv) The formula $x_1 < x_2$ represents $\{(a,b) \in \mathbf{N}^2 : a < b\}$ in $\underline{\mathrm{N}}$:
The proof is similar to that of (i).

(v) $\chi_{\leq} : \mathbf{N}^2 \to \mathbf{N}$ is $\underline{\mathrm{N}}$-representable:
By (i) and (iv), the formula $x_1 < x_2 \vee x_1 = x_2$ represents the set $\{(a,b) \in \mathbf{N}^2 : a \leq b\}$ in $\underline{\mathrm{N}}$. So by Proposition 5.4.2, $\chi_{\leq} : \mathbf{N}^2 \to \mathbf{N}$ is $\underline{\mathrm{N}}$-representable.

(vi) For $n \geq 1$ and $1 \leq i \leq n$, the term $t_i^n(x_1, \ldots, x_n) := x_i$, represents the function $I_i^n : \mathbf{N}^n \to \mathbf{N}$ in $\underline{\mathrm{N}}$. This is obvious.

(R2)′ : Let $H_1, \ldots, H_k : \mathbf{N}^t \to \mathbf{N}$ be $\underline{\mathrm{N}}$-represented by $\varphi_i(x_1, \ldots, x_t, y_i)$ and let $G : \mathbf{N}^k \to \mathbf{N}$ be $\underline{\mathrm{N}}$-represented by $\psi(y_1, \ldots, y_k, z)$ where z is distinct from $x_1, \ldots, x_t$ and $y_1, \ldots, y_k$.

Claim : $F = G(H_1, \ldots, H_k)$ is $\underline{\mathrm{N}}$-represented by

$$\theta(x_1, \ldots, x_t, z) := \exists y_1 \ldots \exists y_k((\bigwedge_{i=1}^{k} \varphi_i(x_1, \ldots, x_t, y_i)) \wedge \psi(y_1, \ldots, y_k, z)).$$

Put $a = (a_1, \ldots, a_t)$ and let $c = F(a)$. We have to show that

$$\underline{\mathrm{N}} \vdash \theta(S^a0, z) \leftrightarrow z = S^c0$$

where $S^a0 := (S^{a_1}0, \ldots, S^{a_t}0)$. Let $b_i = H_i(a)$ and put $b = (b_1, \ldots, b_k)$. Then $F(a) = G(b) = c$. Therefore, $\underline{\mathrm{N}} \vdash \psi(S^b0, z) \leftrightarrow z = S^c0$ and

$$\underline{\mathrm{N}} \vdash \varphi_i(S^a0, y_i) \leftrightarrow y_i = S^{b_i}0, \qquad (i = 1, \ldots, k)$$

Argue in models to conclude : $\underline{\mathrm{N}} \vdash \theta(S^a0, z) \leftrightarrow z = S^c0$.

$(R3)'$: Let $G : \mathbf{N}^{n+1} \to \mathbf{N}$ be such that for all $a \in \mathbf{N}^n$ there exists $b \in \mathbf{N}$ with $G(a,b) = 0$. Define $F : \mathbf{N}^n \to \mathbf{N}$ by $F(a) = \mu b(G(a,b) = 0)$. Suppose that G is $\underline{\mathrm{N}}$-represented by $\varphi(x_1, \ldots, x_n, y, z)$. We claim that the formula

$$\psi(x_1, \ldots, x_n, y) := \varphi(x_1, \ldots, x_n, y, 0) \wedge \forall w(w < y \to \neg\varphi(x_1, \ldots, x_n, w, 0))$$

$\underline{\mathrm{N}}$-represents F. Let $a \in \mathbf{N}^n$ and let $b = F(a)$. Then $G(a,i) \neq 0$ for $i < b$ and $G(a,b) = 0$. Therefore, $\underline{\mathrm{N}} \vdash \varphi(S^a0, S^b0, z) \leftrightarrow z = 0$ and for $i < b$, $G(a,i) \neq 0$ and $\underline{\mathrm{N}} \vdash \varphi(S^a0, S^i0, z) \leftrightarrow z = S^{G(a,i)}0$. By arguing in models using Lemma 5.4.1 we obtain $\underline{\mathrm{N}} \vdash \psi(S^a0, y) \leftrightarrow y = S^b0$, as claimed.

□

Remark. The converse of this theorem is also true, and is plausible from the Church-Turing Thesis. We shall prove the converse in the next section.

Exercises. In the exercises below, L is a numerical language and Σ is a set of L-sentences.

(1) Suppose $\Sigma \vdash S^m 0 \neq S^n 0$ whenever $m \neq n$. If a function $F : \mathbf{N}^m \to \mathbf{N}$ is Σ-represented by the L-formula $\varphi(x_1, \dots, x_m, y)$, then the graph of F, as a relation of arity $m+1$ on $\mathbf{N}$, is Σ-represented by $\varphi(x_1, \dots, x_m, y)$. (This result applies to $\Sigma = \underline{\mathrm{N}}$, since $\underline{\mathrm{N}} \vdash S^m 0 \neq S^n 0$ whenever $m \neq n$.)

(2) Suppose $\Sigma \supseteq \underline{\mathrm{N}}$. Then the set of all Σ-representable functions $F : \mathbf{N}^m \to \mathbf{N}$, $(m = 0, 1, 2, \dots)$ is closed under composition and minimalization.

5.5 Decidability and Gödel Numbering

Definition. An *L-theory* T is a set of L-sentences closed under provability, that is, whenever $T \vdash \sigma$, then $\sigma \in T$.

Examples.

(1) Given a set Σ of L-sentences, the set $\mathrm{Th}(\Sigma) := \{\sigma : \Sigma \vdash \sigma\}$ of theorems of Σ, is an L-theory. If we need to indicate the dependence on L we write $\mathrm{Th}_L(\Sigma)$ for $\mathrm{Th}(\Sigma)$. We say that Σ *axiomatizes* an L-theory T (or *is an axiomatization of* T) if $T = \mathrm{Th}(\Sigma)$. For $\Sigma = \emptyset$ we also refer to $\mathrm{Th}_L(\Sigma)$ as "predicate logic in L."

(2) Given an L-structure $\mathcal{A}$, the set $\mathrm{Th}(\mathcal{A}) := \{\sigma : \mathcal{A} \models \sigma\}$ is also an L-theory, called the *theory of* $\mathcal{A}$. Note that the theory of $\mathcal{A}$ is automatically complete.

(3) Given any class $\mathcal{K}$ of L-structures, the set

$$\mathrm{Th}(\mathcal{K}) := \{\sigma : \mathcal{A} \models \sigma \text{ for all } \mathcal{A} \in \mathcal{K}\}$$

is an L-theory, called the *theory of* $\mathcal{K}$. For example, for $L = L_{\mathrm{Ri}}$, and $\mathcal{K}$ the class of finite fields, $\mathrm{Th}(\mathcal{K})$ is the set of L-sentences that are true in all finite fields.

