Chapter 2

First-Order Logic

Mathematics and some other disciplines such as computer science often consider domains of individuals in which certain relations and operations are singled out. When using the language of propositional logic, our ability to talk about the properties of such relations and operations is very limited. Thus, it is necessary to refine our linguistic means of expression, in order to procure new possibilities of description. To this end, one needs not only logical symbols but also variables for the individuals of the domain being considered, as well as a symbol for equality and symbols for the relations and operations in question. First-order logic, sometimes called also predicate logic, is the part of logic that subjects properties of such relations and operations to logical analysis.

Linguistic particles such as "for all" and "there exists" (called *quantifiers*) play a central role here, whose analysis should be based on a well prepared semantic background. Hence, we first consider mathematical structures and classes of structures. Some of these are relevant both to logic (in particular model theory) and to computer science. Neither the newcomer nor the advanced student needs to read all of **2.1**, with its mathematical flavor, at once. The first five pages should suffice. The reader may continue with **2.2** and later return to what is needed.

Next we home in on the most important class of formal languages, the *first-order* languages, also called *elementary languages.* Their main characteristic is a restriction of the quantification possibilities. We discuss in detail the semantics of these languages and arrive at a notion of *logical consequence* from arbitrary premises. In this context, the notion of a formalized theory is made more precise.

W. Rautenberg, *A Concise Introduction to Mathematical Logic*, Universitext, DOI 10.1007/978-1-4419-1221-3_2,

Finally, we treat the introduction of new notions by explicit definitions and other expansions of a language, for instance by Skolem functions. Not until Chapter **3** do we talk about methods of formal logical deduction. While a multitude of technical details have to be considered in this chapter, nothing is especially profound. Anyway, most of it is important for the undertakings of the subsequent chapters.

2.1 Mathematical Structures

By a *structure* $\mathcal{A}$ we understand a nonempty set A together with certain distinguished relations and operations of A, as well as certain constants distinguished therein. The set A is also termed the *domain* of $\mathcal{A}$, or its *universe*. The distinguished relations, operations, and constants are called the *(basic) relations, operations, and constants of* $\mathcal{A}$. A *finite structure* is one with a finite domain. An easy example is $(\{0,1\}, \wedge, \vee, \neg)$. Here $\wedge$, $\vee$, $\neg$ have their usual meanings on the domain $\{0,1\}$, and no distinguished relations or constants occur. An *infinite structure* has an infinite domain. $\mathcal{A} = (\mathbb{N}, <, +, \cdot, 0, 1)$ is an example with the domain $\mathbb{N}$; here $<$, $+$, $\cdot$, 0, 1 have again their ordinary meaning.

Without having to say so every time, for a structure $\mathcal{A}$ the corresponding letter A will always denote the domain of $\mathcal{A}$; similarly B denotes the domain of $\mathcal{B}$, etc. If $\mathcal{A}$ contains no operations or constants, then $\mathcal{A}$ is also called a *relational structure*. If $\mathcal{A}$ has no relations it is termed an *algebraic structure*, or simply an *algebra*. For example, $(\mathbb{Z}, <)$ is a relational structure, whereas $(\mathbb{Z}, +, 0)$ is an algebraic structure, the *additive group* $\mathbb{Z}$ (it is customary to use here the symbol $\mathbb{Z}$ as well). Also the set of propositional formulas from **1.1** can be understood as an algebra, equipped with the operations $(\alpha, \beta) \mapsto (\alpha \wedge \beta)$, $(\alpha, \beta) \mapsto (\alpha \vee \beta)$, and $\alpha \mapsto \neg\alpha$. Thus, one may speak of the *formula algebra* $\mathcal{F}$ whenever it is useful to do so.

Despite our interest in specific structures, whole classes of structures are also often considered, for instance the classes of groups, rings, fields, vector spaces, Boolean algebras, and so on. Even when initially just a single structure is viewed, call it the paradigm structure, one often needs to talk about similar structures in the same breath, in *one* language, so to speak. This can be achieved by setting aside the concrete meaning of the relation and operation symbols in the paradigm structure and considering

the symbols in themselves, creating thereby a formal language that enables one to talk at once about all structures relevant to a topic. Thus, one distinguishes in this context clearly between denotation and what is denoted. To emphasize this distinction, for instance for $\mathcal{A} = (A, +, <, 0)$, it is better to write $\mathcal{A} = (A, +^{\mathcal{A}}, <^{\mathcal{A}}, 0^{\mathcal{A}})$, where $+^{\mathcal{A}}$, $<^{\mathcal{A}}$, and $0^{\mathcal{A}}$ mean the relation, operation, and constant denoted by $+$, $<$, and 0 in $\mathcal{A}$. Only if it is clear from the context what these symbols denote may the superscripts be omitted. In this way we are free to talk on the one hand about the structure $\mathcal{A}$, and on the other hand about the symbols $+, <, 0$.

A finite or infinite set L resulting in this way, consisting of relation, operation, and constant symbols of a given arity, is called an *extralogical signature*. For the class of all groups (see page 47), $L = \{\circ, e\}$ exemplifies a favored signature; that is, one often considers groups as structures of the form $(G, \circ, e)$, where $\circ$ denotes the group operation and e the unit element. But one can also define groups as structures of the signature $\{\circ\}$, because e is definable in terms of $\circ$, as we shall see later. Of course, instead of $\circ$, another operation symbol could be chosen such as $\cdot$, $*$, or $+$. The latter is mainly used in connection with commutative groups. In this sense, the actual appearance of a symbol is less important; what matters is its arity. $r \in L$ always means that r is a relation symbol, and $f \in L$ that f is an operation symbol, each time of some arity $n > 0$, which of course depends on the symbols r and f, respectively.[1]

An *L-structure* is a pair $\mathcal{A} = (A, L^{\mathcal{A}})$, where $L^{\mathcal{A}}$ contains for every $r \in L$ a relation $r^{\mathcal{A}}$ on A of the same arity as r, for every $f \in L$ an operation $f^{\mathcal{A}}$ on A of the arity of f, and for every $c \in L$ a constant $c^{\mathcal{A}} \in A$. We may omit the superscripts, provided it is clear from the context which operation or relation on A is meant. We occasionally shorten also the notation of structures. For instance, we sometimes speak of the ring $\mathbb{Z}$ or the field $\mathbb{R}$ provided there is no danger of misunderstanding.

Every structure is an L-structure for a certain signature, namely that consisting of the symbols for its relations, functions, and constants. But this does not make the name L-structure superfluous. Basic concepts,

[1] Here r and f represent the general case and look different in a concrete situation. Relation symbols are also called predicate symbols, in particular in the unary case, and operation symbols are sometimes called function symbols. In special contexts, we also admit $n = 0$, regarding constants as 0-ary operations.

such as isomorphism and substructure, each refer to structures of the same signature. From **2.2** on, once the first-order language $\mathcal{L}$ belonging to L has been defined, L-structures will mostly be called $\mathcal{L}$-structures. We then also often say that r, f, or c belongs to $\mathcal{L}$ instead of L.

If $A \subseteq B$ and f is an n-ary operation on B then A is *closed* under f, briefly *f-closed*, if $f\vec{a} \in A$ for all $\vec{a} \in A^n$. If $n = 0$, i.e., if f is a constant c, this simply means $c \in A$. The intersection of any nonempty family of f-closed subsets of B is itself f-closed. Accordingly, we can talk of the smallest (the intersection) of all f-closed subsets of B that contain a given subset $E \subseteq B$. All of this extends in a natural way if f is here replaced by an arbitrary family of operations of B.

Example. For a given positive m, the set $m\mathbb{Z} := \{m \cdot n \mid n \in \mathbb{Z}\}$ of integers divisible by m is closed in $\mathbb{Z}$ under $+$, $-$, and $\cdot$, and is in fact the smallest such subset of $\mathbb{Z}$ containing m.

The *restriction* of an n-ary relation $r^B \subseteq B^n$ to a subset $A \subseteq B$ is $r^A = r^B \cap A^n$. For instance, the restriction of the standard order of $\mathbb{R}$ to $\mathbb{N}$ is the standard order of $\mathbb{N}$. Only because of this fact can the same symbol be used to denote these relations. The restriction f^A of an operation f^B on B to a set $A \subseteq B$ is defined analogously whenever A is f-closed. Simply let $f^A\vec{a} = f^B\vec{a}$ for $\vec{a} \in A^n$. For instance, addition in $\mathbb{N}$ is the restriction of addition in $\mathbb{Z}$ to $\mathbb{N}$, or addition in $\mathbb{Z}$ is an extension of this operation in $\mathbb{N}$. Again, only this state of affairs allows us to denote the two operations by the same symbol.

Let $\mathcal{B}$ be an L-structure and let $A \subseteq B$ be nonempty and closed under all operations of $\mathcal{B}$; this will be taken to include $c^{\mathcal{B}} \in A$ for constant symbols $c \in L$. To such a subset A corresponds in a natural way an L-structure $\mathcal{A} = (A, L^{\mathcal{A}})$, where $r^{\mathcal{A}}$ and $f^{\mathcal{A}}$ for $r, f \in L$ are the restrictions of $r^{\mathcal{B}}$ respectively $f^{\mathcal{B}}$ to A. Finally, let $c^{\mathcal{A}} = c^{\mathcal{B}}$ for $c \in L$. The structure $\mathcal{A}$ so defined is then called a *substructure* of $\mathcal{B}$, and $\mathcal{B}$ is called an *extension* of $\mathcal{A}$, in symbols $\mathcal{A} \subseteq \mathcal{B}$. This is a certain abuse of $\subseteq$ but it does not cause confusion, since the arguments indicate what is meant.

$\mathcal{A} \subseteq \mathcal{B}$ implies $A \subseteq B$ but not conversely, in general. For example, $\mathcal{A} = (\mathbb{N}, <, +, 0)$ is a substructure of $\mathcal{B} = (\mathbb{Z}, <, +, 0)$ since $\mathbb{N}$ is closed under addition in $\mathbb{Z}$ and 0 has the same meaning in $\mathcal{A}$ and $\mathcal{B}$. Here we dropped the superscripts for $<$, $+$, and 0 because there is no risk of misunderstanding.

A nonempty subset G of the domain B of a given L-structure $\mathcal{B}$ defines a smallest substructure $\mathcal{A}$ of $\mathcal{B}$ containing G. The domain of $\mathcal{A}$ is the smallest subset of B containing G and closed under all operations of B. $\mathcal{A}$ is called the substructure *generated from* G *in* $\mathcal{B}$. For instance, $3\mathbb{N}$ $(= \{3n \mid n \in \mathbb{N}\})$ is the domain of the substructure generated from $\{3\}$ in $(\mathbb{N}, +, 0)$, since $3\mathbb{N}$ contains 0 and 3, is closed under $+$, and is clearly the smallest such subset of $\mathbb{N}$. A structure $\mathcal{A}$ is called *finitely generated* if for some finite $G \subseteq A$ the substructure generated from G in $\mathcal{A}$ coincides with $\mathcal{A}$. For instance, $(\mathbb{Z}, +, -, 0)$ is finitely generated by $G = \{1\}$.

If $\mathcal{A}$ is an L-structure and $L_0 \subseteq L$ then the L_0-structure $\mathcal{A}_0$ with domain A and where $\mathsf{s}^{\mathcal{A}_0} = \mathsf{s}^{\mathcal{A}}$ for all symbols $\mathsf{s} \in L_0$ is termed the L_0-*reduct of* $\mathcal{A}$, and $\mathcal{A}$ is called an L-*expansion* of $\mathcal{A}_0$. For instance, the group $(\mathbb{Z}, +, 0)$ is the $\{+, 0\}$-reduct of the ordered ring $(\mathbb{Z}, <, +, \cdot, 0)$. The notions reduct and substructure must clearly be distinguished. A reduct of $\mathcal{A}$ has always the same domain as $\mathcal{A}$, while the domain of a substructure of $\mathcal{A}$ is as a rule a proper subset of A.

Below we list some frequently cited properties of a binary relation $\lhd$ in a set A. It is convenient to write $a \lhd b$ instead of $(a, b) \in \lhd$, and $a \not\lhd b$ for $(a, b) \notin \lhd$. Just as $a < b < c$ often stands for $a < b \;\&\; b < c$, we write $a \lhd b \lhd c$ for $a \lhd b \;\&\; b \lhd c$. In the listing below, 'for all a' and 'there exists an a' respectively mean 'for all $a \in A$' and 'there exists some $a \in A$'. The relation $\lhd \subseteq A^2$ is called

reflexive	if $a \lhd a$ for all a,
irreflexive	if $a \not\lhd a$ for all a,
symmetric	if $a \lhd b \;\Rightarrow\; b \lhd a$, for all a, b,
antisymmetric	if $a \lhd b \lhd a \;\Rightarrow\; a = b$, for all a, b,
transitive	if $a \lhd b \lhd c \;\Rightarrow\; a \lhd c$, for all a, b, c,
connex	if $a = b$ or $a \lhd b$ or $b \lhd a$, for all a, b.

Reflexive, transitive, and symmetric relations are also called *equivalence relations*. These are often denoted by $\sim$, $\approx$, $\equiv$, $\simeq$, or similar symbols. Such a relation generates a partition of its domain whose parts, consisting of mutually equivalent elements, are called *equivalence classes*.

We now present an overview of classes of structures to which we will later refer, mainly in Chapter **5**. Hence, for the time being, the beginner may skip the following and jump to **2.2**.

1. Graphs, partial orders, and orders. A relational structure $(A, \lhd)$ with some relation $\lhd \subseteq A^2$ is often termed a (directed) *graph*. If $\lhd$ is irreflexive and transitive we usually write $<$ for $\lhd$ and speak of a (strict) *partial order* or a *partially ordered set*, also called a *poset* for short. If we define $x \leqslant y$ by $x < y$ or $x = y$, then $\leqslant$ is reflexive, transitive, and antisymmetric, called a *reflexive partial order*, the one that belongs to $<$. If one starts with a reflexive partial order on A and defines $x < y$ by $x \leqslant y \;\&\; x \neq y$, then $(A, <)$ is clearly a poset.

A connex partial order $\mathcal{A} = (A, <)$ is called a *total* or *linear* order, also termed an *ordered* or a *strictly ordered* set. $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ are examples with respect to their standard orders. Here we follow the traditional habit of referring to ordered sets by their domains only.

Let U be a nonempty subset of some ordered set A such that for all $a, b \in A$, $a < b \in U \Rightarrow a \in U$. Such a U is called an *initial segment* of A. In addition, let $V := A \backslash U \neq \emptyset$. Then the pair (U, V) is called a *cut*. The cut is said to be a *gap* if U has no largest and V no smallest element. However, if U *has* a largest element a, and V a smallest element b, then (U, V) is called a *jump*. b is in this case called *the immediate successor* of a, and a *the immediate predecessor* of b, because then there is no element from A between a and b. An infinite ordered set without gaps and jumps, like $\mathbb{R}$, is said to be *continuously ordered*. Such a set is easily seen to be *densely ordered*, i.e., between any two elements lies another one.

A totally ordered subset K of a partially ordered set H is called a *chain* in H. Such a K is said to be *bounded* (to the above) if there is some $b \in H$ with $a \leqslant b$ for all $a \in K$. Call $c \in H$ *maximal* in H if no $a \in H$ exists with $a > c$. An infinite partial order need not have a maximal element, nor need all chains be bounded, as is seen by the example $(\mathbb{N}, <)$. With these notions, a basic mathematical tool can now be stated:

Zorn's lemma. *If every chain in a nonempty poset H is bounded then H has a maximal element.*

A (totally) ordered set A is *well-ordered* if every nonempty subset of A has a smallest element; equivalently, there are no infinite decreasing sequences $a_0 > a_1 > \cdots$ of elements from A. Clearly, every finite ordered set is well-ordered. The simplest example of an infinite well-ordered set is $\mathbb{N}$ together with its standard order.

2. Groupoids, semigroups, and groups. Algebras $\mathcal{A} = (A, \circ)$ with an operation $\circ\colon A^2 \to A$ are termed *groupoids*. If $\circ$ is associative then $\mathcal{A}$ is called a *semigroup*, and if $\circ$ is additionally invertible, then $\mathcal{A}$ is said to be a *group*. It is provable that a group $(G, \circ)$ in this sense contains exactly one *unit element*, that is, an element e such that $x \circ e = e \circ x = x$ for all $x \in G$, also called a *neutral element*. A well-known example is the group of bijections of a set M. If the group operation $\circ$ is commutative, we speak of a *commutative* or *abelian* group.

Here are some examples of semigroups that are not groups: (a) the set of strings on some alphabet A with respect to concatenation, the *word-semigroup* or *free semigroup generated from* A. (b) the set M^M of mappings from M to itself with respect to composition. (c) $(\mathbb{N}, +)$ and $(\mathbb{N}, \cdot)$; these two are commutative semigroups. With the exception of $(M^M, \circ)$, all mentioned examples of semigroups are *regular*, which is to mean $x \circ y = x \circ z \;\Rightarrow\; y = z$, and $x \circ z = y \circ z \;\Rightarrow\; x = y$, for all x, y, z.

Substructures of semigroups are again semigroups. Substructures of groups are in general only semigroups, as seen from $(\mathbb{N}, +) \subseteq (\mathbb{Z}, +)$. Not so in the signature $\{\circ, e, {}^{-1}\}$, where e denotes the unit element and x^{-1} the inverse of x. Here all substructures are indeed subgroups. The reason is that in $\{\circ, e, {}^{-1}\}$, the group axioms can be written as universally quantified equations, where for brevity, we omit the writing of "for all x, y, z," namely as $x \circ (y \circ z) = (x \circ y) \circ z$, $x \circ e = x$, $x \circ x^{-1} = e$. These equations certainly retain their validity in the transition to substructures. We mention that from the last three equations, $e \circ x = x$ and $x^{-1} \circ x = e$ are derivable, although $\circ$ is not supposed to be commutative.

Ordered semigroups and groups possess along with $\circ$ some order, with respect to which $\circ$ is monotonic in both arguments, like $(\mathbb{N}, +, 0, \leqslant)$. A commutative ordered semigroup $(A, +, 0, \leqslant)$ with zero element 0, which at the same time is the smallest element in A, and where $a \leqslant b$ iff there is some c with $a + c = b$, is called a *domain of magnitude*. Everyday examples are the domains of *length, mass, money*, etc.

3. Rings and fields. These belong to the most commonly known structures. Below we list the axioms for the theory T_F of fields in $+, \cdot, 0, 1$. A *field* is a model of T_F. A *ring* is a model of the axiom system T_R for rings that derives from T_F by dropping the constant 1 from the signature and the axioms $\mathrm{N}^\times$, $\mathrm{C}^\times$, and $\mathrm{I}^\times$ from T_F. Here are the axioms of T_F:

$$\begin{array}{lll} \mathrm{N}^+: & x+0=x & \mathrm{N}^\times: \quad x\cdot 1=x \\ \mathrm{C}^+: & x+y=y+x & \mathrm{C}^\times: \quad x\cdot y=y\cdot x \\ \mathrm{A}^+: & (x+y)+z=x+(y+z) & \mathrm{A}^\times: \quad (x\cdot y)\cdot z=x\cdot(y\cdot z) \\ \mathrm{D}: & x\cdot(y+z)=x\cdot y+x\cdot z & \mathrm{D}': \quad (y+z)\cdot x=y\cdot x+z\cdot x \\ \mathrm{I}^+: & \forall x\exists y\, x+y=0 & \mathrm{I}^\times: \quad 0\neq 1\wedge(\forall x\neq 0)\exists y\, x\cdot y=1 \end{array}$$

In view of $\mathrm{C}^\times$, axiom D' is dispensable for T_F but not for T_R. When removing I^+ from T_R, we obtain the theory of *semirings*. A well-known example is $(\mathbb{N},+,\cdot,0)$. A commutative ring that has a unit element 1 but no *zero-divisor* (i.e., $\neg\exists x\exists y(x,y\neq 0\wedge x\cdot y=0)$ is called an *integral domain*. A typical example is $(\mathbb{Z},+,\cdot,0,1)$.

Let $\mathcal{K},\mathcal{K}'$ be any fields with $\mathcal{K}\subset\mathcal{K}'$. We call $a\in K'\setminus K$ *algebraic* or *transcendental* on $\mathcal{K}$, depending on whether a is a zero of a polynomial with coefficients in K or not. If every polynomial of degree $\geqslant 1$ with coefficients in K breaks down into linear factors, as is the case for the field of complex numbers, then $\mathcal{K}$ is called *algebraically closed*, in short, $\mathcal{K}$ is a.c. These fields will be more closely inspected in **3.3** and Chapter **5**. Each field $\mathcal{K}$ has a smallest subfield $\mathcal{P}$, called a *prime field*. One says that $\mathcal{K}$ *has characteristic* 0 or p (a prime number), depending on whether $\mathcal{P}$ is isomorphic to the field $\mathbb{Q}$ or the finite field of p elements. No other prime fields exist. It is not hard to show that $\mathcal{K}$ has the characteristic p iff the sentence $\mathtt{char}_p: \underbrace{1+\cdots+1}_{p}=0$ holds in $\mathcal{K}$.

Rings, fields, etc. may also be *ordered*, whereby the usual monotonicity laws are required. For example, $(\mathbb{Z},<,+,\cdot,0,1)$ is the *ordered ring* of integers and $(\mathbb{N},<,+,\cdot,0,1)$ the *ordered semiring* of natural numbers.

4. Semilattices and lattices. $\mathcal{A}=(A,\circ)$ is called a *semilattice* if $\circ$ is associative, commutative, and idempotent. An example is $(\{0,1\},\circ)$ with $\circ=\wedge$. If we define $a\leqslant b:\Leftrightarrow a\circ b=a$ then $\leqslant$ is a reflexive partial order on A. Reflexivity holds, since $a\circ a=a$. As can be easily verified, $a\circ b$ is in fact the *infimum* of a,b with respect to $\leqslant$, $a\circ b=\inf\{a,b\}$, that is, $a\circ b\leqslant a,b$, and $c\leqslant a,b\Rightarrow c\leqslant a\circ b$, for all $a,b,c\in A$.

