Theory of Automated Reasoning
An Introduction

Antti-Juhani Kaijanaho

Intended as compulsory reading for the Spring 2004 course on Automated Reasononing at Department of Mathematical Information Technology, University of Jyväskylä.

CHAPTER 2

A precis on logic

We assume that all readers already have a basic grasp of formal logic. The purpose of this chapter is to set concrete definitions and recall the main properties for the fundamental concepts used throughout this booklet. It is not meant to teach these concepts, and to a reader with no previous exposure to formal logic, it will undoubtedly be unapproachable. Some concepts (particularly the idea of many-sortedness) may be unfamiliar to most readers, and I hope that the exposition given here is sufficient to introduce those concepts. Since our goal is not to explore the foundations of mathematics, we will not bother with using finitary methods when they are inconvenient.

1. Preliminaries

Our metalanguage is mathematicians' informal English with the semi-formal mathematical notation customarily used in mathematics. We use set theory (in the style of ZFC) freely in our metalanguage. We write 2^X for the power set of a set X (the set of all of its subsets) and $\prod_{i=0}^{n} X_i$ for the Cartesian product $X_1 \times \cdots \times X_n$. We use a variant of Church's λ notation to write down anonymous functions: $\lambda x \in S : f(x)$ denotes $f(x)$ as the function of x, where the domain of the function is S. We regard a function as a set of pairs; we use higher-order functions (functions whose domain or range contains functions) freely. We denote the set of functions from S to T with $S \to T$; the notation $f : S \to T$ is a synonym for $f \in S \to T$. The domain of a function f is denoted by $\mathrm{dom}(f)$, and its range is denoted by $\mathrm{ran}(f)$. Sometimes we need functions whose domain is potentially a subset of a particular set; these functions are called partial functions, and the set of partial functions from S to T is denoted by

$S \nrightarrow T$. Note that $f \in S \to T$ is equivalent to $f \in S \nrightarrow T \wedge \mathrm{dom}(f) = S$ and that $\mathrm{dom}(\lambda x \in S : f(x)) = S$ always holds. When we want to denote the function or relation associated with a binary operator, we enclose the operator in parentheses.

2. Vocabularies and structures

We assume the existence of a countably infinite set $\mathcal{V}$.

METADEFINITION 2.1 (Vocabularies). Let F and R be countable sets such that $F \cap R = R \cap \mathcal{V} = F \cap \mathcal{V} = \emptyset$ holds. Let a be a function from $F \cup R$ to the natural numbers for which $a(r) \neq 0$ holds for all $r \in R$. Then the triple (F, R, a) is an *unsorted vocabulary*. Furthermore, let S be a non-empty set with the property $S \cap R = \emptyset$ and let s be a function from $(F \cup R) \times \mathbb{N}$ to $S \cup \{\bot\}$, where $\bot \notin S$ holds, and where for all $f \in F$ and $n \in \mathbb{N}$ it holds that $s(f, n) = \bot$ is equivalent to $n > a(f)$, and where for all $r \in R$ and $n \in \mathbb{N}$ it holds that $s(r, n) = \bot$ is equivalent to $n \geq a(f)$. Then the quintuple (F, R, a, S, s) is a *many-sorted vocabulary*.

EXAMPLE 2.2. Let F consist of $(+)$, 0 and succ, and let $a(+) = 2$, and $a(0) = 0$, and $a(\mathrm{succ}) = 1$ hold. Let R contain only $(=)$ with $a(=) = 2$. Then (F, R, a) is the vocabulary of Presburger arithmetic.

EXAMPLE 2.3. Let (F', R, a') be the vocabulary of Prezburger arithmetic. Let F be $F' \cup \{(\times)\}$, and let a be $a' \cup \{(\times, 2)\}$. Then (F, R, a) is the vocabulary of first-order arithmetic *PA* (for *Peano arithmetic*).

EXAMPLE 2.4. Let (F, R', a') be the vocabulary of first-order arithmetic, and let R be $R' \cup \{(\in)\}$ and let a be $a' \cup \{(\in, 2)\}$. Now, let S be $\{N, C\}$ and let the following hold: $s((+), 0) = N$, $s((+), 1) = N$, $s((+), 2) = N$, $s((\times), 0) = N$, $s((\times), 1) = N$, $s((\times), 2) = N$, $s(0, 0) = N$, $s((\mathrm{succ}), 0) = N$, $s((\mathrm{succ}), 1) = N$), $s((\in), 0) = N$, $s((\in), 1) = C$. Then (F, R, a, S, s) is the vocabulary of second-order arithmetic Z_2.