The *decision problem* for a given L-theory T is to find an algorithm to decide for any L-sentence σ whether or not σ belongs to T. Since we have not (yet) defined the concept of "algorithm," this is just an informal description at this stage. One of our goals in this section is to define a formal counterpart, called "decidability." In the next section we show that the $L(\underline{\mathrm{N}})$-theory $\mathrm{Th}(\mathfrak{N})$ is *undecidable*; by the Church-Turing Thesis, this means that the decision problem for $\mathrm{Th}(\mathfrak{N})$ has no solution. (This result is a version of Church's Theorem, and is closely related to the Incompleteness Theorem.)

For simplicity, assume from now on that the language L is finite. (We indicate in ... how this assumption can be relaxed.) We shall number the terms and formulas of L in such a way that various statements about these formulas and about formal proofs in this language can be translated "effectively" into equivalent statements about natural numbers expressible by sentences in $L(\underline{\mathrm{N}})$.

Recall that $\mathsf{v}_0, \mathsf{v}_1, \mathsf{v}_2, \ldots$ are our variables. We assign to each symbol

$$s \in L \sqcup \{\text{logical symbols}\} \sqcup \{\mathsf{v}_0, \mathsf{v}_1, \mathsf{v}_2, \ldots\}$$

a *symbol number* $\mathrm{SN}(s) \in \mathbf{N}$ as follows: $\mathrm{SN}(\mathsf{v}_i) := 2i$ and to each remaining symbol (in the finite set $L \sqcup \{\text{logical symbols}\}$) we assign an odd natural number as symbol number, subject to the condition that different symbols have different symbol numbers.

Definition. The Gödel number $\ulcorner t \urcorner$ of an L-term t is defined recursively:

$$\ulcorner t \urcorner = \begin{cases} \langle \mathrm{SN}(\mathsf{v}_i) \rangle & \text{if } t = \mathsf{v}_i, \\ \langle \mathrm{SN}(F), \ulcorner t_1 \urcorner, \ldots, \ulcorner t_n \urcorner \rangle & \text{if } t = F t_1, \ldots, t_n. \end{cases}$$

The Gödel number $\ulcorner \varphi \urcorner$ of an L-formula φ is given recursively by

$$\ulcorner \varphi \urcorner = \begin{cases} \langle \mathrm{SN}(\top) \rangle & \text{if } \varphi = \top, \\ \langle \mathrm{SN}(\bot) \rangle & \text{if } \varphi = \bot, \\ \langle \mathrm{SN}(=), \ulcorner t_1 \urcorner, \ulcorner t_2 \urcorner \rangle & \text{if } \varphi = (t_1 = t_2), \\ \langle \mathrm{SN}(R), \ulcorner t_1 \urcorner, \ldots, \ulcorner t_m \urcorner \rangle & \text{if } \varphi = R t_1 \ldots t_m, \\ \langle \mathrm{SN}(\neg), \ulcorner \psi \urcorner \rangle & \text{if } \varphi = \neg \psi, \\ \langle \mathrm{SN}(\vee), \ulcorner \varphi_1 \urcorner, \ulcorner \varphi_2 \urcorner \rangle & \text{if } \varphi = \varphi_1 \vee \varphi_2, \\ \langle \mathrm{SN}(\wedge), \ulcorner \varphi_1 \urcorner, \ulcorner \varphi_2 \urcorner \rangle & \text{if } \varphi = \varphi_1 \wedge \varphi_2, \\ \langle \mathrm{SN}(\exists), \ulcorner x \urcorner, \ulcorner \psi \urcorner \rangle & \text{if } \varphi = \exists x\, \psi, \\ \langle \mathrm{SN}(\forall), \ulcorner x \urcorner, \ulcorner \psi \urcorner \rangle & \text{if } \varphi = \forall x\, \psi. \end{cases}$$

Lemma 5.5.1. *The following subsets of* $\mathbf{N}$ *are computable:*

(1) $\mathrm{Vble} := \{\ulcorner x \urcorner : x \textit{ is a variable}\}$
(2) $\mathrm{Term} := \{\ulcorner t \urcorner : t \textit{ is an } L\textit{-term}\}$
(3) $\mathrm{AFor} := \{\ulcorner \varphi \urcorner : \varphi \textit{ is an atomic } L\textit{-formula}\}$
(4) $\mathrm{For} := \{\ulcorner \varphi \urcorner : \varphi \textit{ is an } L\textit{-formula}\}$

Proof. (1) $a \in \mathrm{Vble}$ iff $a = \langle 2b \rangle$ for some $b \le a$.
(2) $a \in \mathrm{Term}$ iff $a \in \mathrm{Vble}$ or $a = \langle \mathrm{SN}(F), \ulcorner t_1 \urcorner, \ldots, \ulcorner t_n \urcorner \rangle$ for some function symbol F of L of arity n and L-terms $t_1, \ldots, t_n$ with Gödel numbers $< a$.
We leave (3) to the reader.

$$\text{(4) We have } \mathrm{For}(a) \Leftrightarrow \begin{cases} \mathrm{For}((a)_1) & \text{if } a = \langle \mathrm{SN}(\neg), (a)_1 \rangle, \\ \mathrm{For}((a)_1) \text{ and } \mathrm{For}((a)_2) & \text{if } a = \langle \mathrm{SN}(\vee), (a)_1, (a)_2 \rangle \\ & \text{or } a = \langle \mathrm{SN}(\wedge), (a)_1, (a)_2 \rangle, \\ \mathrm{Vble}((a)_1) \text{ and } \mathrm{For}((a)_2) & \text{if } a = \langle \mathrm{SN}(\exists), (a)_1, (a)_2 \rangle \\ & \text{or } a = \langle \mathrm{SN}(\forall), (a)_1, (a)_2 \rangle, \\ \mathrm{AFor}(a) & \text{otherwise.} \end{cases}$$

So For is computable. $\square$

In the next two lemmas, x ranges over variables, φ and ψ over L-formulas, and t and τ over L-terms.

Lemma 5.5.2. *The function* $\mathrm{Sub} : \mathbf{N}^3 \to \mathbf{N}$ *defined by* $\mathrm{Sub}(a,b,c) =$

$$\begin{cases} c & \text{if } \mathrm{Vble}(a) \text{ and } a = b, \\ \langle (a)_0, \mathrm{Sub}((a)_1,b,c), \dots, \mathrm{Sub}((a)_n,b,c)\rangle & \text{if } a = \langle (a)_0, \dots, (a)_n\rangle \text{ with } n > 0 \text{ and} \\ & (a)_0 \neq \mathrm{SN}(\exists),\ (a)_0 \neq \mathrm{SN}(\forall), \\ \langle \mathrm{SN}(\exists), (a)_1, \mathrm{Sub}((a)_2,b,c)\rangle & \text{if } a = \langle \mathrm{SN}(\exists), (a)_1, (a)_2\rangle \text{ and } (a)_1 \neq b, \\ \langle \mathrm{SN}(\forall), (a)_1, \mathrm{Sub}((a)_2,b,c)\rangle & \text{if } a = \langle \mathrm{SN}(\forall), (a)_1, (a)_2\rangle \text{ and } (a)_1 \neq b, \\ a & \text{otherwise} \end{cases}$$

is computable, and satisfies

$$\mathrm{Sub}(\ulcorner t \urcorner, \ulcorner x \urcorner, \ulcorner \tau \urcorner) = \ulcorner t(\tau/x) \urcorner \text{ and } \mathrm{Sub}(\ulcorner \varphi \urcorner, \ulcorner x \urcorner, \ulcorner \tau \urcorner) = \ulcorner \varphi(\tau/x) \urcorner.$$