$\mathcal{A}=(A,\cap,\cup)$ is called a *lattice* if $(A,\cap)$ and $(A,\cup)$ are both semilattices and the following so-called *absorption laws* hold: $a\cap(a\cup b)=a$ and $a\cup(a\cap b)=a$. These imply $a\cap b=a\Leftrightarrow a\cup b=b$. As above, $a\leqslant b:\Leftrightarrow a\cap b=a$ defines a partial order such that $a\cap b=\inf\{a,b\}$.

In addition, one has $a \cup b = \sup\{a, b\}$ (the *supremum* of a, b), which is to mean $a, b \leqslant a \cup b$, and $a, b \leqslant c \;\Rightarrow\; a \cup b \leqslant c$, for all $c \in A$. If $\mathcal{A}$ satisfies, moreover, the *distributive laws* $x \cap (y \cup c) = (x \cap y) \cup (x \cap c)$ and $x \cup (y \cap c) = (x \cup y) \cap (x \cup c)$, then $\mathcal{A}$ is termed a *distributive* lattice. For instance, the power set $\mathfrak{P}M$ with the operations $\cap$ and $\cup$ for $\cap$ and $\cup$ respectively is a distributive lattice, as is every nonempty family of subsets of M closed under $\cap$ and $\cup$, a so-called *lattice of sets*. Another important example is $(\mathbb{N}, \gcd, \mathrm{lcm})$. Here $\gcd(a, b)$ and $\mathrm{lcm}(a, b)$ denote the greatest common divisor and the least common multiple of $a, b \in \mathbb{N}$.

5. Boolean algebras. An algebra $\mathcal{A} = (A, \cap, \cup, \neg)$ where $(A, \cap, \cup)$ is a distributive lattice and in which at least the equations

$$\neg\neg x = x, \quad \neg(x \cap y) = \neg x \cup \neg y, \quad x \cap \neg x = y \cap \neg y$$

are valid is called a *Boolean algebra*. The paradigm structure is the two-element Boolean algebra $\mathbf{2} := (\{0, 1\}, \wedge, \vee, \neg)$, with $\cap, \cup$ interpreted as $\wedge, \vee$, respectively. One defines the constants 0 and 1 by $0 := a \cap \neg a$ for any $a \in A$ and $1 := \neg 0$. There are many ways to characterize Boolean algebras $\mathcal{A}$, for instance, by saying that $\mathcal{A}$ satisfies all equations valid in $\mathbf{2}$. The signature can also be variously selected. For example, the signature $\wedge, \vee, \neg$ is well suited to deal algebraically with two-valued propositional logic. Terms of this signature are, up to the denotation of variables, precisely the Boolean formulas from **1.1**, and a valid logical equivalence $\alpha \equiv \beta$ corresponds to the equation $\alpha = \beta$, valid in $\mathbf{2}$. Further examples of Boolean algebras are the *algebras of sets* $\mathcal{A} = (A, \cap, \cup, \neg)$. Here A consists of a nonempty system of subsets of a set I, closed under $\cap$, $\cup$, and $\neg$ (complementation in I). These are the most general examples; a famous theorem, *Stone's representation theorem*, says that each Boolean algebra is isomorphic to an algebra of sets.

6. Logical L-matrices. These are structures $\mathcal{A} = (A, L^{\mathcal{A}}, D^{\mathcal{A}})$, where L contains only operation symbols (the "logical" symbols) and D denotes a unary predicate, *the set of distinguished values* of $\mathcal{A}$. Best known is the two-valued *Boolean matrix* $\mathcal{B} = (\mathbf{2}, D^{\mathcal{B}})$ with $D^{\mathcal{B}} = \{1\}$. The consequence relation $\models_{\mathcal{A}}$ in the propositional language $\mathcal{F}$ of signature L is defined as in the two-valued case: Let $X \subseteq \mathcal{F}$ and $\varphi \in \mathcal{F}$. Then $X \models_{\mathcal{A}} \varphi$ if $w\varphi \in D^{\mathcal{A}}$ for every $w : PV \to A$ with $wX \subseteq D^{\mathcal{A}}$ ($wX := \{w\alpha \mid \alpha \in X\}$). In words, if the values of all $\alpha \in X$ are distinguished, then so too is the value of φ.

Homomorphisms and isomorphisms. The following notions are important for both mathematical and logical investigations. Much of the material presented here will be needed in Chapter **5**. In the following definition, n (>0) denotes as always the arity of f or r.

Definition. Let $\mathcal{A}, \mathcal{B}$ be L-structures and $h\colon \mathcal{A} \to \mathcal{B}$ (strictly speaking $h\colon A \to B$) a mapping such that for all $f, c, r \in L$ and $\vec{a} \in A^n$,

$$\text{(H):}\quad hf^{\mathcal{A}}\vec{a} = f^{\mathcal{B}}h\vec{a},\ hc^{\mathcal{A}} = c^{\mathcal{B}},\ r^{\mathcal{A}}\vec{a} \Rightarrow r^{\mathcal{B}}h\vec{a} \quad \big(h\vec{a} = (ha_1, \ldots, ha_n)\big).$$

Then h is called a *homomorphism*. If the third condition in (H) is replaced by the stronger condition (S): $(\exists \vec{b} \in A^n)(h\vec{a} = h\vec{b}\ \&\ r^{\mathcal{A}}\vec{b}) \Leftrightarrow r^{\mathcal{B}}h\vec{a}$ [2] then h is said to be a *strong homomorphism* (for algebras, the word "strong" is dispensable). An injective strong homomorphism $h\colon \mathcal{A} \to \mathcal{B}$ is called an *embedding* of $\mathcal{A}$ into $\mathcal{B}$. If, in addition, h is bijective then h is called an *isomorphism*, and in case $\mathcal{A} = \mathcal{B}$, an *automorphism*.

An embedding or isomorphism $h\colon \mathcal{A} \to \mathcal{B}$ satisfies $r^{\mathcal{A}}\vec{a} \Leftrightarrow r^{\mathcal{B}}h\vec{a}$. Indeed, since $h\vec{a} = h\vec{b} \Leftrightarrow \vec{a} = \vec{b}$, (S) yields $r^{\mathcal{B}}h\vec{a} \Rightarrow (\exists \vec{b} \in A^n)(\vec{a} = \vec{b}\ \&\ r^{\mathcal{A}}\vec{b}) \Rightarrow r^{\mathcal{A}}\vec{a}$. $\mathcal{A}, \mathcal{B}$ are said to be *isomorphic*, in symbols $\mathcal{A} \simeq \mathcal{B}$, if there is an isomorphism from $\mathcal{A}$ to $\mathcal{B}$. It is readily verified that $\simeq$ is reflexive, symmetric, and transitive, hence an equivalence relation on the class of all $\mathcal{L}$-structures.

Examples 1. (a) A valuation w considered in **1.1** can be regarded as a homomorphism of the propositional formula algebra $\mathcal{F}$ into the two-element Boolean algebra $\mathit{2}$. Such a $w\colon \mathcal{F} \to \mathit{2}$ is necessarily onto.

(b) Let $\mathcal{A} = (A, *)$ be a word semigroup with the concatenation operation $*$ and $\mathcal{B}$ the additive semigroup of natural numbers, considered as L-structures for $L = \{\circ\}$ with $\circ^{\mathcal{A}} = *$ and $\circ^{\mathcal{B}} = +$. Let $\mathrm{lh}(\xi)$ denote the length of a word or string $\xi \in A$. Then $\xi \mapsto \mathrm{lh}(\xi)$ is a homomorphism since $\mathrm{lh}(\xi * \eta) = \mathrm{lh}(\xi) + \mathrm{lh}(\eta)$, for all $\xi, \eta \in A$. If $\mathcal{A}$ is generated from a single letter, lh is evidently bijective, hence an isomorphism.

(c) The mapping $a \mapsto (a, 0)$ from $\mathbb{R}$ to $\mathbb{C}$ (= set of complex numbers, understood as ordered pairs of real numbers) is a good example of an embedding of the field $\mathbb{R}$ into the field $\mathbb{C}$. Nonetheless, in this case, we are used to saying that $\mathbb{R}$ is a subfield of $\mathbb{C}$, and that $\mathbb{R}$ is a subset of $\mathbb{C}$.

[2] $(\exists \vec{b} \in A^n)(h\vec{a} = h\vec{b}\ \&\ r^{\mathcal{A}}\vec{b})$ abbreviates 'there is some $\vec{b} \in A^n$ with $h\vec{a} = h\vec{b}$ and $r^{\mathcal{A}}\vec{b}$'. If $h\colon \mathcal{A} \to \mathcal{B}$ is onto (and only this case will occur in our applications) then (S) is equivalent to the more suggestive condition $r^{\mathcal{B}} = \{h\vec{a} \mid r^{\mathcal{A}}\vec{a}\}$.

(d) Let $\mathcal{A} = (\mathbb{R}, +, <)$ be the ordered additive group of real numbers and $\mathcal{B} = (\mathbb{R}_+, \cdot, <)$ the multiplicative group of positive reals. Then for any $b \in \mathbb{R}_+ \setminus \{1\}$ there is precisely one isomorphism $\eta\colon \mathcal{A} \to \mathcal{B}$ such that $\eta 1 = b$, namely $\eta\colon x \mapsto b^x$, the exponential function $\exp_b$ to the base b. It is even possible to *define* $\exp_b$ as this isomorphism, by first proving that—up to isomorphism—there is only one continuously ordered abelian group (first noticed in [Ta2] though not explicitly put into words).
(e) The algebras $\mathcal{A} = (\{0,1\}, +)$ and $\mathcal{B} = (\{0,1\}, \leftrightarrow)$ are only apparently different, but are in fact isomorphic, with the isomorphism δ where $\delta 0 = 1$, $\delta 1 = 0$. Thus, since $\mathcal{A}$ is a group, $\mathcal{B}$ is a group as well, which is not obvious at first glance. By adjoining the unary predicate $D = \{1\}$, $\mathcal{A}$ and $\mathcal{B}$ become (nonisomorphic) logical matrices. These actually define the two "dual" fragmentary two-valued logics for the connectives *either ... or ...*, and ... *if and only if* ..., which have many properties in common.

Congruences. A *congruence relation* (or simply a *congruence*) in a structure $\mathcal{A}$ of signature L is an equivalence relation $\approx$ in A such that for all $n > 0$, all $f \in L$ of arity n, and all $\vec{a}, \vec{b} \in A^n$,

$$\vec{a} \approx \vec{b} \;\Rightarrow\; f^A\vec{a} \approx f^A\vec{b}.$$

Here $\vec{a} \approx \vec{b}$ means $a_i \approx b_i$ for $i = 1, \ldots, n$. A trivial example is the identity in $\mathcal{A}$. If $h\colon \mathcal{A} \to \mathcal{B}$ is a homomorphism then $\approx_h \subseteq A^2$, defined by $a \approx_h b \Leftrightarrow ha = hb$, is a congruence in $\mathcal{A}$, called the *kernel* of h. Let A' be the set of equivalence classes $a/\!\approx := \{x \in A \mid a \approx x\}$ for $a \in A$, also called the *congruence classes* of $\approx$, and set $\vec{a}/\!\approx := (a_1/\!\approx, \ldots, a_n/\!\approx)$ for $\vec{a} \in A^n$. Define $f^{A'}(\vec{a}/\!\approx) := (f^A\vec{a})/\!\approx$ and let $r^{A'}\vec{a}/\!\approx :\Leftrightarrow (\exists \vec{b} \approx \vec{a}) r^A\vec{b}$. These definitions are *sound*, that is, independent of the choice of the n-tuple $\vec{a}$ of representatives. Then A' becomes an L-structure $\mathcal{A}'$, the *factor structure of* $\mathcal{A}$ *modulo* $\approx$, denoted by $\mathcal{A}/\!\approx$. Interesting, in particular for Chapter **5**, is the following very general and easily provable

Homomorphism theorem. *Let $\mathcal{A}$ be L-structure and $\approx$ a congruence in $\mathcal{A}$. Then $k\colon a \mapsto a/\!\approx$ is a strong homomorphism from $\mathcal{A}$ onto $\mathcal{A}/\!\approx$, the canonical homomorphism. Conversely, if $h\colon \mathcal{A} \to \mathcal{B}$ is a strong homomorphism from $\mathcal{A}$ onto an $\mathcal{L}$-structure $\mathcal{B}$ with kernel $\approx$ then $\imath : a/\!\approx \mapsto ha$ is an isomorphism from $\mathcal{A}/\!\approx$ to $\mathcal{B}$, and $h = \imath \circ k$.*

Proof. We omit here the superscripts for f and r just for the sake of legibility. Clearly, $kf\vec{a} = (f\vec{a})/\!\approx = f(\vec{a}/\!\approx) = fk\vec{a}$ $\big(= f(ka_1, \ldots, ka_n)\big)$,

and $(\exists \vec{b}\in A^n)(k\vec{a} = k\vec{b}\ \&\ r\vec{b}) \Leftrightarrow (\exists \vec{b}\approx\vec{a})r\vec{b} \Leftrightarrow r\,\vec{a}/{\approx} \Leftrightarrow r\,k\vec{a}$ by definition. Hence k is what we claimed. The definition of $\imath$ is sound, and $\imath$ is bijective since $ha = hb \Rightarrow a/{\approx} = b/{\approx}$. Furthermore, $\imath$ is an isomorphism because

$$\imath f(\vec{a}/{\approx}) = hf\vec{a} = fh\vec{a} = f\imath(\vec{a}/{\approx}) \text{ and } r\,\vec{a}/{\approx} \Leftrightarrow r\,h\vec{a} \Leftrightarrow r\,\imath(\vec{a}/{\approx}).$$

Finally, h is the composition $\imath \circ k$ by the definitions of $\imath$ and k. ❏

Remark. For algebras $\mathcal{A}$, this theorem is the usual homomorphism theorem of universal algebra. $\mathcal{A}/{\approx}$ is then named the *factor algebra*. The theorem covers groups, rings, etc. In groups, the kernel of a homomorphism is already determined by the congruence class of the unit element, called a *normal subgroup*, in rings by the congruence class of 0, called an *ideal*. Hence, in textbooks on basic algebra the homomorphism theorem is separately formulated for groups and rings, but is easily derivable from the general theorem present here.

Direct products. These provide the basis for many constructions of new structures, especially in **5.7**. A well-known example is the n-dimensional vector group $(\mathbb{R}^n, 0, +)$. This is the n-fold direct product of the group $(\mathbb{R}, 0, +)$ with itself. The addition in $\mathbb{R}^n$ is defined componentwise, as is also the case in the following

Definition. Let $(\mathcal{A}_i)_{i\in I}$ be a nonempty family of L-structures. The *direct product* $\mathcal{B} = \prod_{i\in I}\mathcal{A}_i$ is the structure defined as follows: Its domain is $B = \prod_{i\in I} A_i$, called the *direct product* of the sets A_i. The elements $a = (a_i)_{i\in I}$ of B are functions defined on I with $a_i \in A_i$ for each $i \in I$. Relations and operations in $\mathcal{B}$ are defined componentwise, that is,

$$r^{\mathcal{B}}\vec{a} \ \Leftrightarrow\ r^{\mathcal{A}_i}\vec{a}_i \text{ for all } i \in I, \quad f^{\mathcal{B}}\vec{a} = (f^{\mathcal{A}_i}\vec{a}_i)_{i\in I}, \quad c^{\mathcal{B}} = (c^{\mathcal{A}_i})_{i\in I},$$

where $\vec{a} = (a^1, \dots, a^n) \in B^n$ (here the superscripts count the components) with $a^\nu := (a^\nu_i)_{i\in I}$ for $\nu = 1, \dots, n$, and $\vec{a}_i := (a^1_i, \dots, a^n_i) \in A^n_i$.

Whenever $\mathcal{A}_i = \mathcal{A}$ for all $i \in I$, then $\prod_{i\in I}\mathcal{A}_i$ is denoted by $\mathcal{A}^I$ and called a *direct power* of the structure $\mathcal{A}$. Note that $\mathcal{A}$ is embedded in $\mathcal{A}^I$ by the mapping $a \mapsto (a)_{i\in I}$, where $(a)_{i\in I}$ denotes the I-tuple with the constant value a, that is, $(a)_{i\in I} = (a, a, \dots)$. For $I = \{1, \dots, m\}$, the product $\prod_{i\in I}\mathcal{A}_i$ is also written as $\mathcal{A}_1 \times \cdots \times \mathcal{A}_m$. If $I = \{0, \dots, n{-}1\}$ one mostly writes $\mathcal{A}^n$ for $\mathcal{A}^I$.

Examples 2. (a) Let $I = \{1, 2\}$, $\mathcal{A}_i = (A_i, <^i)$, and $\mathcal{B} = \prod_{i\in I}\mathcal{A}_i$. Then $a <^{\mathcal{B}} b \Leftrightarrow a_1 <^1 b_1\ \&\ a_2 <^2 b_2$, for all $a, b \in B = A_1 \times A_2$. Note that if $\mathcal{A}_1, \mathcal{A}_2$ are ordered sets then $\mathcal{B}$ is only a partial order. The deeper reason for this observation will become clear in Chapter **5**.

(b) Let $\mathcal{B} = \mathcal{2}^I$ be a direct power of the two-element Boolean algebra $\mathcal{2}$. The elements $a \in B$ are I-tuples of 0 and 1. These uniquely correspond to the subsets of I via the mapping $\imath\colon a \mapsto I_a := \{i \in I \mid a_i = 1\}$. As a matter of fact, $\imath$ is an isomorphism from $\mathcal{B}$ to $(\mathfrak{P}I, \cap, \cup, \neg)$, as can readily be verified; Exercise 4.

Exercises

1. Show that there are (up to isomorphism) exactly five two-element proper groupoids. Here a groupoid $(H, \cdot)$ is termed *proper* if the operation $\cdot$ is essentially binary.

2. $\approx$ $(\subseteq A^2)$ is termed *Euclidean* if $a \approx b \;\&\; a \approx c \;\Rightarrow\; b \approx c$, for all $a, b, c \in A$. Show that $\approx$ is an equivalence relation in A if and only if $\approx$ is reflexive and Euclidean.

3. Prove that an equivalence relation $\approx$ on an algebraic L-structure $\mathcal{A}$ is a congruence iff for all $f \in L$ of arity n, all $i = 1, \dots, n$, and all $a_1, \dots, a_{i-1}, a, a', a_{i+1}, \dots, a_n \in A$ with $a \approx a'$,
$f(a_1, \dots, a_{i-1}, a, a_{i+1}, \dots, a_n) \approx f(a_1, \dots, a_{i-1}, a', a_{i+1}, \dots, a_n)$.

4. Prove in detail that $\mathcal{2}^I \simeq (\mathfrak{P}I, \cap, \cup, \neg)$ for a nonempty index set I. Prove the corresponding statement for any subalgebra of $\mathcal{2}^I$.

5. Show that $h\colon \prod_{i\in I} \mathcal{A}_i \to \mathcal{A}_j$ with $ha = a_j$ is a homomorphism for each $j \in I$.

2.2 Syntax of First-Order Languages

Standard mathematical language enables us to talk precisely about structures, such as the field of real numbers. However, for logical (and metamathematical) issues it is important to delimit the theoretical framework to be considered; this is achieved most simply by means of a formalization. In this way one obtains an *object language*; that is, the formalized elements of the language, such as the components of a structure, are *objects* of our consideration. To formalize interesting properties of a structure in this language, one requires at least variables for the elements of its domain, called *individual variables*. Further are required sufficiently many logical

symbols, along with symbols for the distinguished relations, functions, and constants of the structure. These *extralogical* symbols constitute the signature L of the formal language that we are going to define.

In this manner one arrives at the *first-order languages*, also termed *elementary* languages. Nothing is lost in terms of generality if the set of variables is the same for all elementary languages; we denote this set by *Var* and take it to consist of the countably many symbols $\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots$ Two such languages therefore differ only in the choice of their extralogical symbols. Variables for subsets of the domain are consciously excluded, since languages containing variables both for individuals and sets of these individuals—second-order languages, discussed in **3.8**—have different semantic properties from those investigated here.

We first determine the *alphabet*, the set of *basic symbols* of a first-order language determined by a signature L. It includes, of course, the already specified variables $\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots$ In what follows, these will mostly be denoted by x, y, z, u, v, though sometimes other letters with or without indices may serve the same purpose. The boldface printed original variables are useful in writing down a formula in the variables $\boldsymbol{v}_{i_1}, \ldots, \boldsymbol{v}_{i_n}$, for these can then be denoted, for instance, by $v_1, \ldots, v_n$, or by $x_1, \ldots, x_n$.

Further, the *logical* symbols $\wedge$ (and), $\neg$ (not), $\forall$ (*for all*), the equality sign $\boldsymbol{=}$, and, of course, all extralogical symbols from L should belong to the alphabet. Note that the boldface symbol $\boldsymbol{=}$ is taken as a basic symbol; simply taking $=$ could lead to unintended mix-ups with the metamathematical use of the equality symbol $=$ (in Chapter **4** also *identity-free* languages without $\boldsymbol{=}$ will be considered). Finally, the parentheses (,) are included in the alphabet. Other symbols are introduced by definition, e.g., $\vee$, $\rightarrow$, $\leftrightarrow$ are defined as in **1.4** and the symbols $\exists$ (*there exists*) and $\exists!$ (*there exists exactly one*) will be defined later. Let $\mathcal{S}_{\mathcal{L}}$ denote the set of all strings made up of symbols that belong to the alphabet of $\mathcal{L}$.