We call the elements of F *function symbols* and the elements of R *predicate symbols*. The value of a for a function or predicate symbol is called its *arity*. Function symbols with arity 0 are called *constants*. Function and predicate symbols with arity n are called n-ary symbols; we use *unary* for 1-ary, *binary* for 2-ary and *tertiary* for 3-ary. We call elements of S *sorts*. The value of s for a function or predicate symbol and a natural number strictly smaller than the value of a for that symbol is called the $(n+1)$*th place sort* of the symbol. The value of s for a function symbol and the value of a for that symbol is called the *result sort* of the symbol.

EXAMPLE 2.5. The vocabulary of set theory is unsorted with no function symbols or constant symbols, and just one predicate symbol, $(\in)$.

Note that there is a bijective function mapping any unsorted vocabulary (F, R, a) to a sorted vocabulary (F, R, a, S, s) with one sort (that is, S contains one element; in such a case, s is uniquely determined). We will, from now on, identify the one-sort case with the unsorted case. All definitions phrased for the many-sorted case is carried to the unsorted case via this identification. Since this identification essentially makes the unsorted case a special case of the many-sorted case, there is usually no need to explicate whether unsorted or many-sorted vocabularies are meant.

METADEFINITION 2.6 (Structures). Let $V = (F, R, a, S, s)$ be a vocabulary. Let U be a nonempty set, and let $p : S \to 2^U$ be a partitioning (i.e. for all $\tau_1, \tau_2 \in S$ it is the case that $p(\tau_1) \cap p(\tau_2) \neq \emptyset$ holds if and only if $\tau_1 = \tau_2$ holds). Furthermore, let m be a function from $F \cup R$ with the following properties:

(1) For all $f \in F$ which are not constants, it holds that $m(f)$ is a function
$$\left(\prod_{i=0}^{a(f)-1} s(f, i) \right) \to s(f, a(f)).$$
(2) For all $r \in R$, it holds that $m(r)$ is a subset of
$$\prod_{i=0}^{a(f)-1} s(f, i).$$
(3) For all constants $c \in F$, the assertion $m(c) \in s(c, 0)$ holds.

Then the triple (U, m, p) is a *many-sorted (first-order) structure* over the vocabulary V. The pair (U, m) is an *unsorted (first-order) structure* over the unsorted vocabulary corresponding to V in the one-sorted special case.

The set U of a structure is called the *universe* of that structure. The value of m for a symbol is called its *interpretation* in the structure being considered. The value of p for each sort is called the *domain* of the sort.

EXAMPLE 2.7. Let V be the vocabulary of *PA* (Ex. 2.3). Recall that $\mathbb{N}$ stands for the set of natural numbers (including zero), and also recall the customary meanings of addition $(+)$ and multiplication $(\cdot)$ of natural numbers. Now $(\mathbb{N}, m)$ is the *standard model of PA*, if m is defined as follows:

(1) $m(+) = \lambda(n, m) \in \mathbb{N} \times \mathbb{N} : n + m$
(2) $m(\times) = \lambda(n, m) \in \mathbb{N} \times \mathbb{N} : n \cdot m$
(3) $m(0) = 0$
(4) $m(\text{succ}) = \lambda n \in \mathbb{N} : n + 1$
(5) $m(=) = \{ (n, m) \in N \times \mathbb{N} \mid n = m \}$

Like in the case of vocabularies, there is a bijection that maps unsorted structures (U, m) to sorted structures (U, m, p), where $p(\tau) = U$ holds for the sole sort τ. We will identify unsorted structures with many-sorted structures with a single sort through this bijection, and we will liberally drop the qualifiers "many-sorted" and "unsorted".

METADEFINITION 2.8 (Equality vocabulary and structure). A vocabulary is an *equality vocabulary* if in its set of predicate symbols there is a distinguished *equality* symbol "$=_\tau$" for each sort τ with $a(=_\tau) = 2$, and $s((=_\tau), 0) = s((=_\tau), 1) = \tau$. A structure is an *equality structure* if its vocabulary is an equality vocabulary and the interpretation of each equality symbol is an equivalence relation in the domain of its sort. The unsorted case is defined through the identification of unsorted and one-sorted vocabularies and structures.

EXAMPLE 2.9. The vocabulary of *group theory* is unsorted and consists of one constant symbol, one unary function symbol, one binary function symbol and an equality symbol. The choice of actual symbols is immaterial, but customary choices are e or 1 for the constant symbol, $(^{-1})$ (as in, x^{-1}) for the unary function symbol and $(\circ)$ for the binary function symbol, as well as 0 for the constant symbol, $(-)$ for the unary function symbol and $(+)$ for the binary symbol.