Proof. Exercise. □

Lemma 5.5.3. *The following relations on* $\mathbf{N}$ *are computable:*
(1) $\mathrm{Fr} := \{(\ulcorner\varphi\urcorner, \ulcorner x\urcorner) : x \textit{ occurs free in } \varphi\} \subseteq \mathbf{N}^2$
(2) $\mathrm{FrSub} := \{(\ulcorner\varphi\urcorner, \ulcorner x\urcorner, \ulcorner\tau\urcorner) : \tau \textit{ is free for } x \textit{ in } \varphi\} \subseteq \mathbf{N}^3$
(3) $\mathrm{PrAx} := \{\ulcorner\varphi\urcorner : \varphi \textit{ is a propositional axiom}\} \subseteq \mathbf{N}$
(4) $\mathrm{Eq} := \{\ulcorner\varphi\urcorner : \varphi \textit{ is an equality axiom}\} \subseteq \mathbf{N}$
(5) $\mathrm{Quant} := \{\ulcorner\psi\urcorner : \psi \textit{ is a quantifier axiom}\} \subseteq \mathbf{N}$
(6) $\mathrm{MP} := \{(\ulcorner\varphi_1\urcorner, \ulcorner\varphi_1 \to \varphi_2\urcorner, \ulcorner\varphi_2\urcorner) : \varphi_1, \varphi_2 \textit{ are } L\textit{-formulas}\} \subseteq \mathbf{N}^3$
(7) $\mathrm{Gen} := \{(\ulcorner\varphi\urcorner, \ulcorner\psi\urcorner) : \psi \textit{ follows from } \varphi \textit{ by the generalization rule}\} \subseteq \mathbf{N}^2$
(8) $\mathrm{Sent} := \{\ulcorner\varphi\urcorner : \varphi \textit{ is a sentence}\} \subseteq \mathbf{N}$

Proof. The usual inductive or explicit description of each of these notions translates easily into a description of its "Gödel image" that establishes computability of this image. As an example, note that

$$\mathrm{Sent}(a) \Longleftrightarrow \mathrm{For}(a) \text{ and } \forall i_{<a} \neg \mathrm{Fr}(a,i),$$

so (8) follows from (1) and earlier results. □

In the rest of this Section Σ is a set of L-sentences. Put

$$\ulcorner\Sigma\urcorner := \{\ulcorner\sigma\urcorner : \sigma \in \Sigma\},$$

and call Σ *computable* if $\ulcorner\Sigma\urcorner$ is computable.

Definition. We define Prf_Σ to be the "set of all Gödel numbers of proofs from Σ", i. e. $\mathrm{Prf}_\Sigma := \{\langle \ulcorner\varphi_1\urcorner, \dots, \ulcorner\varphi_n\urcorner\rangle : \varphi_1, \dots, \varphi_n \text{ is a proof from } \Sigma\}$. So every element of Prf_Σ is of the form $\langle \ulcorner\varphi_1\urcorner, \dots, \ulcorner\varphi_n\urcorner\rangle$ where $n \geq 1$ and every φ_k is either in Σ, or a logical axiom, or obtained from some φ_i, φ_j with $1 \leq i, j < k$ by Modus Ponens, or obtained from some φ_i with $1 \leq i < k$ by Generalization.

Lemma 5.5.4. *If* Σ *is computable, then* Prf_Σ *is computable.*

Proof. This is because a is in Prf_Σ iff $\mathrm{Seq}(a)$ and $\mathrm{lh}(a) \neq 0$ and for every $k < \mathrm{lh}(a)$ either $(a)_k \in \ulcorner\Sigma\urcorner \cup \mathrm{PrAx} \cup \mathrm{Eq} \cup \mathrm{Quant}$ or $\exists i, j < k : \mathrm{MP}((a)_i, (a)_j, (a)_k)$ or $\exists i < k : \mathrm{Gen}((a)_i, (a)_k)$. □

Definition. A relation $R \subseteq \mathbf{N}^n$ is said to be *computably generated* if there is a computable relation $Q \subseteq \mathbf{N}^{n+1}$ such that for all $a \in \mathbf{N}^n$ we have

$$R(a) \Leftrightarrow \exists x Q(a, x)$$

"Recursively enumerable" is also used for "computably generated."

Remark. Every computable relation is obviously computably generated. We leave it as an exercise to check that the union and intersection of two computably generated n-ary relations on $\mathbf{N}$ are computably generated. The complement of a computably generated subset of $\mathbf{N}$ is not always computably generated, as we shall see later.

Lemma 5.5.5. *If Σ is computable, then $\ulcorner\mathrm{Th}(\Sigma)\urcorner$ is computably generated.*

Proof. Apply Lemma 5.5.4 and the fact that for all $a \in \mathbf{N}$

$$a \in \ulcorner\mathrm{Th}(\Sigma)\urcorner \Longleftrightarrow \exists b\big(\mathrm{Prf}_\Sigma(b) \text{ and } a = (b)_{\mathrm{lh}(b) \dot{-} 1} \text{ and } \mathrm{Sent}(a)\big).$$

□

Definition. An L-theory T is said to be *computably axiomatizable* if T has a computable axiomatization.[2]

We say that T is *decidable* if $\ulcorner T\urcorner$ is computable, and *undecidable* otherwise. (Thus "T is decidable" means the same thing as "T is computable," but for L-theories "decidable" is more widely used than "computable".)

Proposition 5.5.6 (Negation Theorem). *Let $A \subseteq \mathbf{N}^n$ and suppose A and $\neg A$ are computably generated. Then A is computable.*

Proof. Let $P, Q \subseteq \mathbf{N}^{n+1}$ be computable such that for all $a \in \mathbf{N}^n$ we have

$$A(a) \Longleftrightarrow \exists x P(a, x), \qquad \neg A(a) \Longleftrightarrow \exists x Q(a, x).$$

Then there is for each $a \in \mathbf{N}^n$ an $x \in \mathbf{N}$ such that $(P \vee Q)(a, x)$. The computability of A follows by noting that for all $a \in \mathbf{N}^n$ we have

$$A(a) \Longleftrightarrow P(a, \mu x (P \vee Q)(a, x)).$$

□

Proposition 5.5.7. *Every complete and computably axiomatizable L-theory is decidable.*

[2] Instead of "computably axiomatizable," also "recursively axiomatizable" and "effectively axiomatizable" are used.

Proof. Let T be a complete L-theory with computable axiomatization Σ. Then $\ulcorner T \urcorner = \ulcorner \mathrm{Th}(\Sigma) \urcorner$ is computably generated. Now observe:

$$\begin{aligned} a \notin \ulcorner T \urcorner &\Longleftrightarrow a \notin \mathrm{Sent} \text{ or } \langle \mathrm{SN}(\neg), a \rangle \in \ulcorner T \urcorner \\ &\Longleftrightarrow a \notin \mathrm{Sent} \text{ or } \exists b \big(\mathrm{Prf}_\Sigma(b) \text{ and } (b)_{\mathrm{lh}(b) \dot{-} 1} = \langle \mathrm{SN}(\neg), a \rangle\big). \end{aligned}$$

Hence the complement of $\ulcorner T \urcorner$ is computably generated. Thus T is decidable by the Negation Theorem. $\square$

Exercises.