From the set $\mathcal{S}_{\mathcal{L}}$ of all strings we pick out the meaningful ones, namely terms and formulas, according to certain rules. A term, under an interpretation of the language, will always denote an element of a domain, provided an assignment of the occurring variables to elements of that domain has been given. In order to keep the syntax as simple as possible, terms will be understood as certain parenthesis-free strings, although this kind of writing may look rather unusual at the first glance.

Terms in L:

(T1) Variables and constants, considered as atomic strings, are terms, also called *prime terms*.

(T2) If $f \in L$ is n-ary and $t_1, \ldots, t_n$ are terms, then $ft_1 \cdots t_n$ is a term.

This is a recursive definition of the set of terms as a subset of $\mathfrak{S}_{\mathcal{L}}$. Any string that is not generated by (T1) and (T2) is not a term in this context (cf. the related definition of $\mathfrak{F}$ in **1.1**). Parenthesis-free term notation simplifies the syntax, but for binary operations we proceed differently in practice and write, for example, the term $\cdot + xyz$ as $(x + y) \cdot z$. The reason is that a high density of information in the notation complicates reading. Our brain does not process information sequentially like a computer. Officially, terms are parenthesis-free, and the parenthesized notation is just an alternative way of rewriting terms. Similarly to the unique reconstruction property of propositional formulas in **1.1**, here the *unique term reconstruction* property holds, that is,

$$ft_1 \cdots t_n = fs_1 \cdots s_n \text{ implies } s_i = t_i \text{ for } i = 1, \ldots, n \quad (t_i, s_i \text{ terms}),$$

which immediately follows from the *unique term concatenation* property

$$t_1 \cdots t_n = s_1 \cdots s_m \text{ implies } n = m \text{ and } t_i = s_i \text{ for } i = 1, \ldots, n.$$

The latter is shown in Exercise 2. $\mathcal{T}$ $(= \mathcal{T}_L)$ denotes the set of all terms of a given signature L. Variable-free terms, which can exist only with the availability of constant symbols, are called *constant terms* or *ground terms*, mainly in logic programming. With the operations given in $\mathcal{T}$ by setting $f^{\mathcal{T}}(t_1, \ldots, t_n) = ft_1 \cdots t_n$, $\mathcal{T}$ forms an algebra, the *term algebra*. From the definition of terms immediately follows the useful

Principle of proof by term induction. *Let $\mathcal{E}$ be a property of strings such that $\mathcal{E}$ holds for all prime terms, and for each $n > 0$ and each n-ary function symbol f, the assumptions $\mathcal{E}t_1, \ldots, \mathcal{E}t_n$ imply $\mathcal{E}ft_1 \cdots t_n$. Then all terms have the property $\mathcal{E}$.*

Indeed, $\mathcal{T}$ is by definition the smallest set of strings satisfying the conditions of this principle, and hence a subset of the set of all strings with the property $\mathcal{E}$. A simple application of term induction is the proof that each compound term t is a *function term* in the sense that $t = ft_1 \cdots t_n$ for some n-ary function symbol f and some terms $t_1, \ldots, t_n$. Simply consider the property 't is either prime or a function term'. Term induction can also be executed on certain subsets of $\mathcal{T}$, for instance on ground terms.

We also have at our disposal a *definition principle* by term recursion which, rather than defining it generally, we present through examples. The set $\mathit{var}\, t$ of variables occurring in a term t is recursively defined by

$$\mathit{var}\, c = \emptyset \ ; \ \mathit{var}\, x = \{x\} \ ; \ \mathit{var}\, f t_1 \cdots t_n = \mathit{var}\, t_1 \cup \cdots \cup \mathit{var}\, t_n.$$

$\mathit{var}\, t$, and even $\mathit{var}\, \xi$ for any $\xi \in \mathcal{S}_{\mathcal{L}}$, can also be defined explicitly using concatenation. $\mathit{var}\, \xi$ is the set of all $x \in \mathit{Var}$ for which there are strings η, ϑ with $\xi = \eta x \vartheta$. The notion of a *subterm* of a term can also be defined recursively. Again, we can also do it more briefly using concatenation. Definition by term induction should more precisely be called *definition by term recursion*. But most authors are sloppy in this respect.

We now define recursively those strings of the alphabet of L to be called *formulas*, also termed (first-order) *expressions* or *well-formed formulas*.

Formulas in L:

(F1) If s, t are terms, then the string $s = t$ is a formula.

(F2) If $t_1, \ldots, t_n$ are terms and $r \in L$ is n-ary, then $r t_1 \cdots t_n$ is a formula.

(F3) If α, β are formulas and x is a variable, then $(\alpha \wedge \beta)$, $\neg\alpha$, and $\forall x \alpha$ are formulas.

Any string not generated according to (F1), (F2), (F3) is in this context not a formula. Other logical symbols serve throughout merely as abbreviations, namely $\exists x \alpha := \neg\forall x \neg\alpha$, $(\alpha \vee \beta) := \neg(\neg\alpha \wedge \neg\beta)$, and as in **1.1**, $(\alpha \to \beta) := \neg(\alpha \wedge \neg\beta)$, and $(\alpha \leftrightarrow \beta) := ((\alpha \to \beta) \wedge (\beta \to \alpha))$. In addition, $s \neq t$ will throughout be written for $\neg\, s = t$. The formulas $\forall x \alpha$ and $\exists x \alpha$ are said to arise from α by *quantification*.

Examples. (a) $\forall x \exists y\, x + y = 0$ (more explicitly, $\forall x \neg\forall y \neg\, x + y = 0$) is a formula, expressing 'for all x there exists a y such that $x + y = 0$'. Here we assume tacitly that x, y denote distinct variables. The same is assumed in all of the following whenever this can be made out from the context.

(b) $\forall x \forall x\, x = y$ is a formula, since repeated quantification of the same variable is not forbidden. $\forall z\, x = y$ is a formula also if $z \neq x, y$, although z does then not appear in the formula $x = y$.

Example (b) indicates that the grammar of our formal language is more liberal than one might expect. This will spare us a lot of writing. The formulas $\forall x \forall x\, x = y$ and $\exists x \forall x\, x = y$ both have the same meaning as $\forall x\, x = y$.

These three formulas are logically equivalent (in a sense still to be defined), as are $\forall z\, x = y$ and $x = y$. It would be to our disadvantage to require any restriction here. In spite of this liberality, the formula syntax corresponds roughly to the syntax of natural language.

The formulas procured by (F1) and (F2) are said to be *prime* or *atomic* formulas, or simply called *prime*. As in propositional logic, prime formulas and their negations are called *literals*.

Prime formulas of the form $s = t$ are called *equations*. These are the only prime formulas if L contains no relation symbols, in which case L is called an *algebraic* signature. Prime formulas that are not equations begin with a relation symbol, although in practice a binary symbol tends to separate the two arguments as, for example, in $x \leqslant y$. The official notation is, however, that of clause (F2). The unique term concatenation property clearly implies the *unique prime formula reconstruction* property

$$rt_1 \cdots t_n = rs_1 \cdots s_n \text{ implies } t_i = s_i \text{ for } i = 1, \ldots, n.$$

The set of all formulas in L is denoted by $\mathcal{L}$. If $L = \{\in\}$ or $L = \{\circ\}$ then $\mathcal{L}$ is also denoted by $\mathcal{L}_\in$ or $\mathcal{L}_\circ$, respectively. If L is more complex, e.g. $L = \{\circ, e\}$, we write $\mathcal{L} = \mathcal{L}\{\circ, e\}$. The case $L = \emptyset$ is also permitted; it defines the *language of pure identity*, denoted by $\mathcal{L}_=$.

Instead of terms, formulas, and structures of signature L, we will talk of $\mathcal{L}$-terms (writing $\mathcal{T}_\mathcal{L}$ for $\mathcal{T}_L$), $\mathcal{L}$-formulas, and $\mathcal{L}$-structures respectively. We also omit the prefix if $\mathcal{L}$ has been given earlier and use the same conventions of parenthesis economy as in **1.1**. We will also allow ourselves other informal aids in order to increase readability. For instance, variously shaped brackets may be used as in $\forall x \exists y \forall z [z \in y \leftrightarrow \exists u (z \in u \wedge u \in x)]$. Even verbal descriptions (partial or complete) are permitted, as long as the intended formula is uniquely recognizable.

The strings $\forall x$ and $\exists x$ (read "for all x" respectively "there is an x") are called *prefixes*. Also concatenations of these such as $\forall x \exists y$ are prefixes. No other prefixes are considered here. Formulas in which $\forall, \exists$ do not occur are termed *quantifier-free* or *open*. These are the Boolean combinations of prime formulas. Generally, the *Boolean combinations* of formulas from a set $X \subseteq \mathcal{L}$ are the ones generated by $\neg$, $\wedge$ (and $\vee$) from those of X.

X, Y, Z always denote sets of formulas, $\alpha, \beta, \gamma, \delta, \pi, \varphi, \ldots$ denote formulas, and s, t terms, while Φ, Ψ are reserved to denote finite sequences of

formulas and formal proofs. Substitutions (to be defined below) will be denoted by $\sigma, \tau, \omega, \rho$, and ι.

Principles of *proof by formula induction* and of *definition by formula induction* (more precisely *formula recursion*) also exist for first-order and other formal languages. After the explanation of these principles for propositional languages in **1.1**, it suffices to present here some examples, adhering to the maxim verba docent, exempla trahunt. Formula recursion is based on the *unique formula reconstruction*, which is similar to the corresponding property in **1.1**: Each composed $\varphi \in \mathcal{L}$ can uniquely be written as $\varphi = \neg\alpha$, $\varphi = (\alpha \wedge \beta)$, or $\forall x\alpha$ for some $\alpha, \beta \in \mathcal{L}$ and $x \in \mathit{Var}$. A simple example of a recursive definition is $\operatorname{rk}\varphi$, the *rank* of a formula φ. Starting with $\operatorname{rk}\pi = 0$ for prime formulas π it is defined as on page 8, with the additional clause $\operatorname{rk}\forall x\alpha = \operatorname{rk}\alpha + 1$. Functions on $\mathcal{L}$ are sometimes defined by recursion on $\operatorname{rk}\varphi$, not on φ, as for instance on page 60.

Useful for some purposes is also the *quantifier rank*, $\operatorname{qr}\varphi$. It represents a measure of nested quantifiers in φ. For prime π let $\operatorname{qr}\pi = 0$, and let $\operatorname{qr}\neg\alpha = \operatorname{qr}\alpha$, $\operatorname{qr}(\alpha \wedge \beta) = \max\{\operatorname{qr}\alpha, \operatorname{qr}\beta\}$, $\operatorname{qr}\forall x\alpha = \operatorname{qr}\alpha + 1$.

Note that $\operatorname{qr}\exists x\varphi = \operatorname{qr}\neg\forall x\neg\varphi = \operatorname{qr}\forall x\varphi$. A *subformula* of a formula is defined analogously to the definition in **1.1**. Hence, we need say no more on this. We write $x \in \mathit{bnd}\,\varphi$ (or x occurs bound in φ) if φ contains the prefix $\forall x$. In subformulas of φ of the form $\forall x\alpha$, the formula α is called the *scope* of $\forall x$. The same prefix can occur repeatedly and with nested scopes in φ, as for instance in $\forall x(\forall x\, x = 0 \wedge x < y)$. In practice we avoid this way of writing, though for a computer this would pose no problem.

Intuitively, the formulas (a) $\forall x\exists y\, x + y = 0$ and (b) $\exists y\, x + y = 0$ are different in that in every context with a given meaning for $+$ and 0, the former is either true or false, whereas in (b) the variable x is waiting to be assigned a value. One also says that all variables in (a) are bound, while (b) contains the "free" variable x. The syntactic predicate 'x occurs free in φ', or '$x \in \mathit{free}\,\varphi$' is defined inductively: Let $\mathit{free}\,\alpha = \mathit{var}\,\alpha$ for prime formulas α ($\mathit{var}\,\alpha$ was defined on page 56), and

$$\mathit{free}\,(\alpha \wedge \beta) = \mathit{free}\,\alpha \cup \mathit{free}\,\beta,\quad \mathit{free}\,\neg\alpha = \mathit{free}\,\alpha,\quad \mathit{free}\,\forall x\alpha = \mathit{free}\,\alpha \setminus \{x\}.$$

For instance, $\mathit{free}\,(\forall x\exists y\, x + y = 0) = \emptyset$, while $\mathit{free}\,(x \leqslant y \wedge \forall x\exists y\, x + y = 0)$ equals $\{x, y\}$. As the last formula shows, x can occur both free and bound in a formula. This too will be avoided in practice whenever possible. In some proof-theoretically oriented presentations, even different symbols are

chosen for free and bound variables. Each of these approaches has its advantages and its disadvantages.

Formulas without free variables are called *sentences*, or *closed formulas*. $1+1=0$ and $\forall x\exists y\, x+y=0$ $(=\forall x\neg\forall y\neg x+y=0)$ are examples. Throughout take $\mathcal{L}^0$ to denote the set of all sentences of $\mathcal{L}$. More generally, let $\mathcal{L}^k$ be the set of all formulas φ such that $\mathit{free}\,\varphi \subseteq \mathit{Var}_k := \{\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{k-1}\}$. Clearly, $\mathcal{L}^0 \subseteq \mathcal{L}^1 \subseteq \cdots$ and $\mathcal{L} = \bigcup_{k\in\mathbb{N}} \mathcal{L}^k$.

At this point we meet a for the remainder of the book valid

Convention. As long as not otherwise stated, the notation $\varphi = \varphi(x)$ means that the formula φ contains at most x as a free variable; more generally, $\varphi = \varphi(x_1, \ldots, x_n)$ or $\varphi = \varphi(\vec{x})$ is to mean $\mathit{free}\,\varphi \subseteq \{x_1, \ldots, x_n\}$, where $x_1, \ldots, x_n$ stand for arbitrary but distinct variables. Not all of these variables need actually occur in φ. Further, $t = t(\vec{x})$ for terms t is to be read completely analogously.

The term $ft_1\cdots t_n$ is often denoted by $f\vec{t}$, the prime formula $rt_1\cdots t_n$ by $r\vec{t}$. Here $\vec{t}$ denotes the string concatenation $t_1\cdots t_n$. Fortunately, $\vec{t}$ behaves exactly like the sequence $(t_1, \ldots, t_n)$ as was pointed out already; it has the unique term concatenation property, see page 55.

Substitutions. We begin with the substitution $\frac{t}{x}$ of some term t for a single variable x, called a *simple substitution*. Put intuitively, $\varphi\frac{t}{x}$ (also denoted by $\varphi_x(t)$ and read "φ t for x") is the formula that results from replacing all free occurrences of x in φ by the term t. This intuitive characterization is made precise recursively, first for terms by

$$x\tfrac{t}{x} = t, \quad y\tfrac{t}{x} = y \quad (x \neq y), \quad c\tfrac{t}{x} = c, \quad (ft_1\cdots t_n)\tfrac{t}{x} = ft_1'\cdots t_n',$$

where, for brevity, t_i' stands for $t_i\frac{t}{x}$, and next for formulas as follows:

$$(t_1=t_2)\tfrac{t}{x} = t_1'=t_2',\ (r\vec{t})\tfrac{t}{x} = rt_1'\cdots t_n', \qquad (\forall y\alpha)^{\frac{t}{x}} = \begin{cases}\forall y\alpha \text{ if } x = y,\\ \forall y(\alpha\tfrac{t}{x}) \text{ otherwise.}\end{cases}$$
$$(\alpha\wedge\beta)\tfrac{t}{x} = \alpha\tfrac{t}{x}\wedge\beta\tfrac{t}{x},\ (\neg\alpha)\tfrac{t}{x} = \neg(\alpha\tfrac{t}{x}),$$

Then also $(\alpha\to\beta)\frac{t}{x} = \alpha\frac{t}{x} \to \beta\frac{t}{x}$, and the corresponding holds for $\vee$, while $(\exists y\alpha)\frac{t}{x} = \exists y\alpha$ for $y = x$, and $\exists y(\alpha\frac{t}{x})$ otherwise. Simple substitutions are special cases of so-called *simultaneous substitutions*

$$\varphi\tfrac{t_1\cdots t_n}{x_1\cdots x_n} \qquad (x_1, \ldots, x_n \text{ distinct}).$$

For brevity, this will be written $\varphi\frac{\vec{t}}{\vec{x}}$ or $\varphi_{\vec{x}}(\vec{t})$ or just $\varphi(\vec{t})$, provided there is no danger of misunderstanding. Here the variables x_i are simultaneously replaced by the terms t_i at free occurrences. Simultaneous substitutions

easily generalize to *global* substitutions σ. Such a σ assigns to *every* variable x a term $x^\sigma \in \mathcal{T}$. It extends to the whole of $\mathcal{T}$ by the clauses $c^\sigma = c$ and $(f\vec{t}\,)^\sigma = ft_1^\sigma \cdots t_n^\sigma$, and subsequently to $\mathcal{L}$ by recursion on $\operatorname{rk}\varphi$, so that σ is defined for the whole of $\mathcal{T} \cup \mathcal{L}$: $(t_1 \mathbin{=} t_2)^\sigma = \ t_1^\sigma \mathbin{=} t_2^\sigma$, $(r\vec{t}\,)^\sigma = rt_1^\sigma \cdots t_n^\sigma$, $(\alpha \wedge \beta)^\sigma = \alpha^\sigma \wedge \beta^\sigma$, $(\neg\alpha)^\sigma = \neg\alpha^\sigma$, and $(\forall x\varphi)^\sigma = \forall x\varphi^\tau$, where τ is defined by $x^\tau = x$ and $y^\tau = y^\sigma$ for $y \neq x$.[3]

These clauses cover also the case of a simultaneous substitution, because $\frac{\vec{t}}{\vec{x}}$ can be identified with the global substitution σ such that $x_i^\sigma = t_i$ for $i = 1, \ldots, n$ and $x^\sigma = x$ otherwise. In other words, a simultaneous substitution can be understood as a global substitution σ such that $x^\sigma = x$ for *almost all* variables x, i.e., with the exception of finitely many. The *identical substitution*, always denoted by ι, is defined by $x^\iota = x$ for all x; hence $t^\iota = t$ and $\varphi^\iota = \varphi$ for all terms t and formulas φ.

Clearly, a global substitution yields *locally*, i.e. with respect to individual formulas, the same as a suitable simultaneous substitution. Moreover, it will turn out below that simultaneous substitutions are products of simple ones. Nonetheless, a separate study of simultaneous substitutions is useful mainly for Chapter **4**.

It always holds that $\frac{t_1 t_2}{x_1 x_2} = \frac{t_2 t_1}{x_2 x_1}$, whereas the compositions $\frac{t_1}{x_1}\,\frac{t_2}{x_2}$ and $\frac{t_2}{x_2}\,\frac{t_1}{x_1}$ are distinct, in general. Let us elaborate by explaining the difference between $\varphi\,\frac{t_1 t_2}{x_1 x_2}$ and $\varphi\,\frac{t_1}{x_1}\,\frac{t_2}{x_2}$ $\big(= (\varphi\,\frac{t_1}{x_1})\,\frac{t_2}{x_2}\big)$. For example, if one wants to swap x_1, x_2 at their free occurrences in φ then the desired formula is $\varphi\,\frac{x_2 x_1}{x_1 x_2}$, but not, in general, $\varphi\,\frac{x_2}{x_1}\,\frac{x_1}{x_2}$ (choose for instance $\varphi = x_1{<}x_2$). Rather $\varphi\,\frac{x_2 x_1}{x_1 x_2} = \varphi\,\frac{y}{x_2}\,\frac{x_2}{x_1}\,\frac{x_1}{y}$ for any $y \notin \mathit{var}\,\varphi \cup \{x_1, x_2\}$, as is readily shown by induction on φ after first treating terms. We recommend to carry out this induction in detail. In the same way we obtain

$$(1)\quad \varphi\frac{\vec{t}}{\vec{x}} = \varphi\,\frac{y}{x_n}\,\frac{t_1 \cdots t_{n-1}}{x_1 \cdots x_{n-1}}\,\frac{t_n}{y} \qquad (y \notin \mathit{var}\,\varphi \cup \mathit{var}\,\vec{x} \cup \mathit{var}\,\vec{t},\ n \geqslant 2).$$

This formula shows that a simultaneous substitution is a suitable product (composition) of simple substitutions. Conversely, it can be shown that each such product can be written as a single simultaneous substitution. In some cases (1) can be simplified. Useful, for example, is the following equation which holds in particular when all terms t_i are variable-free:

$$(2)\quad \varphi\frac{\vec{t}}{\vec{x}} = \varphi\,\frac{t_1}{x_1} \cdots \frac{t_n}{x_n} \qquad (x_i \notin \mathit{var}\,t_j \text{ for } i \neq j).$$

[3] Since $\operatorname{rk}\varphi < \operatorname{rk}\forall x\varphi$, we may assume according to the recursive construction of σ that φ^τ is already defined for all global substitutions τ.

Getting on correctly with substitutions is not altogether simple; it requires practice, because our ability to regard complex strings is not especially trustworthy. A computer is not only much faster but also more reliable in this respect.

Exercises

1. Show by term induction that a terminal segment of a term t is a concatenation $s_1 \cdots s_m$ of terms s_i for some $m \geqslant 1$. Thus, a symbol in t is at each position in t the initial symbol of a unique subterm s of t. The uniqueness of s is an easy consequence of Exercise 2(a).

2. Let $\mathcal{L}$ be a first-order language, $\mathcal{T} = \mathcal{T}_\mathcal{L}$, and $\mathcal{E}t$ the property 'No proper initial segment of t ($\in \mathcal{T}$) is a term, nor is t a proper initial segment of a term from $\mathcal{T}$'. Prove (a) $\mathcal{E}t$ for all $t \in \mathcal{T}$, hence $t\xi = t'\xi' \Rightarrow t = t'$ for all $t, t' \in \mathcal{T}$ and arbitrary $\xi, \xi' \in \mathcal{S}_\mathcal{L}$, and (b) the unique term concatenation property (page 55).