(1) Let a and b denote positive real numbers. Call a *computable* if there are computable functions $f, g : \mathbf{N} \to \mathbf{N}$ such that for all $n > 0$,

$$g(n) \neq 0 \text{ and } |a - f(n)/g(n)| < 1/n.$$

Then:

(i) every positive rational number is computable, and e is computable;

(ii) if a and b are computable, so are $a + b$, ab, and $1/a$, and if in addition $a > b$, then $a - b$ is also computable;

(iii) a is computable if and only if the binary relation R_a on $\mathbf{N}$ defined by

$$R_a(m, n) \iff n > 0 \text{ and } m/n < a$$

is computable. (Hint: use the Negation Theorem.)

(2) A nonempty $S \subseteq \mathbf{N}$ is computably generated iff there is a computable function $f : \mathbf{N} \to \mathbf{N}$ such that $S = f(\mathbf{N})$. Moreover, if S is infinite and computably generated, then f can be chosen injective.

(3) If $f : \mathbf{N} \to \mathbf{N}$ is computable and $f(x) > x$ for all $x \in \mathbf{N}$, then $f(\mathbf{N})$ is computable.

(4) Every infinite computably generated subset of $\mathbf{N}$ has an infinite computable subset.

(5) A function $F : \mathbf{N}^n \to \mathbf{N}$ is computable iff its graph is computably generated.

(6) Suppose Σ is a computable and consistent set of sentences in the numerical language L. Then every Σ-representable relation $R \subseteq \mathbf{N}^n$ is computable.

(7) A function $F : \mathbf{N}^m \to \mathbf{N}$ is computable if and only if it is Σ-representable for some finite, consistent set $\Sigma \supseteq \underline{\mathrm{N}}$ of sentences in some numerical language $L \supseteq L(\underline{\mathrm{N}})$. The last exercise gives an alternative characterization of "computable function."

5.6 Theorems of Gödel and Church

In this section we assume that the finite language L extends $L(\underline{\mathrm{N}})$.

Theorem 5.6.1 (Church). *No consistent L-theory extending $\underline{\mathrm{N}}$ is decidable.*

Before giving the proof we record the following consequence:

Corollary 5.6.2 (Weak form of Gödel's Incompleteness Theorem). *Each computably axiomatizable L-theory extending $\underline{\mathrm{N}}$ is incomplete.*

Proof. Immediate from 5.5.7 and Church's Theorem. $\square$

We will indicate in the next Section how to *construct* for any consistent computable set of L-sentences $\Sigma \supseteq \underline{\mathrm{N}}$ an L-sentence σ such that $\Sigma \nvdash \sigma$ and $\Sigma \nvdash \neg\sigma$. (The corollary above only says that such a sentence exists.)

For the proof of Church's Theorem we need a few lemmas.

Lemma 5.6.3. *The function* $\mathrm{Num} : \mathbf{N} \to \mathbf{N}$ *defined by* $\mathrm{Num}(a) = \ulcorner S^a 0 \urcorner$ *is computable.*

Proof. $\mathrm{Num}(0) = \ulcorner 0 \urcorner$ and $\mathrm{Num}(a+1) = \langle \mathrm{SN}(S), \mathrm{Num}(a) \rangle$. $\square$

Let $P \subseteq A^2$ be any binary relation on a set A. For $a \in A$, we let $P(a) \subseteq A$ be given by the equivalence $P(a)(b) \Leftrightarrow P(a,b)$.

Lemma 5.6.4 (Cantor). *Given any $P \subseteq A^2$, its antidiagonal $Q \subseteq A$ defined by*

$$Q(b) \Longleftrightarrow \neg P(b,b)$$

is not of the form $P(a)$ for any $a \in A$.

Proof. Suppose $Q = P(a)$, where $a \in A$. Then $Q(a)$ iff $P(a,a)$. But by definition, $Q(a)$ iff $\neg P(a,a)$, a contradiction. $\square$

This is essentially Cantor's proof that no $f : A \to \mathfrak{P}(A)$ can be surjective. (Use $P(a,b) :\Leftrightarrow b \in f(a)$; then $P(a) = f(a)$.)

Definition. Let Σ be a set of L-sentences. We fix a variable x (e. g. $x = \mathsf{v}_0$) and define the binary relation $P^\Sigma \subseteq \mathbf{N}^2$ by

$$P^\Sigma(a,b) \Longleftrightarrow \mathrm{Sub}(a, \ulcorner x \urcorner, \mathrm{Num}(b)) \in \ulcorner \mathrm{Th}(\Sigma) \urcorner$$

For an L-formula $\varphi(x)$ and $a = \ulcorner \varphi(x) \urcorner$, we have

$$\mathrm{Sub}(\ulcorner \varphi(x) \urcorner, \ulcorner x \urcorner, \ulcorner S^b 0 \urcorner) = \ulcorner \varphi(S^b 0) \urcorner,$$

so

$$P^\Sigma(a,b) \Longleftrightarrow \Sigma \vdash \varphi(S^b 0).$$

Lemma 5.6.5. *Suppose $\Sigma \supseteq \underline{\mathrm{N}}$ is consistent. Then each computable set $X \subseteq \mathbf{N}$ is of the form $X = P^\Sigma(a)$ for some $a \in \mathbf{N}$.*

Proof. Let $X \subseteq \mathbf{N}$ be computable. Then X is Σ-representable by Theorem 5.4.4, say by the formula $\varphi(x)$, i. e. $X(b) \Rightarrow \Sigma \vdash \varphi(S^b 0)$, and $\neg X(b) \Rightarrow \Sigma \vdash \neg\varphi(S^b 0)$. So $X(b) \Leftrightarrow \Sigma \vdash \varphi(S^b 0)$ (using consistency to get "$\Leftarrow$"). Take $a = \ulcorner \varphi(x) \urcorner$; then $X(b)$ iff $\Sigma \vdash \varphi(S^b 0)$ iff $P^\Sigma(a,b)$, that is, $X = P^\Sigma(a)$. $\square$

Proof of Church's Theorem. Let $\Sigma \supseteq \underline{\mathrm{N}}$ be consistent. We have to show that then $\mathrm{Th}(\Sigma)$ is undecidable, that is, $\ulcorner \mathrm{Th}(\Sigma) \urcorner$ is not computable. Suppose that $\ulcorner \mathrm{Th}(\Sigma) \urcorner$ is computable. Then the antidiagonal $Q^\Sigma \subseteq \mathbf{N}$ of P^Σ is computable:

$$b \in Q^\Sigma \Leftrightarrow (b,b) \notin P^\Sigma \Leftrightarrow \mathrm{Sub}(b, \ulcorner x \urcorner, \mathrm{Num}(b)) \notin \ulcorner \mathrm{Th}(\Sigma) \urcorner.$$

By Lemma 5.6.4, Q^Σ is not among the $P^\Sigma(a)$. Therefore by Lemma 5.6.5, Q^Σ is not computable, a contradiction. This concludes the proof.

By Lemma 5.5.5 the subset $\ulcorner \mathrm{Th}(\underline{\mathrm{N}}) \urcorner$ of $\mathbf{N}$ is computably generated. But this set is not computable:

Corollary 5.6.6. $\mathrm{Th}(\underline{\mathrm{N}})$ *and* $\mathrm{Th}(\emptyset)$ (in the language $L(\underline{\mathrm{N}})$) are undecidable.

Proof. The undecidability of $\mathrm{Th}(\underline{\mathrm{N}})$ is a special case of Church's Theorem. Let $\wedge\underline{\mathrm{N}}$ be the sentence $\underline{\mathrm{N1}} \wedge \cdots \wedge \underline{\mathrm{N9}}$. Then, for any $L(\underline{\mathrm{N}})$-sentence σ,

$$\underline{\mathrm{N}} \vdash \sigma \iff \emptyset \vdash \wedge\underline{\mathrm{N}} \to \sigma,$$

that is, for all $a \in \mathbf{N}$,

$$a \in \ulcorner \mathrm{Th}(\underline{\mathrm{N}}) \urcorner \iff a \in \mathrm{Sent} \text{ and } \langle \mathrm{SN}(\vee), \langle \mathrm{SN}(\neg), \ulcorner \wedge\underline{\mathrm{N}} \urcorner \rangle, a \rangle \in \ulcorner \mathrm{Th}(\emptyset) \urcorner.$$

Therefore, if $\mathrm{Th}(\emptyset)$ were decidable, then $\mathrm{Th}(\underline{\mathrm{N}})$ would be decidable; but $\mathrm{Th}(\underline{\mathrm{N}})$ is undecidable. So $\mathrm{Th}(\emptyset)$ is undecidable. ☐

Discussion. We have seen that $\underline{\mathrm{N}}$ is quite weak. A very strong set of axioms in the language $L(\underline{\mathrm{N}})$ is PA (1st order Peano Arithmetic). Its axioms are those of $\underline{\mathrm{N}}$ together with all induction axioms, that is, all sentences of the form

$$\forall x \left[\big(\varphi(x,0) \wedge \forall y \left(\varphi(x,y) \to \varphi(x,Sy) \right) \big) \to \forall y\, \varphi(x,y) \right]$$

where $\varphi(x,y)$ is an $L(\underline{\mathrm{N}})$-formula, $x = (x_1, \ldots, x_n)$, and $\forall x$ stands for $\forall x_1 \ldots \forall x_n$.

Note that PA is consistent, since it has $\mathfrak{N} = (\mathbf{N};\ <, 0, S, +, \cdot)$ as a model. Also $\ulcorner \mathrm{PA} \urcorner$ is computable (exercise). Thus by the theorems above, Th(PA) is undecidable and incomplete. To appreciate the significance of this result, one needs a little background knowledge, including some history.

Over a century of experience has shown that number theoretic assertions can be expressed by sentences of $L(\underline{\mathrm{N}})$, admittedly in an often contorted way. (That is, we know how to construct for any number theoretic statement a sentence σ of $L(\underline{\mathrm{N}})$ such that the statement is true if and only if $\mathfrak{N} \models \sigma$. In most cases we just *indicate* how to construct such a sentence, since an actual sentence would be too unwieldy without abbreviations.)

What is more important, we know from experience that any established fact of classical number theory—including results obtained by sophisticated analytic and algebraic methods—can be proved from PA, in the sense that $\mathrm{PA} \vdash \sigma$ for the sentence σ expressing that fact. Thus before Gödel's Incompleteness Theorem it seemed natural to conjecture that PA is complete. (Apparently it was not widely recognized at the time that completeness of PA would have had the astonishing consequence that $\mathrm{Th}(\mathfrak{N})$ is decidable.) Of course, the situation cannot be remedied by adding new axioms to PA, at least if we insist that the axioms are true in $\mathfrak{N}$ and that we have effective means to tell which sentences are axioms. In this sense, the Incompleteness Theorem is pervasive.

5.7 A more explicit incompleteness theorem

In Section 5.6 we obtained Gödel's Incompleteness Theorem as an immediate corollary of Church's theorem. In this section, we prove the incompleteness theorem in the more explicit form stated in the introduction to this chapter.

In this section $L \supseteq L(\underline{\mathrm{N}})$ is a finite language, and Σ is a set of L-sentences. We also fix two distinct variables x and y.

We shall indicate how to construct, for any computable consistent $\Sigma \supseteq \underline{\mathrm{N}}$, a formula $\varphi(x)$ of $L(\underline{\mathrm{N}})$ with the following properties:

(i) $\underline{\mathrm{N}} \vdash \varphi(S^m 0)$ for each m;

(ii) $\Sigma \nvdash \forall x \varphi(x)$.

Note that then the sentence $\forall x \varphi(x)$ is true in $\mathfrak{N}$ but not provable from Σ. Here is a sketch of how to make such a sentence. Assume for simplicity that $L = L(\underline{\mathrm{N}})$ and $\mathfrak{N} \models \Sigma$. The idea is to construct sentences σ and σ' such that

(1) $\mathfrak{N} \models \sigma \leftrightarrow \sigma'$; and (2) $\mathfrak{N} \models \sigma' \iff \Sigma \nvdash \sigma$.

From (1) and (2) we get $\mathfrak{N} \models \sigma \iff \Sigma \nvdash \sigma$. Assuming that $\mathfrak{N} \models \neg\sigma$ produces a contradiction. Hence σ is true in $\mathfrak{N}$, and thus(!) not provable from Σ.

How to implement this strange idea? To take care of (2), one might guess that $\sigma' = \forall x \neg Pr(x, S^{\ulcorner \sigma \urcorner} 0)$ where $Pr(x, y)$ is a formula representing in $\underline{\mathrm{N}}$ the binary relation $\mathrm{Pr} \subseteq \mathbf{N}^2$ defined by

$$\begin{aligned} \mathrm{Pr}(m, n) \iff {} & m \text{ is the Gödel number of a proof from } \Sigma \\ & \text{of a sentence with Gödel number } n. \end{aligned}$$

But how do we arrange (1)? Since $\sigma' := \forall x \neg Pr(x, S^{\ulcorner \sigma \urcorner} 0)$ depends on σ, the solution is to apply the fixed-point lemma below to $\rho(y) := \forall x \neg Pr(x, y)$.

This finishes our sketch. What follows is a rigorous implementation.