3. Prove (a) No proper initial segment of a formula φ is a formula. (b) The unique formula reconstruction property stated on page 58. (c) $\neg\xi \in \mathcal{L} \Rightarrow \xi \in \mathcal{L}$ and $\alpha, (\alpha \wedge \xi) \in \mathcal{L} \Rightarrow \xi \in \mathcal{L}$. (c) easily yields (d) $\alpha, (\alpha \rightarrow \xi) \in \mathcal{L} \Rightarrow \xi \in \mathcal{L}$, for all $\xi \in \mathcal{S}_\mathcal{L}$.

4. Prove $\varphi \frac{t}{x} = \varphi$ for $x \notin \mathit{free}\,\varphi$, and $\varphi \frac{y}{x} \frac{t}{y} = \varphi \frac{t}{x}$ for $y \notin \mathit{var}\,\varphi$. It can be shown that these restrictions are indispensable, provided $t \neq x$.

5. Let $X \subseteq \mathcal{L}$ be a nonempty formula set and $X^* = X \cup \{\neg\varphi \mid \varphi \in X\}$. Show that a Boolean combination of formulas from X is equivalent to a disjunction of conjunctions of formulas from X^*.

2.3 Semantics of First-Order Languages

Intuitively it is clear that the formula $\exists y\, y + y = x$ can be allocated a truth value in the domain $(\mathbb{N}, +)$ only if to the free variable x there corresponds a value in $\mathbb{N}$. Thus, along with an interpretation of the extralogical symbols, a truth value allocation for a formula φ requires a valuation of at least the variables occurring free in φ. However, it is technically more convenient

to work with a global assignment of values to all variables, even if in a concrete case only the values of finitely many variables are needed. We therefore begin with the following

Definition. A *model* $\mathcal{M}$ is a pair $(\mathcal{A}, w)$ consisting of an $\mathcal{L}$-structure $\mathcal{A}$ and a *valuation* $w\colon Var \to A$, $w\colon x \mapsto x^w$. We denote $r^{\mathcal{A}}, f^{\mathcal{A}}, c^{\mathcal{A}}$, and x^w also by $r^{\mathcal{M}}, f^{\mathcal{M}}, c^{\mathcal{M}}$, and $x^{\mathcal{M}}$, respectively. The domain of $\mathcal{A}$ will also called the *domain of* $\mathcal{M}$.

Models are sometimes called *interpretations*, occasionally also *$\mathcal{L}$-models* if the connection to $\mathcal{L}$ is to be highlighted. Some authors identify models with structures from the outset. This also happens in **2.5**, where we are talking about models of theories. The notion of a model is to be maintained sufficiently flexible in logic and mathematics.

A model $\mathcal{M}$ allocates in a natural way to every term t a value in A, denoted by $t^{\mathcal{M}}$ or $t^{\mathcal{A},w}$ or just by t^w. Clearly, for prime terms the value is already given by $\mathcal{M}$. This evaluation extends to compound terms by term induction as follows: $(f\vec{t})^{\mathcal{M}} = f^{\mathcal{M}}\vec{t}^{\,\mathcal{M}}$, where $\vec{t}^{\,\mathcal{M}}$ abbreviates here the sequence $(t_1^{\mathcal{M}}, \ldots, t_n^{\mathcal{M}})$. If the context allows we neglect the superscripts and retain just an imaginary distinction between symbols and their interpretation. For instance, if $\mathcal{A} = (\mathbb{N}, +, \cdot, 0, 1)$ and $x^w = 2$, say, we write somewhat sloppily $(0 \cdot x + 1)^{\mathcal{A},w} = 0 \cdot 2 + 1 = 1$.

The value of t under $\mathcal{M}$ depends only on the meaning of the symbols that effectively occur in t; using induction on t, the following slightly more general claim is obtained: if $var\, t \subseteq V \subseteq Var$ and $\mathcal{M}, \mathcal{M}'$ are models with the same domain such that $x^{\mathcal{M}} = x^{\mathcal{M}'}$ for all $x \in V$ and $\mathsf{s}^{\mathcal{M}} = \mathsf{s}^{\mathcal{M}'}$ for all remaining symbols s occurring in t, then $t^{\mathcal{M}} = t^{\mathcal{M}'}$. Clearly, $t^{\mathcal{A},w}$ may simply be denoted by $t^{\mathcal{A}}$, provided the term t contains no variables.

We now are going to define a satisfiability relation $\vDash$ between models $\mathcal{M} = (\mathcal{A}, w)$ and formulas φ, using induction on φ as in **1.3**. We read $\mathcal{M} \vDash \varphi$ as $\mathcal{M}$ *satisfies* φ, or $\mathcal{M}$ *is a model for* φ.

Sometimes $\mathcal{A} \vDash \varphi\,[w]$ is written instead of $\mathcal{M} \vDash \varphi$. A similar notation, just as frequently encountered, is introduced later. Each of these notations has its advantages, depending on the context. If $\mathcal{M} \vDash \varphi$ for all $\varphi \in X$ we write $\mathcal{M} \vDash X$ and call $\mathcal{M}$ a *model for* X. For the formulation of the satisfaction clauses below (taken from [Ta1]) we consider for given $\mathcal{M} = (\mathcal{A}, w)$, $x \in Var$, and $a \in A$ also the model $\mathcal{M}_x^a$ (generalized to $\mathcal{M}_{\vec{x}}^{\vec{a}}$

below). $\mathcal{M}_x^a$ differs from $\mathcal{M}$ only in that the variable x receives the value $a \in A$ instead of $x^{\mathcal{M}}$. Thus, $\mathcal{M}_x^a = (\mathcal{A}, w')$ with $x^{w'} = a$ and $y^{w'} = y^w$ otherwise. The satisfaction clauses then look as follows:

$$\begin{array}{lll} \mathcal{M} \vDash s \equiv t & \Leftrightarrow & s^{\mathcal{M}} = t^{\mathcal{M}}, \\ \mathcal{M} \vDash r\vec{t} & \Leftrightarrow & r^{\mathcal{M}}\vec{t}^{\,\mathcal{M}}, \\ \mathcal{M} \vDash (\alpha \wedge \beta) & \Leftrightarrow & \mathcal{M} \vDash \alpha \text{ and } \mathcal{M} \vDash \beta, \\ \mathcal{M} \vDash \neg\alpha & \Leftrightarrow & \mathcal{M} \nvDash \alpha, \\ \mathcal{M} \vDash \forall x\alpha & \Leftrightarrow & \mathcal{M}_x^a \vDash \alpha \text{ for all } a \in A. \end{array}$$

Remark 1. The last satisfaction clause can be stated differently if a name for each $a \in A$, say $\boldsymbol{a}$, is available in the signature: $\mathcal{M} \vDash \forall x\alpha \Leftrightarrow \mathcal{M} \vDash \alpha\frac{\boldsymbol{a}}{x}$ for all $a \in A$. This assumption permits the definition of the satisfaction relation for sentences using induction on sentences while bypassing arbitrary formulas. If not every $a \in A$ has a name in L, one could "fill up" L in advance by adjoining to L a name $\boldsymbol{a}$ for each a. But expanding the language is not always wanted and does not really simplify the matter.

$\mathcal{M}_x^a$ is slightly generalized to $\mathcal{M}_{\vec{x}}^{\vec{a}} := \mathcal{M}_{x_1\cdots x_n}^{a_1\cdots a_n}$ $(= (\mathcal{M}_{x_1}^{a_1})_{x_2}^{a_2} \dots)$, which differs from $\mathcal{M}$ in the values of a sequence $x_1, \dots, x_n$ of distinct variables. This and writing $\forall\vec{x}\varphi$ for $\forall x_1 \cdots \forall x_n\varphi$ permits a short notation of a useful generalization of the last clause above, namely

$$\mathcal{M} \vDash \forall\vec{x}\varphi \;\Leftrightarrow\; \mathcal{M}_{\vec{x}}^{\vec{a}} \vDash \varphi \text{ for all } \vec{a} \in A^n.$$

The definitions of $\alpha \vee \beta$, $\alpha \to \beta$, and $\alpha \leftrightarrow \beta$ from page 56 readily imply the additional clauses $\mathcal{M} \vDash \alpha \vee \beta$ *iff* $\mathcal{M} \vDash \alpha$ or $\mathcal{M} \vDash \beta$, $\mathcal{M} \vDash \alpha \to \beta$ *iff* $\mathcal{M} \vDash \alpha \Rightarrow \mathcal{M} \vDash \beta$, and analogously for $\leftrightarrow$. Clearly, if $\vee, \to, \leftrightarrow$ were treated as independent connectives, these equivalences would have to be added to the above ones. Further, the definition of $\exists x\varphi$ in **2.2** corresponds to its intended meaning, because $\mathcal{M} \vDash \exists x\varphi \Leftrightarrow \mathcal{M}_x^a \vDash \varphi$ for some $a \in A$. Indeed, whenever $\mathcal{M} \vDash \neg\forall x\neg\varphi$ $(= \exists x\varphi)$ then $\mathcal{M}_x^a \vDash \neg\varphi$ does not hold for all a; hence there is some $a \in A$ such that $\mathcal{M}_x^a \nvDash \neg\varphi$, or equivalently, $\mathcal{M}_x^a \vDash \varphi$. And this chain of reasoning is obviously reversible.

Example 1. $\mathcal{M} \vDash \exists x\, x = t$ for arbitrary $\mathcal{M}$, provided $x \notin \mathit{var}t$. Indeed, $\mathcal{M}_x^a \vDash x = t$ with $a := t^{\mathcal{M}}$, since $x^{\mathcal{M}_x^a} = a = t^{\mathcal{M}} = t^{\mathcal{M}_x^a}$ in view of $x \notin \mathit{var}t$. The assumption $x \notin \mathit{var}t$ is essential. For instance, $\mathcal{M} \vDash \exists x\, x = fx$ holds only if the function $f^{\mathcal{M}}$ has a fixed point.

We now introduce several fundamental notions that will be treated more systematically in **2.4** and **2.5**, once certain necessary preparations have been completed.

Definition. A formula or set of formulas in $\mathcal{L}$ is termed *satisfiable* if it has a model. $\varphi \in \mathcal{L}$ is called *generally valid, logically valid*, or a *tautology*, in short, $\models \varphi$, if $\mathcal{M} \models \varphi$ for every model $\mathcal{M}$. Formulas α, β are called (logically or semantically) *equivalent*, in symbols, $\alpha \equiv \beta$, if

$$\mathcal{M} \models \alpha \;\Leftrightarrow\; \mathcal{M} \models \beta, \text{ for each } \mathcal{L}\text{-model } \mathcal{M}.$$

Further, let $\mathcal{A} \models \varphi$ (read φ *holds in* $\mathcal{A}$ or $\mathcal{A}$ *satisfies* φ) if $(\mathcal{A}, w) \models \varphi$ for all $w\colon Var \to A$. One writes $\mathcal{A} \models X$ in case $\mathcal{A} \models \varphi$ for all $\varphi \in X$. Finally, let $X \models \varphi$ (read *from* X *follows* φ, or φ *is a consequence of* X) if every model $\mathcal{M}$ of X also satisfies the formula φ, i.e., $\mathcal{M} \models X \Rightarrow \mathcal{M} \models \varphi$.

As in Chapter 1, $\models$ denotes both the satisfaction and the consequence relation. Here, as there, we write $\varphi_1, \ldots, \varphi_n \models \varphi$ for $\{\varphi_1, \ldots, \varphi_n\} \models \varphi$. Note that in addition, $\models$ denotes the validity relation in structures, which is illustrated by the following

Example 2. We show that $\mathcal{A} \models \forall x \exists y\, x \neq y$, where the domain of $\mathcal{A}$ contains at least two elements. Indeed, let $\mathcal{M} = (\mathcal{A}, w)$ and let $a \in A$ be given arbitrarily. Then there exists some $b \in A$ with $a \neq b$. Hence, $(\mathcal{M}^a_x)^b_y = \mathcal{M}^{a\,b}_{xy} \models x \neq y$, and so $\mathcal{M}^a_x \models \exists y\, x \neq y$. Since a was arbitrary, $\mathcal{M} \models \forall x \exists y\, x \neq y$. Clearly the actual values of w are irrelevant in this argument. Hence $(\mathcal{A}, w) \models \forall x \exists y\, x \neq y$ for all w, that is, $\mathcal{A} \models \forall x \exists y\, x \neq y$.

Here some care is needed. While $\mathcal{M} \models \varphi$ or $\mathcal{M} \models \neg\varphi$ for all formulas, $\mathcal{A} \models \varphi$ or $\mathcal{A} \models \neg\varphi$ (the law of the excluded middle for validity in structures) is in general correct only for sentences φ, as Theorem 3.1 will show. If $\mathcal{A}$ contains more than one element, then, for example, neither $\mathcal{A} \models x = y$ nor $\mathcal{A} \models x \neq y$. Indeed, $x = y$ is falsified by any w such that $x^w \neq y^w$, and $x \neq y$ by any w with $x^w = y^w$. This is one of the reasons why models were not simply identified with structures.

For $\varphi \in \mathcal{L}$ let φ^g be the sentence $\forall x_1 \cdots \forall x_m \varphi$, where $x_1, \ldots, x_m$ is an enumeration of *free*$\,\varphi$ according to index size, say. φ^g is called the *generalized of* φ, also called its *universal closure*. For $\varphi \in \mathcal{L}^0$ clearly $\varphi^g = \varphi$. From the definitions immediately results

(1) $\quad \mathcal{A} \models \varphi \;\Leftrightarrow\; \mathcal{A} \models \varphi^g$,

and more generally, $\mathcal{A} \models X \Leftrightarrow \mathcal{A} \models X^g$ $(:= \{\varphi^g \mid \varphi \in X\})$. (1) explains why φ and φ^g are often notionally identified, and the information that formally runs φ^g is often shortened to φ. It must always be clear from

the context whether our eye is on validity in a structure, or on validity in a model with its fixed valuation. Only in the first case can a generalization (or globalization) of the free variables be thought of as carried out. However, independent of this discussion, $\vDash \varphi \;\Leftrightarrow\; \vDash \varphi^g$ always holds.

Even after just these incomplete considerations it is already clear that numerous properties of structures and whole systems of axioms can adequately be described by first-order formulas and sentences. Thus, for example, an axiom system for groups in $\circ, e, {}^{-1}$, mentioned already in **2.1**, can be formulated as follows:

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

Precisely, the sentences that follow from these axioms form the *elementary group theory in* $\circ, e, {}^{-1}$. It will be denoted by $T_G^{=}$. In the sense elaborated in Exercise 3 in **2.6** an equivalent formulation of the theory of groups in $\circ, e$, denoted by T_G, is obtained if the third $T_G^{=}$-axiom is replaced by $\forall x \exists y\, x \circ y = e$. Let us mention that $\forall x\, e \circ x = x$ and $\forall x \exists y\, y \circ x = e$ are provable in T_G and also in $T_G^{=}$.

An axiom system for ordered sets can also easily be provided, in that one formalizes the properties of being irreflexive, transitive, and connex. Here and elsewhere, $\forall x_1 \cdots x_n \varphi$ stands for $\forall x_1 \cdots \forall x_n \varphi$:

$$\forall x\ x \not< x;\ \forall xyz(x < y \wedge y < z \to x < z);\ \forall xy(x \neq y \to x < y \vee y < x).$$

In writing down these and other axioms the outer $\forall$-prefixes are very often omitted so as to save on writing, and we think implicitly of the generalization of variables as having been carried out. This kind of economical writing is employed also in the formulation of (1) above, which strictly speaking runs 'for all $\mathcal{A}, \varphi : \mathcal{A} \vDash \varphi \;\Leftrightarrow\; \mathcal{A} \vDash \varphi^g$'.

For sentences α of a given language it is intuitively clear that the values of the variables of w for the relation $(\mathcal{A}, w) \vDash \alpha$ are irrelevant. The precise proof is extracted from the following theorem for $V = \emptyset$. Thus, either $(\mathcal{A}, w) \vDash \alpha$ for all w and hence $\mathcal{A} \vDash \alpha$, or else $(\mathcal{A}, w) \vDash \alpha$ for no w, i.e., $(\mathcal{A}, w) \vDash \neg\alpha$ for all w, and hence $\mathcal{A} \vDash \neg\alpha$. Sentences therefore obey the already-cited tertium non datur.

Theorem 3.1 (Coincidence theorem). *Let $V \subseteq$ Var, free$\varphi \subseteq V$, and $\mathcal{M}, \mathcal{M}'$ be models on the same domain A such that $x^{\mathcal{M}} = x^{\mathcal{M}'}$ for all $x \in V$, and $\mathsf{s}^{\mathcal{M}} = \mathsf{s}^{\mathcal{M}'}$ for all extralogical symbols s occurring in φ. Then $\mathcal{M} \vDash \varphi \;\Leftrightarrow\; \mathcal{M}' \vDash \varphi$.*

Proof by induction on φ. Let $\varphi = r\vec{t}$ be prime, so that $var\,\vec{t} \subseteq V$. As was mentioned earlier, the value of a term t depends only on the meaning of the symbols occurring in t. But in view of the suppositions, these meanings are the same in $\mathcal{M}$ and $\mathcal{M}'$. Therefore, $\vec{t}^{\,\mathcal{M}} = \vec{t}^{\,\mathcal{M}'}$ (i.e., $t_i^{\mathcal{M}} = t_i^{\mathcal{M}'}$ for $i = 1, \ldots, n$), and so $\mathcal{M} \vDash r\vec{t} \Leftrightarrow r^{\mathcal{M}}\vec{t}^{\,\mathcal{M}} \Leftrightarrow r^{\mathcal{M}'}\vec{t}^{\,\mathcal{M}'} \Leftrightarrow \mathcal{M}' \vDash r\vec{t}$. For equations $t_1 = t_2$ one reasons analogously. Further, the induction hypothesis for α, β yields $\mathcal{M} \vDash \alpha \wedge \beta \Leftrightarrow \mathcal{M} \vDash \alpha, \beta \Leftrightarrow \mathcal{M}' \vDash \alpha, \beta \Leftrightarrow \mathcal{M}' \vDash \alpha \wedge \beta$. In the same way one obtains $\mathcal{M} \vDash \neg\alpha \Leftrightarrow \mathcal{M}' \vDash \neg\alpha$. By the induction step on $\forall$ it becomes clear that the induction hypothesis needs to be skillfully formulated. It must be given with respect to any pair $\mathcal{M}, \mathcal{M}'$ of models and any subset V of *Var*.

Therefore let $a \in A$ and $\mathcal{M}_x^a \vDash \varphi$. Since for $V' := V \cup \{x\}$ certainly $free\,\varphi \subseteq V'$ and the models $\mathcal{M}_x^a$, $\mathcal{M}'^a_x$ coincide for all $y \in V'$ (although in general $x^{\mathcal{M}} \neq x^{\mathcal{M}'}$), by the induction hypothesis $\mathcal{M}_x^a \vDash \varphi \Leftrightarrow \mathcal{M}'^a_x \vDash \varphi$, for each $a \in A$. This clearly implies

$$\mathcal{M} \vDash \forall x\varphi \Leftrightarrow \mathcal{M}_x^a \vDash \varphi \text{ for all } a \Leftrightarrow \mathcal{M}'^a_x \vDash \varphi \text{ for all } a \Leftrightarrow \mathcal{M}' \vDash \forall x\varphi. \qquad \square$$

It follows from this theorem that an $\mathcal{L}$-model $\mathcal{M} = (\mathcal{A}, w)$ of φ for the case that $\varphi \in \mathcal{L} \subseteq \mathcal{L}'$ can be completely arbitrarily expanded to an $\mathcal{L}'$-model $\mathcal{M}' = (\mathcal{A}', w)$ of φ, i.e., arbitrarily fixing $\mathsf{s}^{\mathcal{M}'}$ for $\mathsf{s} \in L' \setminus L$ gives $\mathcal{M} \vDash \varphi \Leftrightarrow \mathcal{M}' \vDash \varphi$ by the above theorem with $V = Var$. This readily implies that the consequence relation $\vDash_{\mathcal{L}'}$ with respect to $\mathcal{L}'$ is a *conservative* extension of $\vDash_{\mathcal{L}}$ in that $X \vDash_{\mathcal{L}} \varphi \Leftrightarrow X \vDash_{\mathcal{L}'} \varphi$, for all sets $X \subseteq \mathcal{L}$ and all $\varphi \in \mathcal{L}$. Hence, there is no need here for using indices. In particular, the satisfiability or general validity of φ depends only on the symbols effectively occurring in φ.

Another application of Theorem 3.1 is the following fact, which justifies the already mentioned "omission of superfluous quantifiers."

(2) $\forall x\varphi \equiv \varphi \equiv \exists x\varphi$ whenever $x \notin free\,\varphi$.

Indeed, $x \notin free\,\varphi$ implies $\mathcal{M} \vDash \varphi \;\Leftrightarrow\; \mathcal{M}_x^a \vDash \varphi$ (here $a \in A$ is arbitrary) according to Theorem 3.1; choose $\mathcal{M}' = \mathcal{M}_x^a$ and $V = free\,\varphi$. Therefore,

$$\mathcal{M} \vDash \forall x\varphi \Leftrightarrow \mathcal{M}_x^a \vDash \varphi \text{ for all } a \Leftrightarrow \mathcal{M} \vDash \varphi$$
$$\Leftrightarrow \mathcal{M}_x^a \vDash \varphi \text{ for some } a \Leftrightarrow \mathcal{M} \vDash \exists x\varphi.$$

Very important for the next theorem and elsewhere is

(3) If $\mathcal{A} \subseteq \mathcal{B}$, $\mathcal{M} = (\mathcal{A}, w)$, $\mathcal{M}' = (\mathcal{B}, w)$ and $w : Var \to A$ then $t^{\mathcal{M}} = t^{\mathcal{M}'}$.