Lemma 5.7.1. *Suppose $\Sigma \supseteq \underline{\mathrm{N}}$. Then there is for every L-formula $\rho(y)$ an L-sentence σ such that $\Sigma \vdash \sigma \leftrightarrow \rho(S^n 0)$ where $n = \ulcorner \sigma \urcorner$.*

Proof. The function $(a, b) \mapsto \mathrm{Sub}(a, \ulcorner x \urcorner, \mathrm{Num}(b)) : \mathbf{N}^2 \to \mathbf{N}$ is computable by Lemma 5.5.2. Hence by the representability theorem it is $\underline{\mathrm{N}}$-representable. Let $\mathrm{sub}(x_1, x_2, y)$ be an $L(\underline{\mathrm{N}})$-formula representing it in $\underline{\mathrm{N}}$. We can assume that the variable x does not occur in $\mathrm{sub}(x_1, x_2, y)$. Then for all a, b in $\mathbf{N}$,

$$\underline{\mathrm{N}} \vdash \mathrm{sub}(S^a 0, S^b 0, y) \leftrightarrow y = S^c 0, \quad \text{where } c = \mathrm{Sub}(a, \ulcorner x \urcorner, \mathrm{Num}(b)) \tag{1}$$

Now let $\rho(y)$ be an L-formula. Define $\theta(x) := \exists y(\mathrm{sub}(x,x,y) \wedge \rho(y))$ and let $m = \ulcorner\theta(x)\urcorner$. Let $\sigma := \theta(S^m0)$, and put $n = \ulcorner\sigma\urcorner$. We claim that

$$\Sigma \vdash \sigma \leftrightarrow \rho(S^n0).$$

Indeed,

$$n = \ulcorner\sigma\urcorner = \ulcorner\theta(S^m0)\urcorner = \mathrm{Sub}(\ulcorner\theta(x)\urcorner, \ulcorner x\urcorner, \ulcorner S^m0\urcorner) = \mathrm{Sub}(m, \ulcorner x\urcorner, \mathrm{Num}(m)).$$

So by (1),

$$\underline{\mathrm{N}} \vdash \mathrm{sub}(S^m0, S^m0, y) \leftrightarrow y = S^n0 \tag{2}$$

We have

$$\sigma = \theta(S^m0) = \exists y(\mathrm{sub}(S^m0, S^m0, y) \wedge \rho(y)),$$

so by (2) we get $\Sigma \vdash \sigma \leftrightarrow \exists y(y = S^n0 \wedge \rho(y))$. Hence, $\Sigma \vdash \sigma \leftrightarrow \rho(S^n0)$. $\square$

Theorem 5.7.2. *Suppose Σ is consistent, computable, and proves all axioms of $\underline{\mathrm{N}}$. Then there exists an $L(\underline{\mathrm{N}})$-formula $\varphi(x)$ such that $\underline{\mathrm{N}} \vdash \varphi(S^m0)$ for each m, but $\Sigma \nvdash \forall x\varphi(x)$.*

Proof. Consider the relation $\mathrm{Pr}_\Sigma \subseteq \mathbf{N}^2$ defined by

$$\mathrm{Pr}_\Sigma(m,n) \iff m \text{ is the Gödel number of a proof from } \Sigma \text{ of an } L\text{-sentence with Gödel number } n.$$

Since Σ is computable, Pr_Σ is computable. Hence Pr_Σ is representable in $\underline{\mathrm{N}}$. Let $Pr_\Sigma(x,y)$ be an $L(\underline{\mathrm{N}})$-formula representing Pr_Σ in $\underline{\mathrm{N}}$, and hence in Σ. Because Σ is consistent we have for all m, n:

$$\Sigma \vdash Pr_\Sigma(S^m0, S^n0) \iff \mathrm{Pr}_\Sigma(m,n) \tag{1}$$

$$\Sigma \vdash \neg Pr_\Sigma(S^m0, S^n0) \iff \neg\mathrm{Pr}_\Sigma(m,n) \tag{2}$$

Let $\rho(y)$ be the $L(\underline{\mathrm{N}})$-formula $\forall x \neg Pr_\Sigma(x,y)$. Lemma 5.7.1 (with $L = L(\underline{\mathrm{N}})$ and $\Sigma = \underline{\mathrm{N}}$) provides an $L(\underline{\mathrm{N}})$-sentence σ such that $\underline{\mathrm{N}} \vdash \sigma \leftrightarrow \rho(S^{\ulcorner\sigma\urcorner}0)$. It follows that $\Sigma \vdash \sigma \leftrightarrow \rho(S^{\ulcorner\sigma\urcorner}0)$, that is

$$\Sigma \vdash \sigma \leftrightarrow \forall x \neg Pr_\Sigma(x, S^{\ulcorner\sigma\urcorner}0) \tag{3}$$

Claim: $\Sigma \nvdash \sigma$. Assume towards a contradiction that $\Sigma \vdash \sigma$; let m be the Gödel number of a proof of σ from Σ, so $\mathrm{Pr}_\Sigma(m, \ulcorner\sigma\urcorner)$. Because of (3) we also have $\Sigma \vdash \forall x \neg Pr_\Sigma(x, S^{\ulcorner\sigma\urcorner}0)$, so $\Sigma \vdash \neg Pr_\Sigma(S^m0, S^{\ulcorner\sigma\urcorner}0)$, which by (2) yields $\neg\mathrm{Pr}_\Sigma(m, \ulcorner\sigma\urcorner)$, a contradiction. This establishes the claim.

Now put $\varphi(x) := \neg Pr_\Sigma(x, S^{\ulcorner\sigma\urcorner}0)$. We now show :

(i) $\underline{\mathrm{N}} \vdash \varphi(S^m0)$ for each m. Because $\Sigma \nvdash \sigma$, no m is the Gödel number of a proof of σ from Σ. Hence $\neg\mathrm{Pr}_\Sigma(m, \ulcorner\sigma\urcorner)$ for each m, which by the defining property of Pr_Σ yields $\underline{\mathrm{N}} \vdash \neg Pr_\Sigma(S^m0, S^{\ulcorner\sigma\urcorner}0)$ for each m, that is, $\underline{\mathrm{N}} \vdash \varphi(S^m0)$ for each m.

(ii) $\Sigma \nvdash \forall x\varphi(x)$. This is because $\Sigma \vdash \sigma \leftrightarrow \forall x\varphi(x)$ by (3).

□

Corollary 5.7.3. *Suppose that Σ is computable and true in the L-expansion $\mathfrak{N}^*$ of $\mathfrak{N}$. Then there exists an $L(\underline{\mathrm{N}})$-formula $\varphi(x)$ such that $\underline{\mathrm{N}} \vdash \varphi(S^n 0)$ for each n, but $\Sigma \cup \underline{\mathrm{N}} \nvdash \forall x\varphi(x)$.*

(Note that then $\forall x\varphi(x)$ is true in $\mathfrak{N}^*$ but not provable from Σ.) To obtain this corollary, apply the theorem above to $\Sigma \cup \underline{\mathrm{N}}$ in place of Σ.

Exercises.

(1) Let $\mathfrak{N}^*$ be an L-expansion of $\mathfrak{N}$. Then the set of Gödel numbers of L-sentences true in $\mathfrak{N}^*$ is not definable in $\mathfrak{N}^*$.

This result (of Tarski) is known as the *undefinability of truth*. It strengthens the special case of Church's theorem which says that the set of Gödel numbers of L-sentences true in $\mathfrak{N}^*$ is not computable. Tarski's result follows easily from the fixed point lemma, as does the more general result in the next exercise.

(2) Suppose $\Sigma \supseteq \underline{\mathrm{N}}$ is consistent. Then the set $\ulcorner \mathrm{Th}(\Sigma) \urcorner$ is not Σ-representable, and there is no truth definition for Σ. (Here a *truth definition for* Σ is an L-formula $\mathrm{true}(y)$ such that for all L-sentences σ,

$$\Sigma \vdash \sigma \longleftrightarrow \mathrm{true}(S^n 0), \text{ where } n = \ulcorner \sigma \urcorner.$$

5.8 Undecidable Theories

Church's theorem says that any consistent theory containing a certain basic amount of integer arithmetic is undecidable. How about theories like Th(Fl) (the theory of fields), and Th(Gr) (the theory of groups)? An easy way to prove the undecidability of such theories is due to Tarski: he noticed that if $\mathfrak{N}$ is definable in *some* model of a theory T, then T is undecidable. The aim of this section is to establish this result and indicate some applications. In order not to distract from this theme by boring details, we shall occasionally replace a proof by an appeal to the Church-Turing Thesis. (A conscientious reader will replace these appeals by proofs until reaching a level of skill that makes constructing such proofs a predictable routine.)

In this section, L and L' are finite languages, Σ is a set of L-sentences, and Σ' is a set of L'-sentences.

Lemma 5.8.1. *Let $L \subseteq L'$ and $\Sigma \subseteq \Sigma'$.*

(1) *Suppose Σ' is conservative over Σ. Then*

$$\mathrm{Th}_L(\Sigma) \textit{ is undecidable} \implies \mathrm{Th}_{L'}(\Sigma') \textit{ is undecidable.}$$

(2) *Suppose $L = L'$ and $\Sigma' \setminus \Sigma$ is finite. Then*

$$\mathrm{Th}(\Sigma') \textit{ is undecidable} \implies \mathrm{Th}(\Sigma) \textit{ is undecidable.}$$

(3) *Suppose all symbols of $L' \smallsetminus L$ are constant symbols. Then*

$$\mathrm{Th}_L(\Sigma) \text{ is undecidable} \quad \Longleftrightarrow \quad \mathrm{Th}_{L'}(\Sigma) \text{ is undecidable.}$$

(4) *Suppose Σ' extends Σ by a definition. Then*

$$\mathrm{Th}_L(\Sigma) \text{ is undecidable} \quad \Longleftrightarrow \quad \mathrm{Th}_{L'}(\Sigma') \text{ is undecidable.}$$

Proof. (1) In this case we have for all $a \in \mathbf{N}$,

$$a \in \ulcorner \mathrm{Th}_L(\Sigma) \urcorner \Longleftrightarrow a \in \mathrm{Sent}_L \text{ and } a \in \ulcorner \mathrm{Th}_{L'}(\Sigma') \urcorner.$$

It follows that if $\mathrm{Th}_{L'}(\Sigma')$ is decidable, so is $\mathrm{Th}_L(\Sigma)$.
(2) Write $\Sigma' = \{\sigma_1, \dots, \sigma_N\} \cup \Sigma$, and put $\sigma' := \sigma_1 \wedge \dots \wedge \sigma_N$. Then for each L-sentence σ we have $\Sigma' \vdash \sigma \Longleftrightarrow \Sigma \vdash \sigma' \rightarrow \sigma$, so for all $a \in \mathbf{N}$,

$$a \in \ulcorner \mathrm{Th}(\Sigma') \urcorner \Longleftrightarrow a \in \mathrm{Sent} \text{ and } \langle \mathrm{SN}(\vee), \langle \mathrm{SN}(\neg), \ulcorner \sigma' \urcorner \rangle, a \rangle \in \ulcorner \mathrm{Th}(\Sigma) \urcorner.$$

It follows that if $\mathrm{Th}(\Sigma)$ is decidable then so is $\mathrm{Th}(\Sigma')$.
(3) Let $c_0, \dots, c_n$ be the distinct constant symbols of $L' \smallsetminus L$. Given any L'-sentence σ we define the L-sentence σ' as follows: take $k \in \mathbf{N}$ minimal such that σ contains no variable v_m with $m \geq k$, replace each occurrence of c_i in σ by v_{k+i} for $i = 0, \dots, n$, and let $\varphi(\mathsf{v}_k, \dots, \mathsf{v}_{k+n})$ be the resulting L-formula (so $\sigma = \varphi(c_0, \dots, c_n)$); then $\sigma' := \forall \mathsf{v}_k \dots \forall \mathsf{v}_{k+n} \varphi(\mathsf{v}_k, \dots, \mathsf{v}_{k+n})$. An easy argument using the completeness theorem shows that

$$\Sigma \vdash_{L'} \sigma \quad \Longleftrightarrow \quad \Sigma \vdash_L \sigma'.$$

By the Church-Turing Thesis there is a computable function $a \mapsto a' : \mathbf{N} \rightarrow \mathbf{N}$ such that $\ulcorner \sigma' \urcorner = \ulcorner \sigma \urcorner'$ for all L'-sentences σ; we leave it to the reader to replace this appeal to the Church-Turing Thesis by a proof. Then, for all $a \in \mathbf{N}$:

$$\ulcorner \mathrm{Th}_{L'}(\Sigma) \urcorner \Longleftrightarrow a \in \mathrm{Sent}_{L'} \text{ and } a' \in \ulcorner \mathrm{Th}_L(\Sigma) \urcorner.$$

This yields the $\Leftarrow$ direction of (3); the converse holds by (1).
(4) The $\Rightarrow$ direction holds by (1). For the $\Leftarrow$ we use an algorithm (see ...) that computes for each L'-sentence σ an L-sentence σ^* such that $\Sigma' \vdash \sigma \leftrightarrow \sigma^*$. By the Church-Turing Thesis there is a computable function $a \mapsto a^* : \mathbf{N} \rightarrow \mathbf{N}$ such that $\ulcorner \sigma^* \urcorner = \ulcorner \sigma \urcorner^*$ for all L'-sentences σ. Hence, for all $a \in \mathbf{N}$,

$$a \in \ulcorner \mathrm{Th}_{L'}(\Sigma') \urcorner \Longleftrightarrow \mathrm{Sent}_{L'}(a) \text{ and } a^* \in \ulcorner \mathrm{Th}_L(\Sigma) \urcorner.$$

This yields the $\Leftarrow$ direction of (4). □

Remark. We cannot drop the assumption $L = L'$ in (2): take $L = \emptyset$, $\Sigma = \emptyset$, $L' = L(\underline{\mathrm{N}})$ and $\Sigma' = \emptyset$. Then $\mathrm{Th}_{L'}(\Sigma')$ is undecidable by Corollary 5.6.6, but $\mathrm{Th}_L(\Sigma)$ is decidable (exercise).

Definition. An L-structure $\mathcal{A}$ is said to be *strongly undecidable* if for every set Σ of L-sentences such that $\mathcal{A} \models \Sigma$, $\mathrm{Th}(\Sigma)$ is undecidable.

So $\mathcal{A}$ is strongly undecidable iff every L-theory of which $\mathcal{A}$ is a model is undecidable.

Example. $\mathfrak{N} = (\mathbf{N};\ <, 0, S, +, \cdot)$ is strongly undecidable. To see this, let Σ be a set of $L(\underline{\mathrm{N}})$-sentences such that $\mathfrak{N} \models \Sigma$. We have to show that $\mathrm{Th}(\Sigma)$ is undecidable. Now $\mathfrak{N} \models \Sigma \cup \underline{\mathrm{N}}$. By Church's Theorem $\mathrm{Th}(\Sigma \cup \underline{\mathrm{N}})$ is undecidable, hence $\mathrm{Th}(\Sigma)$ is undecidable by part (2) of Lemma 5.8.1.