This is clear for prime terms, and the induction hypothesis $t_i^{\mathcal{M}} = t_i^{\mathcal{M}'}$ for $i = 1, \dots, n$ together with $f^{\mathcal{M}} = f^{\mathcal{M}'}$ imply

$$(f\vec{t})^{\mathcal{M}} = f^{\mathcal{M}}(t_1^{\mathcal{M}}, \dots, t_n^{\mathcal{M}}) = f^{\mathcal{M}'}(t_1^{\mathcal{M}'}, \dots, t_n^{\mathcal{M}'}) = (f\vec{t})^{\mathcal{M}'}.$$

For $\mathcal{M} = (\mathcal{A}, w)$ and $x_i^w = a_i$ let $t^{\mathcal{A},\vec{a}}$, or more suggestively $t^{\mathcal{A}}(\vec{a})$ denote the value of $t = t(\vec{x})$. Then (3) can somewhat more simply be written as

(4) $\mathcal{A} \subseteq \mathcal{B}$ *and* $t = t(\vec{x})$ *imply* $t^{\mathcal{A}}(\vec{a}) = t^{\mathcal{B}}(\vec{a})$ for all $\vec{a} \in A^n$.

Thus, along with the basic functions, also the so-called *term functions* $\vec{a} \mapsto t^{\mathcal{A}}(\vec{a})$ are the restrictions to their counterparts in $\mathcal{B}$. Clearly, if $n = 0$ or t is variable-free, one may write $t^{\mathcal{A}}$ for $t^{\mathcal{A}}(\vec{a})$. Note that in these cases $t^{\mathcal{A}} = t^{\mathcal{B}}$ whenever $\mathcal{A} \subseteq \mathcal{B}$, according to (4).

By Theorem 3.1 the satisfaction of φ in $(\mathcal{A}, w)$ depends only on the values of the $x \in \mathit{free}\,\varphi$. Let $\varphi = \varphi(\vec{x})$[4] and $\vec{a} = (a_1, \dots, a_n) \in A^n$. Then the statement

$$(\mathcal{A}, w) \vDash \varphi \text{ for a valuation } w \text{ with } x_1^w = a_1, \dots, x_n^w = a_n$$

can more suggestively be expressed by writing

$$(\mathcal{A}, \vec{a}) \vDash \varphi \quad \text{or} \quad \mathcal{A} \vDash \varphi\,[a_1, \dots, a_n] \quad \text{or} \quad \mathcal{A} \vDash \varphi\,[\vec{a}]$$

without mentioning w as a global valuation. Such notation also makes sense if w is restricted to a valuation on $\{x_1, \dots, x_n\}$. One may accordingly extend the concept of a model and call a pair $(\mathcal{A}, \vec{a})$ a model for a formula $\varphi(\vec{x})$ whenever $(\mathcal{A}, \vec{a}) \vDash \varphi(\vec{x})$, in particular if $\varphi \in \mathcal{L}^n$. We return to this extended concept in **4.1**. Until then we use it only for $n = 0$. That is, besides $\mathcal{M} = (\mathcal{A}, w)$ also the structure $\mathcal{A}$ itself is occasionally called a model for a set $S \subseteq \mathcal{L}^0$ of sentences, provided $\mathcal{A} \vDash S$.

As above let $\varphi = \varphi(\vec{x})$. Then $\varphi^{\mathcal{A}} := \{\vec{a} \in A^n \mid \mathcal{A} \vDash \varphi\,[\vec{a}]\}$ is called *the predicate defined by the formula* φ *in the structure* $\mathcal{A}$. For instance, the $\leqslant$-predicate in $(\mathbb{N}, +)$ is defined by $\varphi(x, y) = \exists z\, z + x \mathrel{=\!\!=} y$, but also by several other formulas.

More generally, a predicate $P \subseteq A^n$ is termed (explicitly or elementarily or first-order) *definable in* $\mathcal{A}$ if there is some $\varphi = \varphi(\vec{x})$ with $P = \varphi^{\mathcal{A}}$, and φ is called a *defining formula* for P. Analogously, $f\colon A^n \to A$ is called definable in $\mathcal{A}$ if $\varphi^{\mathcal{A}} = \operatorname{graph} f$ for some $\varphi(\vec{x}, y)$. One often talks in this

[4] Since this equation is to mean $\mathit{free}\,\varphi \subseteq \{x_1, \dots, x_n\}$, $\vec{x}$ is not uniquely determined by φ. Hence, the phrase "Let $\varphi = \varphi(\vec{x}) \dots$" implicitly includes along with a given φ also a tuple $\vec{x}$ given in advance. The notation $\varphi = \varphi(\vec{x})$ does not even state that φ contains free variables at all.

case of *explicit* definability of f in $\mathcal{A}$, to distinguish it from other kinds of definability. Much information is gained from the knowledge of which sets, predicates, or functions are definable in a structure. For instance, the sets definable in $(\mathbb{N}, 0, 1, +)$ are the eventually periodic ones (periodic from some number on). Thus, $\cdot$ cannot explicitly be defined by $+, 0, 1$ because the set of square numbers is not eventually periodic.

$\mathcal{A} \subseteq \mathcal{B}$ and $\varphi = \varphi(\vec{x})$ do not imply $\varphi^{\mathcal{A}} = \varphi^{\mathcal{B}} \cap A^n$, in general. For instance, let $\mathcal{A} = (\mathbb{N}, +)$, $\mathcal{B} = (\mathbb{Z}, +)$, and $\varphi = \exists z\, z + x = y$. Then $\varphi^{\mathcal{A}} = \leqslant^{\mathcal{A}}$, while $\varphi^{\mathcal{B}}$ contains all pairs $(a, b) \in \mathbb{Z}^2$. As the next theorem will show, $\varphi^{\mathcal{A}} = \varphi^{\mathcal{B}} \cap A^n$ holds in general only for open formulas φ, and is even characteristic for $\mathcal{A} \subseteq \mathcal{B}$ provided $A \subseteq B$. Clearly, $A \subseteq B$ is much weaker a condition than $\mathcal{A} \subseteq \mathcal{B}$:

Theorem 3.2 (Substructure theorem). *For structures $\mathcal{A}, \mathcal{B}$ such that $A \subseteq B$ the following conditions are equivalent:*

(i) $\mathcal{A} \subseteq \mathcal{B}$,

(ii) $\mathcal{A} \vDash \varphi\,[\vec{a}] \Leftrightarrow \mathcal{B} \vDash \varphi\,[\vec{a}]$, *for all open* $\varphi = \varphi(\vec{x})$ *and all* $\vec{a} \in A^n$,

(iii) $\mathcal{A} \vDash \varphi\,[\vec{a}] \Leftrightarrow \mathcal{B} \vDash \varphi\,[\vec{a}]$, *for all prime formulas* $\varphi(\vec{x})$ *and* $\vec{a} \in A^n$.

Proof. (i)⇒(ii): It suffices to prove that $\mathcal{M} \vDash \varphi \Leftrightarrow \mathcal{M}' \vDash \varphi$, with $\mathcal{M} = (\mathcal{A}, w)$ and $\mathcal{M}' = (\mathcal{B}, w)$, where $w\colon Var \to A$. In view of (3) the claim is obvious for prime formulas, and the induction steps for $\wedge, \neg$ are carried out just as in Theorem 3.1. (ii)⇒(iii): Trivial. (iii)⇒(i): By (iii), $r^{\mathcal{A}}\vec{a} \Leftrightarrow \mathcal{A} \vDash r\vec{x}\,[\vec{a}] \Leftrightarrow \mathcal{B} \vDash r\vec{x}\,[\vec{a}] \Leftrightarrow r^{\mathcal{B}}\vec{a}$. Analogously,

$$f^{\mathcal{A}}\vec{a} = b \;\Leftrightarrow\; \mathcal{A} \vDash f\vec{x} = y\,[\vec{a}, b] \;\Leftrightarrow\; \mathcal{B} \vDash f\vec{x} = y\,[\vec{a}, b] \;\Leftrightarrow\; f^{\mathcal{B}}\vec{a} = b,$$

for all $\vec{a} \in A^n$, $b \in A$. These conclusions state precisely that $\mathcal{A} \subseteq \mathcal{B}$. ❏

Let α be of the form $\forall\vec{x}\beta$ with open β, where $\forall\vec{x}$ may also be the empty prefix. Then α is a *universal* or $\forall$*-formula* (spoken "A-formula"), and for $\alpha \in \mathcal{L}^0$ also a *universal* or $\forall$*-sentence*. A simple example is $\forall x \forall y\; x = y$, which holds in $\mathcal{A}$ iff A contains precisely one element. Dually, $\exists\vec{x}\beta$ with β open is termed an $\exists$*-formula*, and an $\exists$*-sentence* whenever $\exists\vec{x}\beta \in \mathcal{L}^0$. Examples are the "how-many sentences"

$$\exists_1 := \exists \boldsymbol{v}_0\, \boldsymbol{v}_0 = \boldsymbol{v}_0; \quad \exists_n := \exists \boldsymbol{v}_0 \cdots \exists \boldsymbol{v}_{n-1} \textstyle\bigwedge_{i<j<n} \boldsymbol{v}_i \neq \boldsymbol{v}_j \quad (n > 1).$$

$\exists_n$ states 'there exist at least n elements', $\neg\exists_{n+1}$ thus that 'there exist at most n elements', and $\exists_{=n} := \exists_n \wedge \neg\exists_{n+1}$ says 'there exist exactly

n elements'. Since $\exists_1$ is a tautology, it is convenient to set $\top := \exists_1$, and $\exists_0 := \bot := \neg\top$ in all first-order languages with equality. Clearly, equivalent definitions of $\top$, $\bot$ may be used as well.

Corollary 3.3. *Let $\mathcal{A} \subseteq \mathcal{B}$. Then every $\forall$-sentence $\forall\vec{x}\alpha$ valid in $\mathcal{B}$ is also satisfied in $\mathcal{A}$. Dually, every $\exists$-sentence $\exists\vec{x}\beta$ valid in $\mathcal{A}$ is also valid in $\mathcal{B}$.*

Proof. Let $\mathcal{B} \vDash \forall\vec{x}\beta$ and $\vec{a} \in A^n$. Then $\mathcal{B} \vDash \beta[\vec{a}]$, hence $\mathcal{A} \vDash \beta[\vec{a}]$ by Theorem 3.2. $\vec{a}$ was arbitrary and therefore $\mathcal{A} \vDash \forall\vec{x}\beta$. Now let $\mathcal{A} \vDash \exists\vec{x}\beta$. Then $\mathcal{A} \vDash \beta[\vec{a}]$ for some $\vec{a} \in A^n$, hence $\mathcal{B} \vDash \beta[\vec{a}]$ by Theorem 3.2, and consequently $\mathcal{B} \vDash \exists\vec{x}\beta$. ❏

We now formulate a generalization of certain individual often-used arguments about the invariance of properties under isomorphisms:

Theorem 3.4 (Invariance theorem). *Let $\mathcal{A}, \mathcal{B}$ be isomorphic structures of signature L and let $\imath\colon \mathcal{A} \to \mathcal{B}$ be an isomorphism. Then for all $\varphi = \varphi(\vec{x})$*

$$\mathcal{A} \vDash \varphi[\vec{a}] \;\Leftrightarrow\; \mathcal{B} \vDash \varphi[\imath\vec{a}] \quad \big(\vec{a} \in A^n,\ \imath\vec{a} = (\imath a_1, \ldots, \imath a_n)\big).$$

In particular $\mathcal{A} \vDash \varphi \Leftrightarrow \mathcal{B} \vDash \varphi$, for all sentences φ of $\mathcal{L}$.

Proof. It is convenient to reformulate the claim as

$$\mathcal{M} \vDash \varphi \;\Leftrightarrow\; \mathcal{M}' \vDash \varphi \qquad \big(\mathcal{M} = (\mathcal{A}, w),\ \mathcal{M}' = (\mathcal{B}, w'),\ w' : x \mapsto \imath x^w\big).$$

This is easily confirmed by induction on φ after first proving $\imath(t^{\mathcal{M}}) = t^{\mathcal{M}'}$ inductively on t. This proof clearly includes the case $\varphi \in \mathcal{L}^0$. ❏

Thus, for example, it is once and for all clear that the isomorphic image of a group is a group even if we know at first only that it is a groupoid. Simply let α in the theorem run through all axioms of group theory. Another application: Let $\imath$ be an isomorphism of the group $\mathcal{A} = (A, \circ)$ onto the group $\mathcal{A}' = (A', \circ)$ and let e and e' denote their unit elements, not named in the signature. We claim that nonetheless $\imath e = e'$, using the fact that the unit element of a group is the only solution of $x \circ x = x$ (Example 2, page 83). Thus, since $\mathcal{A} \vDash e \circ e = e$, we get $\mathcal{A}' \vDash \imath e \circ \imath e = \imath e$ by Theorem 3.4, hence $\imath e = e'$. Theorem 3.4, incidentally, holds for formulas of higher order as well. For instance, the property of being a continuously ordered set (formalizable in a second-order language, see **3.8**) is likewise invariant under isomorphism.

$\mathcal{L}$-structures $\mathcal{A}, \mathcal{B}$ are termed *elementarily equivalent* if $\mathcal{A} \vDash \alpha \Leftrightarrow \mathcal{B} \vDash \alpha$, for all $\alpha \in \mathcal{L}^0$. One then writes $\mathcal{A} \equiv \mathcal{B}$. We consider this important notion

in **3.3** and more closely in **5.1**. Theorem 3.4 states in particular that $\mathcal{A} \simeq \mathcal{B} \Rightarrow \mathcal{A} \equiv \mathcal{B}$. The question immediately arises whether the converse of this also holds. For infinite structures the answer is negative (see **3.3**), for finite structures affirmative; a finite structure of a finite signature can, up to isomorphism, even be described by a single sentence. For example, the 2-element group $(\{0,1\},+)$ is up to isomorphism well determined by the following sentence, which tells us precisely how $+$ operates:

$$\exists \boldsymbol{v}_0 \exists \boldsymbol{v}_1 [\boldsymbol{v}_0 \neq \boldsymbol{v}_1 \wedge \forall x (x = \boldsymbol{v}_0 \vee x = \boldsymbol{v}_1) \\ \wedge \boldsymbol{v}_0 + \boldsymbol{v}_0 = \boldsymbol{v}_1 + \boldsymbol{v}_1 = \boldsymbol{v}_0 \wedge \boldsymbol{v}_0 + \boldsymbol{v}_1 = \boldsymbol{v}_1 + \boldsymbol{v}_0 = \boldsymbol{v}_1].$$

We now investigate the behavior of the satisfaction relation under substitution. The definition of $\varphi \frac{t}{x}$ in **2.2** pays no attention to *collision of variables*, which is taken to mean that some variables of the substitution term t fall into the scope of quantifiers after the substitution has been performed. In this case $\mathcal{M} \vDash \forall x \varphi$ does not necessarily imply $\mathcal{M} \vDash \varphi \frac{t}{x}$, although this might have been expected. In other words, $\forall x \varphi \vDash \varphi \frac{t}{x}$ is not unrestrictedly correct. For instance, if $\varphi = \exists y\, x \neq y$ then certainly $\mathcal{M} \vDash \forall x \varphi$ $(= \forall x \exists y\, x \neq y)$ whenever $\mathcal{M}$ has at least two elements, but $\mathcal{M} \vDash \varphi \frac{y}{x}$ $(= \exists y\, y \neq y)$ is certainly false. Analogously $\varphi \frac{t}{x} \vDash \exists x \varphi$ is not correct, in general. For example, choose $\forall y\, x = y$ for φ and y for t.

One could forcibly obtain $\forall x \varphi \vDash \varphi \frac{t}{x}$ without any limitation by renaming bound variables by a suitable modification of the inductive definition of $\varphi \frac{t}{x}$ in the quantifier step. However, such measures are rather unwieldy for the arithmetization of proof method in **6.2**. It is therefore preferable to put up with minor restrictions when we are formulating rules of deduction later. The restrictions we will use are somewhat stronger than they need to be but can be handled more easily; they look as follows:

Call $\varphi, \frac{t}{x}$ *collision-free* if $y \notin \mathit{bnd}\, \varphi$ for all $y \in \mathit{var}\, t$ distinct from x. We need not require $x \notin \mathit{bnd}\, \varphi$ because t is substituted only at free occurrences of x in φ, that is, x cannot fall after substitution within the scope of a prefix $\forall x$, even if $x \in \mathit{var}\, t$. For collision-free $\varphi, \frac{t}{x}$ we always get $\forall x \varphi \vDash \varphi \frac{t}{x}$ by Corollary 3.6 below.

If σ is a global substitution (see **2.2**) then φ, σ are termed *collision-free* if $\varphi, \frac{x^\sigma}{x}$ are collision-free for every $x \in \mathit{Var}$. If $\sigma = \frac{\vec{t}}{\vec{x}}$, this condition clearly need be checked only for the pairs $\varphi, \frac{x^\sigma}{x}$ with $x \in \mathit{var}\, \vec{x}$ and $x \in \mathit{free}\, \varphi$.

For $\mathcal{M} = (\mathcal{A}, w)$ put $\mathcal{M}^\sigma := (\mathcal{A}, w^\sigma)$ with $x^{w^\sigma} := (x^\sigma)^{\mathcal{M}}$ for $x \in \mathit{Var}$, so that $x^{\mathcal{M}^\sigma} = x^{\sigma\mathcal{M}}$ $(= (x^\sigma)^{\mathcal{M}})$. This equation reproduces itself to

(5) $t^{\mathcal{M}^\sigma} = t^{\sigma\mathcal{M}}$ for all terms t.

Indeed, $t^{\mathcal{M}^\sigma} = f^{\mathcal{M}}(t_1^{\mathcal{M}^\sigma}, \ldots, t_n^{\mathcal{M}^\sigma}) = f^{\mathcal{M}}(t_1^{\sigma\mathcal{M}}, \ldots, t_n^{\sigma\mathcal{M}}) = t^{\sigma\mathcal{M}}$ for $t = f\vec{t}$ in view of the induction hypothesis $t_i^{\mathcal{M}^\sigma} = t_i^{\sigma\mathcal{M}}$ $(i = 1, \ldots, n)$. Notice that $\mathcal{M}^\sigma$ coincides with $\mathcal{M}_{\vec{x}}^{\vec{t}^{\mathcal{M}}}$ for the case $\sigma = \frac{\vec{t}}{\vec{x}}$.

Theorem 3.5 (Substitution theorem). *Let $\mathcal{M}$ be a model and σ a global substitution. Then holds for all φ such that φ, σ are collision-free,*

(6) $\mathcal{M} \vDash \varphi^\sigma \;\Leftrightarrow\; \mathcal{M}^\sigma \vDash \varphi$.

In particular, $\mathcal{M} \vDash \varphi\,\frac{\vec{t}}{\vec{x}} \;\Leftrightarrow\; \mathcal{M}_{\vec{x}}^{\vec{t}^{\mathcal{M}}} \vDash \varphi$, provided $\varphi, \frac{\vec{t}}{\vec{x}}$ are collision-free.

Proof by induction on φ. In view of (5), we obtain

$$\mathcal{M} \vDash (t_1 \equiv t_2)^\sigma \;\Leftrightarrow\; t_1^{\sigma\mathcal{M}} = t_2^{\sigma\mathcal{M}} \;\Leftrightarrow\; t_1^{\mathcal{M}^\sigma} = t_2^{\mathcal{M}^\sigma} \;\Leftrightarrow\; \mathcal{M}^\sigma \vDash t_1 \equiv t_2.$$

Prime formulas $r\vec{t}$ are treated analogously. The induction steps for $\wedge, \neg$ in the proof of (6) are harmless. Only the $\forall$-step is interesting. The reader should recall the definition of $(\forall x\alpha)^\sigma$ page 60 and realize that the induction hypothesis refers to an arbitrary global substitution τ.

$$\begin{aligned}
\mathcal{M} \vDash (\forall x\alpha)^\sigma &\Leftrightarrow \mathcal{M} \vDash \forall x\, \alpha^\tau && (x^\tau = x \text{ and } y^\tau = y^\sigma \text{ else})\\
&\Leftrightarrow \mathcal{M}_x^a \vDash \alpha^\tau \text{ for all } a && (\text{definition})\\
&\Leftrightarrow (\mathcal{M}_x^a)^\tau \vDash \alpha \text{ for all } a && (\text{induction hypothesis})\\
&\Leftrightarrow (\mathcal{M}^\sigma)_x^a \vDash \alpha \text{ for all } a && \big((\mathcal{M}_x^a)^\tau = (\mathcal{M}^\sigma)_x^a, \text{ see below}\big)\\
&\Leftrightarrow \mathcal{M}^\sigma \vDash \forall x\alpha.
\end{aligned}$$

We show that $(\mathcal{M}_x^a)^\tau = (\mathcal{M}^\sigma)_x^a$. Since $\forall x\alpha, \sigma$ (hence $\forall x\alpha, \frac{y^\sigma}{y}$ for every y) are collision-free, we have $x \notin \mathit{var}\, y^\sigma$ if $y \neq x$, and since $y^\tau = y^\sigma$ we get in this case $y^{(\mathcal{M}_x^a)^\tau} = y^{\tau\mathcal{M}_x^a} = y^{\sigma\mathcal{M}_x^a} = y^{\sigma\mathcal{M}} = y^{\mathcal{M}^\sigma} = y^{(\mathcal{M}^\sigma)_x^a}$. But also in the case $y = x$ we have $x^{(\mathcal{M}_x^a)^\tau} = x^{\tau\mathcal{M}_x^a} = x^{\mathcal{M}_x^a} = a = x^{(\mathcal{M}^\sigma)_x^a}$. ❏

Corollary 3.6. *For all φ and $\frac{\vec{t}}{\vec{x}}$ such that $\varphi, \frac{\vec{t}}{\vec{x}}$ are collision-free, the following properties hold:*

(a) $\forall\vec{x}\varphi \vDash \varphi\,\frac{\vec{t}}{\vec{x}}$, *in particular* $\forall x\varphi \vDash \varphi\,\frac{t}{x}$, (b) $\varphi\,\frac{\vec{t}}{\vec{x}} \vDash \exists\vec{x}\varphi$,

(c) $\varphi\,\frac{s}{x}, s \equiv t \vDash \varphi\,\frac{t}{x}$, *provided* $\varphi, \frac{s}{x}, \frac{t}{x}$ *are collision-free.*

Proof. Let $\mathcal{M} \vDash \forall\vec{x}\varphi$, so that $\mathcal{M}_{\vec{x}}^{\vec{a}} \vDash \varphi$ for all $\vec{a} \in A^n$. In particular, $\mathcal{M}_{\vec{x}}^{\vec{t}^{\mathcal{M}}} \vDash \varphi$. Therefore, $\mathcal{M} \vDash \varphi\,\frac{\vec{t}}{\vec{x}}$ by Theorem 3.5. (b) follows easily from $\neg\exists\vec{x}\varphi \vDash \neg\varphi\,\frac{\vec{t}}{\vec{x}}$. This holds by (a), for $\neg\exists\vec{x}\varphi \equiv \forall\vec{x}\neg\varphi$ and $\neg(\varphi\,\frac{\vec{t}}{\vec{x}}) \equiv (\neg\varphi)\,\frac{\vec{t}}{\vec{x}}$.