The following result is an easy application of part (3) of the previous lemma.

Lemma 5.8.2. *Let $c_0, \dots, c_n$ be distinct constant symbols not in L, and let $(\mathcal{A}, a_0, \dots, a_n)$ be an $L(c_0, \dots, c_n)$-expansion of the L-structure $\mathcal{A}$. Then*

$$(\mathcal{A}, a_0, \dots, a_n) \textit{ is strongly undecidable} \Longrightarrow \mathcal{A} \textit{ is strongly undecidable.}$$

Theorem 5.8.3 (Tarski). *Suppose the L-structure $\mathcal{A}$ is definable in the L'-structure $\mathcal{B}$ and $\mathcal{A}$ is strongly undecidable. Then $\mathcal{B}$ is strongly undecidable.*

Proof. (Sketch) By the previous lemma (with L' and $\mathcal{B}$ instead of L and $\mathcal{A}$), we can reduce to the case that we have a 0-definition δ of $\mathcal{A}$ in $\mathcal{B}$. One can show that in this case we have (1) an algorithm that computes for any L-sentence σ an L'-sentence $\delta\sigma$, and (2) a finite set Δ of L'-sentences true in $\mathcal{B}$, with the following properties:

(i) $\mathcal{A} \models \sigma \Longleftrightarrow \mathcal{B} \models \delta\sigma$;

(ii) for each set Σ' of L'-sentences with $\mathcal{B} \models \Sigma'$ and each L-sentence σ,

$$\Sigma \vdash \sigma \Longleftrightarrow \Sigma' \cup \Delta \vdash \delta\sigma, \text{ where } \Sigma := \{s : s \text{ is an } L\text{-sentence and } \Sigma' \cup \Delta \models \delta s\}.$$

Let Σ' be a set of L'-sentences such that $\mathcal{B} \models \Sigma'$; we need to show that $\mathrm{Th}_{L'}(\Sigma')$ is undecidable. Suppose towards a contradiction that $\mathrm{Th}_{L'}(\Sigma')$ is decidable. Then $\mathrm{Th}_{L'}(\Sigma' \cup \Delta)$ is decidable, by part (2) of Lemma 5.8.1, so we have an algorithm for deciding whether any given L'-sentence is provable from $\Sigma' \cup \Delta$. Take Σ as in (ii) above. Then $\mathcal{A} \models \Sigma$ by (i), and by (ii) we obtain an algorithm for deciding whether any given L-sentence is provable from Σ. But $\mathrm{Th}_L(\Sigma)$ is undecidable by assumption, and we have a contradiction. $\square$

Corollary 5.8.4. $\mathrm{Th}(\mathrm{Ri})$ *is undecidable, in other words, the theory of rings is undecidable.*

Proof. It suffices to show that the ring $(\mathbf{Z};\ 0, 1, +, -, \cdot)$ of integers is strongly undecidable. Using Lagrange's theorem that

$$\mathbf{N} = \{a^2 + b^2 + c^2 + d^2 : a, b, c, d \in \mathbf{Z}\},$$

we see that the inclusion map $\mathbf{N} \to \mathbf{Z}$ defines $\mathfrak{N}$ in the ring of integers, so by Tarski's Theorem the ring of integers is strongly undecidable. $\square$

For the same reason, the theory of commutative rings, of integral domains, and more generally, the theory of any class of rings containing the ring of integers is undecidable.

Fact. *The set* $\mathbf{Z} \subseteq \mathbf{Q}$ *is* 0*-definable in the field* $(\mathbf{Q};\ 0, 1, +, -, \cdot)$ *of rational numbers, and thus the ring of integers is definable in the field of rational numbers.*

We shall take this here on faith. The only known proof uses non-trivial results about quadratic forms, and is due to Julia Robinson.

Corollary 5.8.5. *The theory* Th(Fl) *of fields is undecidable. The theory of any class of fields containing the field of rational numbers is undecidable.*

Exercises.

(1) Argue informally, using the Church-Turing Thesis, that Th(ACF) is decidable. You can use the fact that ACF has QE.

Below we let a, b, c denote integers. We say that a *divides* b (notation: $a \mid b$) if $ax = b$ for some integer x, and we say that c is a *least common multiple* of a and b if $a \mid c$, $b \mid c$, and $c \mid x$ for every integer x such that $a \mid x$ and $b \mid x$. Recall that if a and b are not both zero, then they have a unique positive least common multiple, and that if a and b are coprime (that is, there is no integer $x > 1$ with $x \mid a$ and $x \mid b$), then they have ab as a least common multiple.

(2) The structure $(\mathbf{Z};\ 0, 1, +, |)$ is strongly undecidable, where $|$ is the binary relation of divisibility on $\mathbf{Z}$. Hint: Show that if $b + a$ is a least common multiple of a and $a+1$, and $b-a$ is a least common multiple of a and $a-1$, then $b = a^2$. Use this to define the squaring function in $(\mathbf{Z};\ 0, 1, +, |)$, and then show that multiplication is 0-definable in $(\mathbf{Z};\ 0, 1, +, |)$.

(3) Consider the group G of bijective maps $\mathbf{Z} \to \mathbf{Z}$, with composition as the group multiplication. Then G (as a model of Gr) is strongly undecidable. Hint: let s be the element of G given by $s(x) = x + 1$. Check that if $g \in G$ commutes with s, then $g = s^a$ for some a. Next show that

$$a \mid b \iff s^b \text{ commutes with each } g \in G \text{ that commutes with } s^a.$$

Use these facts to specify a definition of the structure $(\mathbf{Z};\ 0, 1, +, |)$ in the group G.

Thus by Tarski's theorem, the theory of groups is undecidable. In fact, the theory of any class of groups that includes the group G of the exercise above is undecidable. On the other hand, Th(Ab), the theory of abelian groups, is known to be decidable (Szmielew).

(4) Let $L = \{F\}$ have just a binary function symbol. Then predicate logic in L (that is, $\mathrm{Th}_L(\emptyset)$) is undecidable.

It can also be shown that predicate logic in the language whose only symbol is a binary relation symbol is undecidable. On the other hand, predicate logic in the language whose only symbol is a unary function symbol is decidable.

To do?

- improve titlepage
- improve or delete index
- more exercises (from homework, exams)
- footnotes pointing to alternative terminology, etc.
- $\square$ at the end of results without proof?
- brief discussion on P=NP in connection with propositional logic
- section(s) on boolean algebra, including Stone representation, Lindenbaum-Tarski algebras, etc.
- section on equational logic? (boolean algebras, groups, as examples)
- solution to a problem by Erdös via compactness theorem, and other simple applications of compactness
- include "equality theorem",
- translation of one language in another (needed in connection with Tarski theorem in last section)
- more details on back-and-forth in connection with unnested formulas
- extra elementary model theory (universal classes, model-theoretic criteria for qe, etc., application to ACF, maybe extra section on RCF, Ax's theorem.
- On computability: a few extra results on c.e. sets, and exponential diophantine result.
- basic framework for many-sorted logic.