(c): Let $\mathcal{M} \vDash \varphi\,\frac{s}{x}, s = t$, so that $s^{\mathcal{M}} = t^{\mathcal{M}}$ and $\mathcal{M}_x^{s^{\mathcal{M}}} \vDash \varphi$ by the theorem. Clearly, then also $\mathcal{M}_x^{t^{\mathcal{M}}} \vDash \varphi$. Hence $\mathcal{M} \vDash \varphi\,\frac{t}{x}$. ❑

Remark 2. The identical substitution ι is obviously collision-free with every formula. Thus, $\forall x\varphi \vDash \varphi\ (= \varphi^{\iota})$ is always the case, while $\forall x\varphi \vDash \varphi\,\frac{t}{x}$ is correct in general only if t contains at most the variable x, since $\varphi, \frac{t}{x}$ are then collision-free. Theorem 3.5 and Corollary 3.6 are easily strengthened. Define inductively a ternary predicate 't is free for x in φ', which intuitively is to mean that no free occurrence in φ of the variable x lies within the scope of a prefix $\forall y$ whenever $y \in var\,t$. In this case Theorem 3.5 holds for $\sigma = \frac{t}{x}$ as well, so that nothing needs to be changed in the proofs based on this theorem if one works with 't is free for x in φ', or simply reads "$\varphi, \frac{t}{x}$ are collision-free" as "t is free for x in φ." Though collision-freeness is somewhat cruder and slightly more restrictive, it is for all that more easily manageable, which will pay off, for example, in **6.2**, where proofs will be arithmetized. Once one has become accustomed to the required caution, it is allowable not always to state explicitly the restrictions caused by collisions of variables, but rather to assume them tacitly.

Theorem 3.5 also shows that the quantifier "there exists exactly one," denoted by $\exists!$, is correctly defined by $\exists!x\varphi := \exists x\varphi \wedge \forall x\forall y(\varphi \wedge \varphi\,\frac{y}{x} \to x = y)$ with $y \notin var\,\varphi$. Indeed, it is easily seen that $\mathcal{M} \vDash \forall x\forall y(\varphi \wedge \varphi\,\frac{y}{x} \to x = y)$ means just $\mathcal{M}_x^a \vDash \varphi$ & $\mathcal{M}_y^b \vDash \varphi\,\frac{y}{x} \Rightarrow a = b$. In short, $\mathcal{M}_x^a \vDash \varphi$ for at most one a. Putting everything together, $\mathcal{M} \vDash \exists!x\varphi$ iff there is precisely one $a \in A$ with $\mathcal{M}_x^a \vDash \varphi$. An example is $\mathcal{M} \vDash \exists!x\, x = t$ for arbitrary $\mathcal{M}$ and $x \notin var\,t$. In other words, $\exists!x\, x = t$ is a tautology. Half of this, namely $\vDash \exists x\, x \equiv t$, was shown in Example 1, and $\vDash \forall x\forall y(x \equiv t \wedge y \equiv t \to x \equiv y)$ is obvious. There are various equivalent definitions of $\exists!x\varphi$. For example, a short and catchy formula is $\exists x\forall y(\varphi\,\frac{y}{x} \leftrightarrow x = y)$, where $y \notin var\,\varphi$. The equivalence proof is left to the reader.

Exercises

1. Let $X \vDash \varphi$ and $x \notin free\,X$. Show that $X \vDash \forall x\varphi$.

2. Prove that $\forall x(\alpha \to \beta) \vDash \forall x\alpha \to \forall x\beta$, which is obviously equivalent to $\vDash \forall x(\alpha \to \beta) \to \forall x\alpha \to \forall x\beta$.

3. Suppose $\mathcal{A}'$ results from $\mathcal{A}$ by adjoining a constant symbol $\boldsymbol{a}$ for some $a \in A$. Prove $\mathcal{A} \vDash \alpha\,[a] \Leftrightarrow \mathcal{A}' \vDash \alpha(\boldsymbol{a})\ (= \alpha\,\frac{\boldsymbol{a}}{x})$ for $\alpha = \alpha(x)$, by first verifying $t(x)^{\mathcal{A},a} = t(\boldsymbol{a})^{\mathcal{A}'}$. This is easily generalized to the case of more than one free variable in α.

4. Show that (a) A conjunction of the $\exists_i$ and their negations is equivalent to $\exists_n \wedge \neg\exists_m$ for suitable n, m ($\exists_n \wedge \neg\exists_0 \equiv \exists_n$, $\exists_1 \wedge \neg\exists_m \equiv \neg\exists_m$). (b) A Boolean combination of the $\exists_i$ is equivalent to $\bigvee_{\nu\leqslant n} \exists_{=k_\nu}$ or to $\exists_k \vee \bigvee_{\nu\leqslant n} \exists_{=k_\nu}$, with $k_0 < \cdots < k_n < k$. Note that $\bigvee_{\nu\leqslant n} \exists_{=k_\nu}$ equals $\exists_{=0}$ ($\equiv \bot$) for $n{=}k_0{=}0$ and $\neg\exists_n \equiv \bigvee_{\nu<n} \exists_{=\nu}$ for $n{>}0$.

2.4 General Validity and Logical Equivalence

From the perspective of predicate logic $\alpha \vee \neg\alpha$ ($\alpha \in \mathcal{L}$) is a trivial example of a tautology, because it results by inserting α for p from the propositional tautology $p \vee \neg p$. Every propositional tautology provides generally valid $\mathcal{L}$-formulas by the insertion of $\mathcal{L}$-formulas for the propositional variables. But there are tautologies not arising in this way. $\forall x(x < x \vee x \not< x)$ is an example, though it has still a root in propositional logic. Tautologies without a such a root are $\exists x\, x = x$ and $\exists x\, x = t$ for $x \notin var\, t$. The former arises from the convention that structures are always nonempty, the latter from the restriction to totally defined basic operations. A particularly interesting tautology is given by the following

Example 1 (Russell's antinomy). We will show that the "Russellian set" u, consisting of all sets not containing themselves as a member, does not exist which clearly follows from $\vDash \neg\exists u\forall x(x \in u \leftrightarrow x \notin x)$. We start with $\forall x(x \in u \leftrightarrow x \notin x) \vDash u \in u \leftrightarrow u \notin u$. This holds by Corollary 3.6(a). Clearly, $u \in u \leftrightarrow u \notin u$ is unsatisfiable. Hence, the same holds for $\forall x(x \in u \leftrightarrow x \notin x)$, and thus for $\exists u\forall x(x \in u \leftrightarrow x \notin x)$. Consequently, $\vDash \neg\exists u\forall x(x \in u \leftrightarrow x \notin x)$.

Note that we need not assume in the above argument that $\in$ means membership. The proof of $\vDash \neg\exists u\forall x(x \in u \leftrightarrow x \notin x)$ need not be related to set theory at all. Hence, our example represents rather a logical paradox than a set-theoretic antinomy. What looks like an antinomy here is the expectation that $\exists u\forall x(x \in u \leftrightarrow x \notin x)$ *should* hold in set theory if $\in$ is to mean membership and Cantor's definition of a set is taken literally.

The satisfaction clause for $\alpha \to \beta$ easily yields $\alpha \vDash \beta \;\Leftrightarrow\; \vDash \alpha \to \beta$, a special case of $X, \alpha \vDash \beta \;\Leftrightarrow\; X \vDash \alpha \to \beta$. This can be very useful in checking whether formulas given in implicative form are tautologies, as was mentioned already in **1.3**. For instance, from $\forall x\alpha \vDash \alpha\,\frac{t}{x}$ (which holds for collision-free $\alpha, \frac{t}{x}$) we immediately get $\vDash \forall x\alpha \to \alpha\,\frac{t}{x}$.

As in propositional logic, $\alpha \equiv \beta$ is again equivalent to $\vDash \alpha \leftrightarrow \beta$. By inserting $\mathcal{L}$-formulas for the variables of a propositional equivalence one automatically procures one of predicate logic. Thus, for instance, $\alpha \to \beta \equiv \neg\alpha \vee \beta$, because certainly $p \to q \equiv \neg p \vee q$. Since every $\mathcal{L}$-formula results from the insertion of propositionally irreducible $\mathcal{L}$-formulas in a formula of propositional logic, one also sees that every $\mathcal{L}$-formula can be converted into a conjunctive normal form. But there are also numerous other equivalences, for example $\neg\forall x\alpha \equiv \exists x\neg\alpha$ and $\neg\exists x\alpha \equiv \forall x\neg\alpha$. The first of these means just $\neg\forall x\alpha \equiv \neg\forall x\neg\neg\alpha$ $(= \exists x\neg\alpha)$, obtained by replacing α by the equivalent formula $\neg\neg\alpha$ under the prefix $\forall x$. This is a simple application of Theorem 4.1 below with $\equiv$ for $\approx$.

As in propositional logic, semantic equivalence is an equivalence relation in $\mathcal{L}$ and, moreover, a *congruence in* $\mathcal{L}$. Speaking more generally, an equivalence relation $\approx$ in $\mathcal{L}$ satisfying the congruence property

$$\text{CP:}\quad \alpha \approx \alpha',\ \beta \approx \beta' \ \Rightarrow\ \alpha\wedge\beta \approx \alpha'\wedge\beta',\ \neg\alpha \approx \neg\alpha',\ \forall x\alpha \approx \forall x\alpha'$$

is termed a *congruence in* $\mathcal{L}$. Its most important property is expressed by

Theorem 4.1 (Replacement theorem). *Let $\approx$ be a congruence in $\mathcal{L}$ and $\alpha \approx \alpha'$. If φ' results from φ by replacing the formula α at one or more of its occurrences in φ by the formula α', then $\varphi \approx \varphi'$.*

Proof by induction on φ. Suppose φ is a prime formula. Both for $\varphi = \alpha$ and $\varphi \neq \alpha$, $\varphi \approx \varphi'$ clearly holds. Now let $\varphi = \varphi_1 \wedge \varphi_2$. In case $\varphi = \alpha$ holds trivially $\varphi \approx \varphi'$. Otherwise $\varphi' = \varphi_1' \wedge \varphi_2'$, where φ_1', φ_2' result from φ_1, φ_1 by possible replacements. By the induction hypothesis $\varphi_1 \approx \varphi_1'$ and $\varphi_2 \approx \varphi_2'$. Hence, $\varphi = \varphi_1 \wedge \varphi_2 \approx \varphi_1' \wedge \varphi_2' = \varphi'$ according to CP above. The induction steps for $\neg$, $\forall$ follow analogously. ❏

This theorem will constantly be used, mainly with $\equiv$ for $\approx$, without actually specifically being cited, just as in the arithmetical rearrangement of terms, where the laws of arithmetic used are hardly ever named explicitly. The theorem readily implies that CP is provable for all defined connectives such as $\to$ and $\exists$. For example, $\alpha \approx \alpha' \Rightarrow \exists x\alpha \approx \exists x\alpha'$, because $\alpha \approx \alpha' \Rightarrow \exists x\alpha = \neg\forall x\neg\alpha \approx \neg\forall x\neg\alpha' = \exists x\alpha'$.

First-order languages have a finer structure than those of propositional logic. There are consequently further interesting congruences in $\mathcal{L}$. In particular, formulas α, β are *equivalent in an $\mathcal{L}$-structure $\mathcal{A}$*, in symbols

$\alpha \equiv_{\mathcal{A}} \beta$, if $\mathcal{A} \vDash \alpha[w] \Leftrightarrow \mathcal{A} \vDash \beta[w]$, for all w. Hence, in $\mathcal{A} = (\mathbb{N}, <, +, 0)$ the formulas $x < y$ and $\exists z\,(z \neq 0 \wedge x + z = y)$ are equivalent. The proof of CP for $\equiv_{\mathcal{A}}$ is very simple and is therefore left to the reader.

Clearly, $\alpha \equiv_{\mathcal{A}} \beta$ is equivalent to $\mathcal{A} \vDash \alpha \leftrightarrow \beta$. Because of $\equiv \subseteq \equiv_{\mathcal{A}}$, properties such as $\neg\forall x\alpha \equiv \exists x\neg\alpha$ carry over from $\equiv$ to $\equiv_{\mathcal{A}}$. But there are often new interesting equivalences in certain structures. For instance, there are structures in which every formula is equivalent to a formula without quantifiers, as we will see in **5.6**.

A very important fact with an almost trivial proof is that the intersection of a family of congruences is itself a congruence. Consequently, for any class $\boldsymbol{K} \neq \emptyset$ of $\mathcal{L}$-structures, $\equiv_{\boldsymbol{K}} := \bigcap\{\equiv_{\mathcal{A}} | \, \mathcal{A} \in \boldsymbol{K}\}$ is necessarily a congruence. For the class $\boldsymbol{K}$ of *all* $\mathcal{L}$-structures, $\equiv_{\boldsymbol{K}}$ equals the logical equivalence $\equiv$, which in this section we deal with exclusively. Below we list its most important features; these should be committed to memory, since they will continually be applied.

(1) $\forall x(\alpha \wedge \beta) \equiv \forall x\alpha \wedge \forall x\beta$, (2) $\exists x(\alpha \vee \beta) \equiv \exists x\alpha \vee \exists x\beta$,

(3) $\forall x\forall y\alpha \equiv \forall y\forall x\alpha$, (4) $\exists x\exists y\alpha \equiv \exists y\exists x\alpha$.

If x does not occur free in the formula β, then also

(5) $\forall x(\alpha \vee \beta) \equiv \forall x\alpha \vee \beta$, (6) $\exists x(\alpha \wedge \beta) \equiv \exists x\alpha \wedge \beta$,

(7) $\forall x\beta \equiv \beta$, (8) $\exists x\beta \equiv \beta$,

(9) $\forall x(\alpha \to \beta) \equiv \exists x\alpha \to \beta$, (10) $\exists x(\alpha \to \beta) \equiv \forall x\alpha \to \beta$.

The simple proofs are left to the reader. (7) and (8) were stated in (2) in **2.3**. Only (9) and (10) look at first sight surprising. But in practice these equivalences are very frequently used. For instance, consider for a fixed set of formulas X the evidently true metalogical assertion 'for all α: if $X \vDash \alpha, \neg\alpha$ then $X \vDash \forall x\, x \neq x$'. This clearly states the same as 'If there is some α such that $X \vDash \alpha, \neg\alpha$ then $X \vDash \forall x\, x \neq x$'.

Remark. In everyday speech variables tend to remain unquantified, partly because in some cases the same meaning results from quantifying with "there exists a" as with "for all." For instance, consider the following three sentences, which obviously tell us the same thing, and of which the last two correspond to the logical equivalence (9):

- If a lawyer finds a loophole in the law it must be changed.
- If there is a lawyer who finds a loophole in the law it must be changed.
- For all lawyers: if one of them finds a loophole in the law then it must be changed.

Often, the type of quantification in linguistic bits of information can be made out only from the context, and this leads not all too seldom to unintentional (or intentional) misunderstandings. "Logical relations in language are almost always just alluded to, left to guesswork, and not actually expressed" (G. Frege).

Let x, y be distinct variables and $\alpha \in \mathcal{L}$. One of the most important logical equivalences is *renaming of bound variables* (in short, *bound renaming*), stated in

(11) (a) $\forall x\alpha \equiv \forall y(\alpha\,\frac{y}{x})$, (b) $\exists x\alpha \equiv \exists y(\alpha\,\frac{y}{x})$ $(y \notin \mathit{var}\,\alpha)$.

(b) follows from (a) by rearranging equivalently. Note that $y \notin \mathit{var}\,\alpha$ is equivalent to $y \notin \mathit{free}\,\alpha$ and $\alpha, \frac{y}{x}$ collision-free. Writing $\mathcal{M}_x^y$ for $\mathcal{M}_x^{y^{\mathcal{M}}}$, (a) derives as follows:

$$\begin{array}{rll} \mathcal{M} \vDash \forall x\alpha & \Leftrightarrow\ \mathcal{M}_x^a \vDash \alpha & \text{for all } a \quad \text{(definition)} \\ & \Leftrightarrow\ (\mathcal{M}_y^a)_x^a \vDash \alpha & \text{for all } a \quad \text{(Theorem 3.1)} \\ & \Leftrightarrow\ (\mathcal{M}_y^a)_x^y \vDash \alpha & \text{for all } a \quad \big((\mathcal{M}_y^a)_x^y = (\mathcal{M}_y^a)_x^a\big) \\ & \Leftrightarrow\ \mathcal{M}_y^a \vDash \alpha\,\frac{y}{x} & \text{for all } a \quad \text{(Theorem 3.5)} \\ & \Leftrightarrow\ \mathcal{M} \vDash \forall y(\alpha\tfrac{y}{x})\,. \end{array}$$

(12) and (13) below are also noteworthy. According to (13), substitutions are completely described up to logical equivalence by so-called *free renamings* (substitutions of the form $\frac{y}{x}$). (13) also embraces the case $x \in \mathit{var}\,t$. In (12) and (13) we tacitly assume that $\alpha, \frac{t}{x}$ are collision-free.

(12) $\forall x(x \mathbin{=} t \to \alpha) \ \equiv \alpha\,\frac{t}{x} \equiv \exists x(x \mathbin{=} t \wedge \alpha)$ $(x \notin \mathit{var}\,t)$.

(13) $\forall y(y \mathbin{=} t \to \alpha\,\frac{y}{x}) \equiv \alpha\,\frac{t}{x} \equiv \exists y(y \mathbin{=} t \wedge \alpha\,\frac{y}{x})$ $(y \notin \mathit{var}\,\alpha, t)$.

Proof of (12): $\forall x(x \mathbin{=} t \to \alpha) \vDash (x \mathbin{=} t \to \alpha)\,\frac{t}{x} = t \mathbin{=} t \to \alpha\,\frac{t}{x} \vDash \alpha\,\frac{t}{x}$ by Corollary 3.6. Conversely, let $\mathcal{M} \vDash \alpha\,\frac{t}{x}$. If $\mathcal{M}_x^a \vDash x \mathbin{=} t$ then clearly $a = t^{\mathcal{M}}$. Hence also $\mathcal{M}_x^a \vDash \alpha$, since $\mathcal{M}_x^{t^{\mathcal{M}}} \vDash \alpha$. Thus, $\mathcal{M}_x^a \vDash x \mathbin{=} t \to \alpha$ for any $a \in A$, i.e., $\mathcal{M} \vDash \forall x(x \mathbin{=} t \to \alpha)$. This proves the left equivalence in (12). The right equivalence reduces to the left one because

$$\exists x(x \mathbin{=} t \wedge \alpha) = \neg\forall x\neg(x \mathbin{=} t \wedge \alpha) \equiv \neg\forall x(x \mathbin{=} t \to \neg\alpha) \equiv \neg\neg\alpha\,\tfrac{t}{x} \equiv \alpha\,\tfrac{t}{x}\,.$$

Item (13) is proved similarly. Note that $\forall y(y \mathbin{=} t \to \alpha\,\frac{y}{x}) \vDash \alpha\,\frac{y}{x}\,\frac{t}{y} = \alpha\,\frac{t}{x}$ by Corollary 3.6 and Exercise 4 in **2.2**.

With the above equivalences we can now regain an equivalent formula starting with any formula in which all quantifiers are standing at the beginning. But this result requires both quantifiers $\forall$ and $\exists$, in the following denoted by $\mathtt{Q}, \mathtt{Q}_1, \mathtt{Q}_2, \ldots$

A formula of the form $\alpha = \mathtt{Q}_1 x_1 \cdots \mathtt{Q}_n x_n \beta$ with an open formula β is termed a *prenex formula* or a *prenex normal form*, in short, a PNF. β is called the *kernel* of α. W.l.o.g. $x_1, \ldots, x_n$ are distinct and x_i occurs free in β since we may drop "superfluous quantifiers," see (2) page 66. Prenex normal forms are very important for classifying definable number-theoretic predicates in **6.3**, and for other purposes. The already mentioned $\forall$- and $\exists$-formulas are the simplest examples.

Theorem 4.2 (on the prenex normal form). *Every formula φ is equivalent to a formula in prenex normal form that can effectively be constructed from φ.*

Proof. Without loss of generality let φ contain only the logical symbols $\neg, \wedge, \forall, \exists$ (besides $=$). For each prefix $\mathtt{Q}x$ in φ consider the number of symbols $\neg$ or $\wedge$ occurring to the left of $\mathtt{Q}x$. Let $s\varphi$ be the sum of these numbers, summed over all prefixes occurring in φ. Clearly, φ is a PNF iff $s\varphi = 0$. Let $s\varphi \neq 0$. Then φ contains some prefix $\mathtt{Q}x$ and $\neg$ or $\wedge$ stands immediately in front of $\mathtt{Q}x$. A successive application of either

$$\neg\forall x\alpha \equiv \exists x\neg\alpha,\ \neg\exists x\alpha \equiv \forall x\neg\alpha,\ \text{ or } \beta \wedge \mathtt{Q}x\alpha \equiv \mathtt{Q}y(\ b \wedge \alpha\tfrac{y}{x})\ (y \notin \mathit{var}\,\alpha, \beta),$$

inside φ obviously reduces $s\varphi$ stepwise. ❑

Example 2. $\forall x\exists y(x \neq 0 \to x \cdot y = 1)$ is a PNF for $\forall x(x \neq 0 \to \exists y\, x \cdot y = 1)$. And $\exists x\forall y\forall z(\varphi \wedge (\varphi\frac{y}{x} \wedge \varphi\frac{z}{x} \to y = z))$ for $\exists x\varphi \wedge \forall y\forall z(\varphi\frac{y}{x} \wedge \varphi\frac{z}{x} \to y = z)$, provided $y, z \notin \mathit{free}\,\varphi$; if not, a bound renaming will help. An equivalent PNF for this formula with minimal quantifier rank is $\exists x\forall y(\varphi\frac{y}{x} \leftrightarrow x = y)$.

The formula $\forall x(x \neq 0 \to \exists y\, x \cdot y = 1)$ from Example 2 may be abbreviated by $(\forall x \neq 0)\exists y\, x \cdot y = 1$. More generally, we shall often write $(\forall x \neq t)\alpha$ for $\forall x(x \neq t \to \alpha)$ and $(\exists x \neq t)\alpha$ for $\exists x(x \neq t \wedge \alpha)$. A similar notation is used for $\leqslant$, $<$, $\in$ and their negations. For instance, $(\forall x \leqslant t)\alpha$ and $(\exists x \leqslant t)\alpha$ are to mean $\forall x(x \leqslant t \to \alpha)$ and $\exists x(x \leqslant t \wedge \alpha)$, respectively. For any binary relation symbol $\lhd$, the "prefixes" $(\forall y \lhd x)$ and $(\exists y \lhd x)$ are related to each other, as are $\forall$ and $\exists$, see Exercise 2.

Exercises

1. Let $\alpha \equiv \beta$. Prove that $\alpha\frac{\vec{t}}{\vec{x}} \equiv \beta\frac{\vec{t}}{\vec{x}}$ ($\alpha, \frac{\vec{t}}{\vec{x}}$ and $\beta, \frac{\vec{t}}{\vec{x}}$ collision-free).
2. Prove that $\neg(\forall x \lhd y)\alpha \equiv (\exists x \lhd y)\neg\alpha$ and $\neg(\exists x \lhd y)\alpha \equiv (\forall x \lhd y)\neg\alpha$. Here $\lhd$ represents any binary relation symbol.

3. Show by means of bound renaming that both the conjunction and the disjunction of $\forall$-formulas α, β is equivalent to some $\forall$-formula. Prove the same for $\exists$-formulas.

4. Show that every formula $\varphi \in \mathcal{L}$ is equivalent to some $\varphi' \in \mathcal{L}$ built up from literals by means of $\wedge$, $\vee$, and $\exists$.

5. Let P be a unary predicate symbol. Prove that $\exists x(Px \rightarrow \forall y Py)$ is a tautology.

6. Call $\alpha, \beta \in \mathcal{L}$ *tautologically equivalent* if $\vDash \alpha \Leftrightarrow \vDash \beta$. Confirm that the following (in general not logically equivalent) formulas are tautologically equivalent: α, $\forall x \alpha$, and $\alpha \frac{c}{x}$, where the constant symbol c does not occur in α.

2.5 Logical Consequence and Theories

Whenever $\mathcal{L}' \supseteq \mathcal{L}$, the language $\mathcal{L}'$ is called an *expansion* or *extension* of $\mathcal{L}$ and $\mathcal{L}$ a *reduct* or *restriction* of $\mathcal{L}'$. Recall the insensitivity of the consequence relation to extensions of a first-order language, mentioned in **2.3**. Theorem 3.1 yields that establishing $X \vDash \alpha$ does not depend on the language to which the set of formulas X and the formula α belong. For this reason, indices for $\vDash$, such as $\vDash_{\mathcal{L}}$, are dispensable.

Because of the unaltered satisfaction conditions for $\wedge$ and $\neg$, all properties of the propositional consequence gained in **1.3** carry over to the first-order logical consequence relation. These include general properties such as, for example, the reflexivity and transitivity of $\vDash$, and the semantic counterparts of the rules $(\wedge 1)$, $(\wedge 2)$, $(\neg 1)$, $(\neg 2)$ from **1.4**, for instance the counterpart of $(\wedge 1)$, $\dfrac{X \vDash \alpha, \beta}{X \vDash \alpha \wedge \beta}$.[5]

In addition, Gentzen-style properties such as the deduction theorem automatically carry over. But there are also completely new properties. Some of these will be elevated to basic rules of a logical calculus for first-order languages in **3.1**, to be found among the following ones:

[5] A suggestive way of writing "$X \vDash \alpha, \beta$ implies $X \vDash \alpha \wedge \beta$," a notation that was introduced already in Exercise 3 in **1.3**. A corresponding notation will also be used in stating the properties of $\vDash$ on the next page.

Some properties of the predicate logical consequence relation.

(a) $\dfrac{X \vDash \forall x\alpha}{X \vDash \alpha\,\frac{t}{x}}$ ($\alpha, \frac{t}{x}$ collision-free),

(b) $\dfrac{X \vDash \alpha\,\frac{s}{x}, s = t}{X \vDash \alpha\,\frac{t}{x}}$ ($\alpha, \frac{s}{x}$ and $\alpha, \frac{t}{x}$ collision-free),

(c) $\dfrac{X, \beta \vDash \alpha}{X, \forall x\beta \vDash \alpha}$ (anterior generalization),

(d) $\dfrac{X \vDash \alpha}{X \vDash \forall x\alpha}$ ($x \notin \mathit{free}\, X$, posterior generalization),

(e) $\dfrac{X, \beta \vDash \alpha}{X, \exists x\beta \vDash \alpha}$ ($x \notin \mathit{free}\, X, \alpha$, anterior particularization),

(f) $\dfrac{X \vDash \alpha\,\frac{t}{x}}{X \vDash \exists x\alpha}$ ($\alpha, \frac{t}{x}$ collision-free, posterior particularization)

(a) follows from $X \vDash \forall x\alpha \vDash \alpha\,\frac{t}{x}$, for $\vDash$ is transitive. Similarly, (b) follows from $\alpha\,\frac{s}{x}, s = t \vDash \alpha\,\frac{t}{x}$, stated in Corollary 3.6. Analogously (c) results from $\forall x\beta \vDash \beta$. To prove (d), suppose that $X \vDash \alpha$, $\mathcal{M} \vDash X$, and $x \notin \mathit{free}\, X$. Then $\mathcal{M}^a_x \vDash X$ for any $a \in A$ by Theorem 3.1, which just means $\mathcal{M} \vDash \forall x\alpha$. As regards (e), let $X, \beta \vDash \alpha$. Observe that by contraposition and by (d),

$$X, \beta \vDash \alpha \;\Rightarrow\; X, \neg\alpha \vDash \neg\beta \;\Rightarrow\; X, \neg\alpha \vDash \forall x\neg\beta,$$

whence $X, \neg\forall x\neg\beta \vDash \alpha$. (e) captures deduction *from* an existence claim, while (f) *confirms* an existence claim. (f) holds since $\alpha\,\frac{t}{x} \vDash \exists x\alpha$ according to Corollary 3.6. Both (e) and (f) are permanently applied in mathematical reasoning and will briefly be discussed in Example 1 on the next page. All above properties have certain variants; for example, a variant of (d) is

(g) $\dfrac{X \vDash \alpha\,\frac{y}{x}}{X \vDash \forall x\alpha}$ ($y \notin \mathit{free}\, X \cup \mathit{var}\, \alpha$).

This results from (d) with $\alpha\,\frac{y}{x}$ for α and y for x, since $\forall y\alpha\,\frac{y}{x} \equiv \forall x\alpha$.

From the above properties, complicated chains of deduction can, where necessary, be justified step by step. But in practice this makes sense only in particular circumstances, because formalized proofs are readable only at the expense of a lot of time, just as with lengthy computer programs, even with well-prepared documentation. What is most important is that a proof, when written down, can be understood and reproduced. This is why mathematical deduction tends to proceed *informally*, i.e., both claims and

their proofs are formulated in a mathematical "everyday" language with the aid of fragmentary and flexible formalization. To what degree a proof is to be formalized depends on the situation and need not be determined in advance. In this way the strict syntactic structure of formal proofs is slackened, compensating for the imperfection of our brains in regard to processing syntactic information.

Further, certain informal proof methods will often be described by a more or less clear reference to so-called background knowledge, and not actually carried out. This method has proven itself to be sufficiently reliable. As a matter of fact, apart from specific cases it has not yet been bettered by any of the existing automatic proof machines. Let us present a very simple example of an informal proof in a language $\mathcal{L}$ for natural numbers that along with $0, 1, +, \cdot$ contains the symbol $|$ for divisibility, defined by $m|n \Leftrightarrow \exists k\, m \cdot k = n$. In addition, let $\mathcal{L}$ contain a symbol f for some given function from $\mathbb{N}$ to $\mathbb{N}$. We need no closer information on this function, but we shall write f_i for $\mathsf{f}(i)$ in Example 1.

Example 1. We want to prove $\forall n \exists x (\forall i \leqslant n) \mathsf{f}_i | x$. That is, for every n, $\mathsf{f}_0, \ldots, \mathsf{f}_n$ have a common multiple. A careful proof proceeds by induction on n. Here we focus solely on $X, \exists x (\forall i \leqslant n) \mathsf{f}_i | x \vDash \exists x (\forall i \leqslant n+1) \mathsf{f}_i | x$, the induction step. X represents our prior knowledge about familiar properties of divisibility. Informally we reason as follows: Suppose $\exists x (\forall i \leqslant n) \mathsf{f}_i | x$ and let x denote any common multiple of $\mathsf{f}_0, \ldots, \mathsf{f}_n$. Then $x \cdot \mathsf{f}_{n+1}$ is clearly a common multiple of $\mathsf{f}_0, \ldots, \mathsf{f}_{n+1}$, hence $\exists x (\forall i \leqslant n+1) \mathsf{f}_i | x$. That's all. To argue here formally like a proof machine, let us start from the obvious $(\forall i \leqslant n) \mathsf{f}_i | x \vDash (\forall i \leqslant n+1) \mathsf{f}_i | (x \cdot \mathsf{f}_{n+1})$. Posterior particularization of x yields $X, (\forall i \leqslant n) \mathsf{f}_i | x \vDash \exists x (\forall i \leqslant n+1) \mathsf{f}_i | x$. From this follows the desired $X, \exists x (\forall i \leqslant n) \mathsf{f}_i | x \vDash \exists x (\forall i \leqslant n+1) \mathsf{f}_i | x$ by anterior particularization. Thus, formalizing a nearly trivial informal argument may need a lot of writing and turns out to be nontrivial in some sense.

Some textbooks deal with a somewhat stricter consequence relation, which we denote here by $\overset{g}{\vDash}$. The reason is that in mathematics one largely considers derivations in theories. For $X \subseteq \mathcal{L}$ and $\varphi \in \mathcal{L}$ define $X \overset{g}{\vDash} \varphi$ if $\mathcal{A} \vDash X \Rightarrow \mathcal{A} \vDash \varphi$, for all $\mathcal{L}$-structures $\mathcal{A}$. In contrast to $\vDash$, which may be called the *local* consequence relation, $\overset{g}{\vDash}$ can be considered as the *global* consequence relation since it cares only about $\mathcal{A}$, not about a concrete valuation w in $\mathcal{A}$ as does $\vDash$.

Let us collect a few properties of $\overset{g}{\models}$. Obviously, $X \models \varphi$ implies $X \overset{g}{\models} \varphi$, but the converse does not hold in general. For example, $x = y \overset{g}{\models} \forall xy\, x = y$, but $x = y \not\models \forall xy\, x = y$. By (d) from page 79, $X \models \varphi \Rightarrow X \models \varphi^g$ holds in general only if the free variables of φ do not occur free in X, while $X \overset{g}{\models} \varphi \Rightarrow X \overset{g}{\models} \varphi^g$ (hence $\varphi \overset{g}{\models} \varphi^g$) holds unrestrictedly. A reduction of $\overset{g}{\models}$ to $\models$ is provided by the following equivalence, which easily follows from $\mathcal{M} \models X^g \;\Leftrightarrow\; \mathcal{A} \models X^g$, for each model $\mathcal{M} = (\mathcal{A}, w)$:

(1) $\quad X \overset{g}{\models} \varphi \;\Leftrightarrow\; X^g \models \varphi.$

Because of $S^g = S$ for sets of sentences S, we clearly obtain from (1)

(2) $\quad S \overset{g}{\models} \varphi \;\Leftrightarrow\; S \models \varphi \quad (S \subseteq \mathcal{L}^0).$

In particular, $\overset{g}{\models} \varphi \Leftrightarrow \models \varphi$. Thus, a distinction between $\models$ and $\overset{g}{\models}$ is apparent only when premises are involved that are not sentences. In this case the relation $\overset{g}{\models}$ must be treated with the utmost care. Neither the rule of case distinction $\dfrac{X, \alpha \overset{g}{\models} \beta \;\big|\; X, \neg\alpha \overset{g}{\models} \beta}{X \overset{g}{\models} \beta}$ nor the deduction theorem $\dfrac{X, \alpha \overset{g}{\models} \beta}{X \overset{g}{\models} \alpha \to \beta}$ is unrestrictedly correct. For example $x = y \overset{g}{\models} \forall xy\, x = y$, but it is false that $\overset{g}{\models} x = y \to \forall xy\, x = y$. This means that the deduction theorem fails to hold for the relation $\overset{g}{\models}$. It holds only under certain restrictions.

One of the reasons for our preference of $\models$ over $\overset{g}{\models}$ is that $\models$ extends the propositional consequence relation conservatively, so that features such as the deduction theorem carry over unrestrictedly, while this is not the case for $\overset{g}{\models}$. It should also be said that $\overset{g}{\models}$ does not reflect the actual procedures of natural deduction in which formulas with free variables are frequently used also in deductions of sentences from sentences, for instance in Example 1.

We now make more precise the notion of a formalized theory in $\mathcal{L}$, where it is useful to think of the examples in **2.3**, such as group theory. Again, the definitions by different authors may look somewhat differently.

Definition. An *elementary theory* or *first-order theory* in $\mathcal{L}$, also termed an *$\mathcal{L}$-theory*, is a set of sentences $T \subseteq \mathcal{L}^0$ *deductively closed in* $\mathcal{L}^0$, i.e., $T \models \alpha \;\Leftrightarrow\; \alpha \in T$, for all $\alpha \in \mathcal{L}^0$. If $\alpha \in T$ then we say that α *is valid* or *true* or *holds in* T, or α *is a theorem* of T. The extralogical symbols of $\mathcal{L}$ are called the *symbols of* T. If $T \subseteq T'$ then T is called a *subtheory* of T', and T' an *extension* of T. An $\mathcal{L}$-structure $\mathcal{A}$ such that $\mathcal{A} \models T$ is also termed a *model of* T, briefly a *T-model*. $\operatorname{Md} T$ denotes the class of all models of T in this sense; $\operatorname{Md} T$ consist of $\mathcal{L}$-structures only.

For instance, $\{\alpha \in \mathcal{L}^0 \mid X \vDash \alpha\}$ is a theory for any set $X \subseteq \mathcal{L}$, since $\vDash$ is transitive. A theory T in $\mathcal{L}$ satisfies $T \vDash \varphi \;\Leftrightarrow\; \mathcal{A} \vDash \varphi$ for all $\mathcal{A} \vDash T$, where $\varphi \in \mathcal{L}$ is any formula. Important is also $T \vDash \varphi \;\Leftrightarrow\; T \vDash \varphi^g$. These readily confirmed facts should be taken in and remembered, since they are constantly used. Different authors may use different definitions for a theory. For example, they may not demand that theories contain sentences only, as we do. Conventions of this type each have their advantages and disadvantages. Proofs regarding theories are always adaptable enough to accommodate small modifications of the definition. Using the definition given above we set the following

Convention. In talking of the *theory* S, where S is a set of sentences, we always mean the theory determined by S, that is, $\{\alpha \in \mathcal{L}^0 \mid S \vDash \alpha\}$. A set $X \subseteq \mathcal{L}$ is called an *axiom system* for T whenever $T = \{\alpha \in \mathcal{L}^0 \mid X^g \vDash \alpha\}$, i.e., we tacitly generalize all possibly open formulas in X. We have always to think of free variables occurring in axioms as being generalized.

Thus, axioms of a theory are always sentences. But we conform to standard practice of writing long axioms as formulas. We will later consider extensive axiom systems (in particular, for arithmetic and set theory) whose axioms are partly written as open formulas just for economy.

There exists a smallest theory in $\mathcal{L}$, namely the set Taut $(= \mathsf{Taut}_{\mathcal{L}})$ of all generally valid sentences in $\mathcal{L}$, also called the "logical" theory. An axiom system for Taut is the empty set of axioms. There is also a largest theory: the set $\mathcal{L}^0$ *of all* sentences, the *inconsistent* theory, which possesses no models. All remaining theories are called *satisfiable* or *consistent*.[6] Moreover, the intersection $T = \bigcap_{i \in I} T_i$ of a nonempty family of theories T_i is in turn a theory: if $T \vDash \alpha \in \mathcal{L}^0$ then clearly $T_i \vDash \alpha$ and so $\alpha \in T_i$ for each $i \in I$, hence $\alpha \in T$ as well. In this book T and T', with or without indices, exclusively denote theories.

For $T \subseteq \mathcal{L}^0$ and $\alpha \in \mathcal{L}^0$ let $T + \alpha$ denote the smallest theory that extends T and contains α. Similarly let $T + S$ for $S \subseteq \mathcal{L}^0$ be the smallest theory containing $T \cup S$. If S is finite then $T' = T + S = T + \bigwedge S$ is called a *finite extension of* T. Here $\bigwedge S$ denotes the conjunction of all sentences in S. A sentence α is termed *compatible* or *consistent* with T if $T + \alpha$ is

[6] *Consistent* mostly refers to a logic calculus, e.g., the calculus in **3.1**. However, it will be shown in **3.2** that consistency and satisfiability of a theory coincide, thus justifying the word's ambiguous use.

satisfiable, and *refutable in* T if $T + \neg\alpha$ is satisfiable. Thus, the theory T_F of fields is *compatible* with the sentence $1 + 1 = 0$. Equivalently, $1 + 1 \neq 0$ is *refutable* in T_F, since the 2-element field satisfies $1 + 1 = 0$.

If both α and $\neg\alpha$ are compatible with T then the sentence α is termed *independent* of T. The classic example is the independence of the parallel axiom from the remaining axioms of Euclidean plane geometry, which define *absolute* geometry. Much more difficult is the independence proof of the continuum hypothesis from the axioms for set theory. These axioms are presented and discussed in **3.4**.

At this point we introduce another important concept; $\alpha, \beta \in \mathcal{L}$ are said to be *equivalent in* or *modulo* T, $\alpha \equiv_T \beta$, if $\alpha \equiv_{\mathcal{A}} \beta$ for all $\mathcal{A} \vDash T$. Being an intersection of congruences, $\equiv_T$ is itself a congruence and hence satisfies the replacement theorem. This will henceforth be used without mention, as will the obvious equivalence of $\alpha \equiv_T \beta$, $T \vDash \alpha \leftrightarrow \beta$, and of $T \vDash (\alpha \leftrightarrow \beta)^g$. A suggestive writing of $\alpha \equiv_T \beta$ would also be $\alpha \mathrel{=\!\!\!\equiv}_T \beta$.

Example 2. Let T_G be as on p. 65. Claim: $x \circ x = x \equiv_{T_G} x = e$. The only tricky proof step is $T_G \vDash x \circ x \equiv x \to x \equiv e$. Let $x \circ x \equiv x$ and choose some y with $x \circ y = e$. The claim then follows from $x = x \circ e = x \circ x \circ y = x \circ y = e$. A strict formal proof of the latter uses anterior particularization.

Another important congruence is term equivalence. Call terms s, t *equivalent* modulo (or in) T, in symbols $s \approx_T t$, if $T \vDash s = t$, that is, $\mathcal{A} \vDash s = t\,[w]$ for all $\mathcal{A} \vDash T$ and $w\colon Var \to A$. For instance, in $T = T_G^{=}$, $(x \circ y)^{-1} \equiv y^{-1} \circ x^{-1}$ is easily provable, so that $(x \circ y)^{-1} \approx_T y^{-1} \circ x^{-1}$. Another example: in the theory of fields, each term is equivalent to a polynomial in several variables with integer coefficients.

If all axioms of a theory T are $\forall$-sentences then T is called a *universal* or $\forall$-theory. Examples are partial orders, orders, rings, lattices, and Boolean algebras. For such a theory, $\operatorname{Md} T$ is closed with respect to substructures, which means $\mathcal{A} \subseteq \mathcal{B} \vDash T \Rightarrow \mathcal{A} \vDash T$. This follows at once from Corollary 3.3. Conversely, a theory closed with respect to substructures is necessarily a universal one, as will turn out in **5.4**. $\forall$-theories are further classified. The most important subclasses are equational, quasi-equational, and universal Horn theories, all of which will be considered to some extent in later chapters. Besides $\forall$-theories, the $\forall\exists$-theories (those having $\forall\exists$-sentences as axioms) are of particular interest for mathematics. More about all these theories will be said in **5.4**.

Theories are frequently given by structures or classes of structures. The elementary theory $Th\,\mathcal{A}$ and the theory $Th\,\boldsymbol{K}$ of a nonempty class $\boldsymbol{K}$ of structures are defined respectively by

$$Th\,\mathcal{A} := \{\alpha \in \mathcal{L}^0 \mid \mathcal{A} \vDash \alpha\}, \quad Th\,\boldsymbol{K} := \bigcap\{Th\,\mathcal{A} \mid \mathcal{A} \in \boldsymbol{K}\}.$$

It is easily seen that $Th\,\mathcal{A}$ and $Th\,\boldsymbol{K}$ are theories in the precise sense defined above. Instead of $\alpha \in Th\,\boldsymbol{K}$ one often writes $\boldsymbol{K} \vDash \alpha$. In general, Md $Th\,\boldsymbol{K}$ is larger than $\boldsymbol{K}$, as we shall see.

One easily confirms that the set of formulas breaks up modulo T (more precisely, modulo $\equiv_T$) into equivalence classes; their totality is denoted by $B_\omega T$. Based on these we can define in a natural manner operations $\wedge, \vee, \neg$. For instance, $\bar{\alpha} \wedge \bar{\beta} = \overline{\alpha \wedge \beta}$, where $\bar{\varphi}$ denotes the equivalence class to which φ belongs. One shows easily that $B_\omega T$ forms a Boolean algebra with respect to $\wedge, \vee, \neg$. For every n, the set $B_n T$ of all $\bar{\varphi}$ in $B_\omega T$ such that the free variables of φ belong to Var_n $(= \{\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{n-1}\})$ is a subalgebra of $B_\omega T$. Note that $B_0 T$ is isomorphic to the Boolean algebra of all sentences modulo $\equiv_T$, also called the *Tarski–Lindenbaum algebra* of T. The significance of the Boolean algebras $B_n T$ is revealed only in the somewhat higher reaches of model theory, and they are therefore mentioned only incidentally.

Exercises

1. Suppose $x \notin free\,X$ and c is not in X, α. Prove the equivalence of

   $$\text{(i) } X \vDash \alpha, \qquad \text{(ii) } X \vDash \forall x \alpha, \qquad \text{(iii) } X \vDash \alpha\,\tfrac{c}{x}.$$

   This holds then in particular if X is the axiom system of a theory or itself a theory. Then $x \notin free\,X$ is trivially satisfied.

2. Let S be a set of sentences, α and β formulas, $x \notin free\,\beta$, and let c be a constant not occurring in S, α, β. Show that

   $$S \vDash \alpha\,\tfrac{c}{x} \to \beta \;\Leftrightarrow\; S \vDash \exists x \alpha \to \beta.$$

3. Verify for all $\alpha, \beta \in \mathcal{L}^0$ that $\beta \in T + \alpha \;\Leftrightarrow\; \alpha \to \beta \in T$.

4. Let $T \subseteq \mathcal{L}$ be a theory, $\mathcal{L}_0 \subseteq \mathcal{L}$, and $T_0 := T \cap \mathcal{L}_0$. Prove that T_0 is also a theory (the so-called *reduct theory* in the language $\mathcal{L}_0$).

2.6 Explicit Definitions—Language Expansions

The deductive development of a theory, be it given by an axiom system or a single structure or classes of those, nearly always goes hand in hand with expansions of the language carried out step by step. For example, in developing elementary number theory in the language $\mathcal{L}(0, 1, +, \cdot)$, the introduction of the divisibility relation by means of the (explicit) definition $x|y \leftrightarrow \exists z\, x \cdot z = y$ has certainly advantages not only for purely technical reasons. This and similar examples motivate the following

Definition I. Let r be an n-ary relation symbol not occurring in $\mathcal{L}$. An *explicit definition of r in $\mathcal{L}$* is to mean a formula of the form

$$\eta_r: \quad r\vec{x} \leftrightarrow \delta(\vec{x})$$

with $\delta(\vec{x}) \in \mathcal{L}$ and distinct variables in $\vec{x}$, called the *defining formula*. For a theory T, the extension $T_r := T + \eta_r^g$ is then called a *definitorial extension* (or *expansion*) of T by r, more precisely, by η_r.

T_r is a theory in $\mathcal{L}[r]$, the language resulting from $\mathcal{L}$ by adjoining the symbol r. It will turn out that T_r is a *conservative extension* of T, which, in the general case, means a theory $T' \supseteq T$ in $\mathcal{L}' \supseteq \mathcal{L}$ such that $T' \cap \mathcal{L} = T$. Thus, T_r contains exactly the same $\mathcal{L}$-sentences as does T. In this sense, T_r is a harmless extension of T. Our claim constitutes part of Theorem 6.1. For $\varphi \in \mathcal{L}[r]$ define the *reduced formula* $\varphi^{rd} \in \mathcal{L}$ as follows: Starting from the left, replace every prime formula $r\vec{t}$ occurring in φ by $\delta_{\vec{x}}(\vec{t})$. Clearly, $\varphi^{rd} = \varphi$, provided r does not appear in φ.

Theorem 6.1 (Elimination theorem). *Let $T_r \subseteq \mathcal{L}[r]$ be a definitorial extension of the theory $T \subseteq \mathcal{L}^0$ by the explicit definition η_r. Then for all formulas $\varphi \in \mathcal{L}[r]$ holds the equivalence*

$$(*) \quad T_r \vDash \varphi \;\Leftrightarrow\; T \vDash \varphi^{rd}.$$

For $\varphi \in \mathcal{L}$ we get in particular $T_r \vDash \varphi \Leftrightarrow T \vDash \varphi$ (since $\varphi^{rd} = \varphi$). Hence, T_r is a conservative extension of T, i.e., $\alpha \in T_r \Leftrightarrow \alpha \in T$, for all $\alpha \in \mathcal{L}^0$.

Proof. Each $\mathcal{A} \vDash T$ is expandable to a model $\mathcal{A}' \vDash T_r$ with the same domain, setting $r^{\mathcal{A}'}\vec{a} :\Leftrightarrow \mathcal{A} \vDash \delta[\vec{a}]$ $(\vec{a} \in A^n)$. Since $r\vec{t} \equiv_{T_r} \delta(\vec{t})$ for any $\vec{t}$, we obtain $\varphi \equiv_{T_r} \varphi^{rd}$ for all $\varphi \in \mathcal{L}[r]$ by the replacement theorem. Thus, $(*)$ follows from

$$\begin{aligned} T_r \vDash \varphi &\Leftrightarrow \mathcal{A}' \vDash \varphi \text{ for all } \mathcal{A} \vDash T \quad (\mathrm{Md}\, T_r = \{\mathcal{A}' \mid \mathcal{A} \vDash T\}) \\ &\Leftrightarrow \mathcal{A}' \vDash \varphi^{rd} \text{ for all } \mathcal{A} \vDash T \quad (\text{because } \varphi \equiv_{T_r} \varphi^{rd}) \\ &\Leftrightarrow \mathcal{A} \vDash \varphi^{rd} \text{ for all } \mathcal{A} \vDash T \quad (\text{Theorem 3.1}) \\ &\Leftrightarrow T \vDash \varphi^{rd}. \quad \square \end{aligned}$$

Operation symbols and constants can be similarly introduced, though in this case there are certain conditions to observe. For instance, in T_G (see page 65) the operation $^{-1}$ is defined by $\eta:\ y = x^{-1} \leftrightarrow x \circ y = e$. This definition is legitimate, since $T_G \vDash \forall x \exists! y\, x \circ y = e$; Exercise 3. Only this requirement (which by the way is a logical consequence of η) ensures that $T_G + \eta^g$ is a conservative extension of T_G. We therefore extend Definition I as follows, keeping in mind that to the end of this section constant symbols are to be counted among the operation symbols.

Definition II. An *explicit definition* of an n-ary operation symbol f not occurring in $\mathcal{L}$ is a formula of the form

$$\eta_f: \quad y = f\vec{x} \leftrightarrow \delta(\vec{x}, y) \qquad (\delta \in \mathcal{L} \text{ and } y, x_1, \ldots, x_n \text{ distinct}).$$

η_f is called *legitimate* in $T \subseteq \mathcal{L}$ if $T \vDash \forall \vec{x} \exists! y \delta$, and $T_f := T + \eta_f^g$ is then called a *definitorial extension by* f, more precisely by η_f. In the case $n = 0$ we write c for f and speak of an *explicit definition of the constant symbol* c. Written more suggestively $y = c \leftrightarrow \delta(y)$.

Some of the free variables of δ are often not explicitly named, and thus downgraded to parameter variables. More on this will be said in the discussion of the axioms for set theory in **3.4**. The elimination theorem is proved in almost exactly the same way as above, provided η_f is legitimate in T. The reduced formula φ^{rd} is defined correspondingly. For a constant c ($n = 0$ in Definition II), let $\varphi^{rd} := \exists z(\varphi\, {}^{z}_{c} \wedge \delta\, {}^{z}_{y})$, where $\varphi\, {}^{z}_{c}$ denotes the result of replacing c in φ by z ($\notin \mathit{var}\varphi$). Now let $n > 0$. If f does not appear in φ, set $\varphi^{rd} = \varphi$. Otherwise, looking at the first occurrence of f in φ from the left, we certainly may write $\varphi = \varphi_0\, \frac{f\vec{t}}{y}$ for appropriate φ_0, $\vec{t}$, and $y \notin \mathit{var}\varphi$. Clearly, $\varphi \equiv_{T_f} \exists y(\varphi_0 \wedge y = f\vec{t}) \equiv_{T_f} \varphi_1$, with $\varphi_1 := \exists y(\varphi_0 \wedge \delta_f(\vec{t}, y))$. If f still occurs in φ_1 then repeat this procedure, which ends in, say, m steps in a formula φ_m that no longer contains f. Then put $\varphi^{rd} := \varphi_m$.

Frequently, operation symbols f are introduced in more or less strictly formalized theories by definitions of the form

$$(*) \quad f\vec{x} := t(\vec{x}),$$

where of course f does not occur in the term $t(\vec{x})$. This procedure is in fact subsumed by Definition II, because the former is nothing more than a definitorial extension of T with the explicit definition

$$\eta_f : \; y = f\vec{x} \leftrightarrow y = t(\vec{x}).$$

This definition is legitimate, since $\forall\vec{x}\,\exists!y\, y = t(\vec{x})$ is a tautology. It can readily be shown that η_f^g is logically equivalent to $\forall\vec{x}\, f\vec{x} = t(\vec{x})$. Hence, $(*)$ can indeed be regarded as a kind of an informative abbreviation of a legitimate explicit definition with the defining formula $y = t(\vec{x})$.

Remark 1. Instead of introducing new operation symbols, so-called *iota-terms* from [HB] could be used. For any formula $\varphi = \varphi(\vec{x}, y)$ in a given language, let $\iota y\varphi$ be a term in which y appears as a variable *bound by* ι. Whenever $T \vDash \forall\vec{x}\exists!y\varphi$, then T is extended by the axiom $\forall\vec{x}\forall y[y = \iota y\varphi(\vec{x}, y) \leftrightarrow \varphi(\vec{x}, y)]$, so that $\iota y\varphi(\vec{x}, y)$ so to speak stands for the function term $f\vec{x}$, which could have been introduced by an explicit definition. We mention that a definitorial language expansion is not a necessity. In principle, formulas of the expanded language can always be understood as abbreviations in the original language. This is in some presentations the actual procedure, though our imagination prefers additional notions over long sentences that would arise if we were to stick to a minimal set of basic notions.

Definitions I and II can be unified in a more general declaration. Let T, T' be theories in the languages $\mathcal{L}$, $\mathcal{L}'$, respectively. Then T' is called a *definitorial extension* (or *expansion*) of T whenever $T' = T + \Delta$ for some list Δ of explicit definitions of new symbols legitimate in T, given in terms of those of T (here *legitimate* refers to operation symbols and constants only). Δ need not be finite, but in most cases it is finite. A reduced formula $\varphi^{rd} \in \mathcal{L}$ is stepwise constructed as above, for every $\varphi \in \mathcal{L}'$. In this way the somewhat long-winded proof of the following theorem is reduced each time to the case of an extension by a single symbol:

Theorem 6.2 (General elimination theorem). *Let T' be a definitorial extension of T. Then $\alpha \in T' \Leftrightarrow \alpha^{rd} \in T$. In particular, $\alpha \in T' \Leftrightarrow \alpha \in T$ whenever $\alpha \in \mathcal{L}$, i.e., T' is a conservative extension of T.*

A relation or operation symbol s occurring in $T \subseteq \mathcal{L}$ is termed *explicitly definable in* T if T contains an explicit definition of s whose defining formula belongs to $\mathcal{L}_0$, the language of symbols of T without s. For example, in the theory T_G of groups the constant e is explicitly defined by $x = e \leftrightarrow x \circ x = x$; Example 2 page 83. Another example is presented

in Exercise 3. In such a case each model of $T_0 := T \cap \mathcal{L}_0$ can be expanded in only one way to a T-model. If this special condition is fulfilled then s is said to be *implicitly definable* in T. This could also be stated as follows: if T' is distinct from T only in that the symbol s is everywhere replaced by a new symbol s', then either $T \cup T' \vDash \forall\vec{x}(\mathsf{s}\vec{x} \leftrightarrow \mathsf{s}'\vec{x})$ or $T \cup T' \vDash \forall\vec{x}(\mathsf{s}\vec{x} = \mathsf{s}'\vec{x})$, depending on whether s, s' are relation or operation symbols. It is highly interesting that this kind of definability is already sufficient for the explicit definability of s in T. But we will go without the proof and only quote the following theorem.

Beth's definability theorem. *A relation or operation symbol implicitly definable in a theory T is also explicitly definable in T.*

Definitorial expansions of a language should be conscientiously distinguished from expansions of languages that arise from the introduction of so-called *Skolem functions*. These are useful for many purposes and are therefore briefly described.

Skolem normal forms. According to Theorem 4.2, every formula α can be converted into an equivalent PNF, $\alpha \equiv \mathsf{Q}_1 x_1 \cdots \mathsf{Q}_k x_k \alpha'$, where α' is open. Obviously then $\neg\alpha \equiv \overline{\mathsf{Q}}_1 x_1 \cdots \overline{\mathsf{Q}}_k x_k \neg\alpha'$, where $\overline{\forall} = \exists$ and $\overline{\exists} = \forall$. Because $\vDash \alpha$ if and only if $\neg\alpha$ is unsatisfiable, the decision problem for general validity can first of all be reduced to the satisfiability problem for formulas in PNF. Using Theorem 6.3 below, the latter—at the cost of introducing new operation symbols—is then completely reduced to the satisfiability problem for $\forall$-formulas.

Call formulas α and β *satisfiably equivalent* if both are satisfiable (not necessarily in the same model), or both are unsatisfiable. We construct for every formula, which w.l.o.g. is assumed to be given in prenex form $\alpha = \mathsf{Q}_1 x_1 \cdots \mathsf{Q}_k x_k \beta$, a satisfiably equivalent $\forall$-formula $\hat{\alpha}$ with additional operation symbols such that $\mathit{free}\,\hat{\alpha} = \mathit{free}\,\alpha$. The construction of $\hat{\alpha}$ will be completed after m steps, where m is the number of $\exists$-quantifiers among the $\mathsf{Q}_1, \ldots, \mathsf{Q}_k$. Take $\alpha = \alpha_0$ and α_i to be already constructed. If α_i is already an $\forall$-formula let $\hat{\alpha} = \alpha_i$. Otherwise α_i has the form $\forall x_1 \cdots \forall x_n \exists y \beta_i$ for some $n \geqslant 0$. With an n-ary operation symbol f (which is a constant in case $n{=}0$) not yet used let $\alpha_{i+1} = \forall\vec{x}\beta_i\,\frac{f\vec{x}}{y}$. Thus, after m steps an $\forall$-formula $\hat{\alpha}$ is obtained such that $\mathit{free}\,\hat{\alpha} = \mathit{free}\,\alpha$; this formula $\hat{\alpha}$ is called a *Skolem normal form* (SNF) of α.

Example 1. If α is the formula $\forall x \exists y\, x < y$ then $\hat{\alpha}$ is just $\forall x\, x < fx$.
For $\alpha = \exists x \forall y\, x \cdot y = y$ we have $\hat{\alpha} = \forall y\, c \cdot y = y$.
If $\alpha = \forall x \forall y \exists z (x < z \wedge y < z)$ then $\hat{\alpha} = \forall x \forall y (x < fxy \wedge y < fxy)$.

Theorem 6.3. *Let $\hat{\alpha}$ be a Skolem normal form for the formula α. Then*

(a) $\hat{\alpha} \vDash \alpha$, (b) *$\alpha$ is satisfiably equivalent to $\hat{\alpha}$.*

Proof. (a): It suffices to show that $\alpha_{i+1} \vDash \alpha_i$ for each of the described construction steps. $\beta_i \frac{f\vec{x}}{y} \vDash \exists y \beta_i$ implies $\alpha_{i+1} = \forall \vec{x} \beta_i \frac{f\vec{x}}{y} \vDash \forall \vec{x} \exists y \beta_i = \alpha_i$, by (c) and (d) in **2.5**. (b): If $\hat{\alpha}$ is satisfiable then by (a) so too is α. Conversely, suppose $\mathcal{A} \vDash \forall \vec{x} \exists y \beta_i(\vec{x}, y, \vec{z})\, [\vec{c}]$. For each $\vec{a} \in A^n$ we choose some $b \in A$ such that $\mathcal{A} \vDash \beta\, [\vec{a}, b, \vec{c}\,]$ (which is possible in view of the axiom of choice AC) and expand $\mathcal{A}$ to $\mathcal{A}'$ by setting $f^{\mathcal{A}'} \vec{a} = b$ for the new operation symbol. Then evidently $\mathcal{A}' \vDash \alpha_{i+1}\, [\vec{c}\,]$. Thus, we finally obtain a model for $\hat{\alpha}$ that expands the initial model. ❏

Now, for each α, a tautologically equivalent $\exists$-formula $\check{\alpha}$ is gained as well (that is, $\vDash \alpha \Leftrightarrow \vDash \check{\alpha}$). By the above theorem, we first produce for $\beta = \neg\alpha$ a satisfiably equivalent SNF $\hat{\beta}$ and put $\check{\alpha} := \neg\hat{\beta}$. Then indeed $\vDash \alpha \Leftrightarrow \vDash \check{\alpha}$, because

$$\vDash \alpha \;\Leftrightarrow\; \beta \text{ unsatisfiable} \;\Leftrightarrow\; \hat{\beta} \text{ unsatisfiable} \;\Leftrightarrow\; \vDash \check{\alpha}.$$

Example 2. For $\alpha := \exists x \forall y (ry \to rx)$ we have $\neg\alpha \equiv \beta := \forall x \exists y (ry \wedge \neg rx)$ and $\hat{\beta} = \forall x (rfx \wedge \neg rx)$. Thus, $\check{\alpha} = \neg\hat{\beta} \equiv \exists x (rfx \to rx)$. The last formula is a tautology. Indeed, if $r^{\mathcal{A}} \neq \emptyset$ then clearly $\mathcal{A} \vDash \exists x (rfx \to rx)$. But the same holds if $r^{\mathcal{A}} = \emptyset$, for then never $\mathcal{A} \vDash rfx$. Thus, $\check{\alpha}$ and hence also α is a tautology, which is not at all obvious after a first glance at α. This shows how useful Skolem normal forms can be for discovering tautologies.

Remark 2. There are many applications of Skolem normal forms, mainly in model theory and in logic programming. For instance, Exercise 5 permits one to reduce the satisfiability problem of an arbitrary first-order formula set to a set of $\forall$-formulas (at the cost of adjoining new function symbols). Moreover, a set X of $\forall$-formulas is satisfiably equivalent to a set X' of open formulas as will be shown in **4.1**, and this problem can be reduced completely to the satisfiability of a suitable set of propositional formulas, see also Remark 1 in **4.1**. The examples of applications of the propositional compactness theorem in **1.5** give a certain feeling for how to proceed in this way.

Exercises

1. Suppose that T_f results from T by adjoining an explicit definition η for f and let α^{rd} be constructed as explained in the text. Show that T_f is a conservative extension of T if and only if η is a legitimate explicit definition.

2. Let $\mathtt{S}: n \mapsto n+1$ denote the successor function in $\mathcal{N} = (\mathbb{N}, 0, \mathtt{S}, +, \cdot)$. Show that $Th\mathcal{N}$ is a definitorial extension of $Th(\mathbb{N}, \mathtt{S}, \cdot)$; in other words, 0 and $+$ are explicitly definable by $\mathtt{S}$ and $\cdot$ in $\mathcal{N}$.

3. Prove that $\eta : y = x^{-1} \leftrightarrow x \circ y = e$ is a legitimate explicit definition in T_G (it suffices to prove $T_G \vDash x \circ y = x \circ z \rightarrow y = z$). Show in addition that $T_G^{=} = T_G + \eta$. Thus, $T_G^{=}$ is a definitorial and hence a conservative extension of T_G. In this sense, the theories $T_G^{=}$ and T_G are equivalent formulations of the theory of groups.

4. As is well known, the natural $<$-relation of $\mathbb{N}$ is explicitly definable in $(\mathbb{N}, 0, +)$, for instance, by $x < y \leftrightarrow (\exists z \neq 0) z + x = y$. Prove that the $<$-relation of $\mathbb{Z}$ is not explicitly definable in $(\mathbb{Z}, 0, +)$.

5. Construct to each $\alpha \in X$ $(\subseteq \mathcal{L})$ an SNF $\hat{\alpha}$ such that X is satisfiably equivalent to $\hat{X} = \{\hat{\alpha} \mid \alpha \in X\}$ and $\hat{X} \vDash X$, called a *Skolemization* of X. Since we do not suppose that X is countable, the function symbols introduced in $\hat{X}$ must properly be indexed.

http://www.springer.com/978-1-4419-1220-6

A Concise Introduction to Mathematical Logic
Rautenberg, W.
2010, XXII, 320 p. 25 illus., Softcover
ISBN: 978-1-4419-1220-6