Examples of equality structures over the vocabulary of group theory are the following:

The additive group of real numbers: Let the universe be the set of real numbers, let the constant symbol map to 0, let the unary function symbol map to $\lambda x \in \mathbb{R} : -x$, and let the binary function symbol map to $\lambda(x, y) \in \mathbb{R}^2 : x + y$.

The multiplicative group of real numbers: Let the universe be the set of real numbers except 0, let the constant symbol map to 1, let the unary function symbol map to $\lambda x \in \mathbb{R} : \frac{1}{x}$, and let the binary function symbol map to $\lambda(x, y) \in \mathbb{R}^2 : xy$.

The cyclic group $\mathbb{Z}_2$: Let the universe be the set $\mathbb{Z}_2 = \{0, 1\}$, let the constant symbol map to 0, let the unary function symbol map to a function that maps 0 to 0 and 1 to 1 (arithmetic negation modulo 2), and let the binary function symbol map to a function that maps $(0, 0)$ and $(1, 1)$ to 0, and $(0, 1)$ and $(1, 0)$ to 1 (addition modulo 2).

EXAMPLE 2.10. A vocabulary of *Hilbert geometry* consists of two sorts, one for *lines* and one for *points*, a binary predicate symbol for *incidence* with the first place sort being the points sort and the second place sort being the lines sort, a tertiary predicate symbol for *betweenness* (all places have the points sort), a 4-ary predicate symbol for *segment congruence* (all places have the points sort), a 6-ary predicate symbol for *angle congruence* (all places have the points sort), and

equality symbols for both sorts; there are no function symbols. The choice of actual symbols is immaterial.

One possible structure over an Hilbert geometry vocabulary is the system of Euclidean plane vectors: the range of the points sort is $\mathbb{R}^2$ (a position vector) and the range of the lines sort is $\mathbb{R}^2 \times \mathbb{R}^2$ (a direction vector and a translation vector); the incidence predicate symbol is mapped to the set

$$\{\, (\vec{v}, (\vec{u}, \vec{w})) \in \mathbb{R}^2 \times (\mathbb{R}^2 \times \mathbb{R}^2) \mid \exists k \in \mathbb{R} : \vec{v} = k\vec{u} + \vec{w} \,\},$$

the betweenness predicate symbol is mapped to the set

$$\{\, (\vec{v}, \vec{u}, \vec{w}) \in (\mathbb{R}^2)^3 \mid \exists k \in \mathbb{R} : \vec{u} - \vec{v} = k\vec{u} - k\vec{w} \wedge k < 0 \,\},$$

the segment congruence predicate symbol is mapped to the set

$$\{\, (\vec{s}, \vec{t}, \vec{u}, \vec{v}) \in (\mathbb{R}^2)^4 \mid \left\|\vec{s} - \vec{t}\right\| = \|\vec{u} - \vec{v}\| \,\},$$

the angle congruence predicate symbol is mapped to the set

$$\left\{ (\vec{a_1}, \vec{a_2}, \vec{a_3}, \vec{b_1}, \vec{b_2}, \vec{b_3}) \in (\mathbb{R}^2)^6 \mid \vec{a_1} \neq \vec{a_2} \neq \vec{a_3} \wedge \vec{b_1} \neq \vec{b_2} \neq \vec{b_3} \wedge \frac{(\vec{a_1} - \vec{a_2}) \cdot (\vec{a_3} - \vec{a_2})}{\|\vec{a_1} - \vec{a_2}\| \, \|\vec{a_3} - \vec{a_2}\|} = \frac{(\vec{b_1} - \vec{b_2}) \cdot (\vec{b_3} - \vec{b_2})}{\|\vec{b_1} - \vec{b_2}\| \|\vec{b_3} - \vec{b_2}\|} \right\},$$

the point equality predicate symbol is mapped to the standard vector equality relation, and the line equality predicate symbol is mapped to the set

$$\left\{\, ((\vec{u_1}, \vec{u_2}), (\vec{v_1}, \vec{v_2})) \in ((\mathbb{R}^2)^2)^2 \mid \exists k \in \mathbb{R} : \vec{u_1} - \vec{u_2} = k\vec{v_1} - k\vec{v_2} \,\right\}.$$

3. First-order languages

METADEFINITION 2.11 (First-order languages). Let $V = (F, R, a, S, s)$ be a many-sorted vocabulary. We define

- a set of *proto-terms* over the vocabulary V, denoted by $\dot{\mathcal{T}}_V$,
- a set of *proto-formulae* over the vocabulary V, denoted by $\dot{\mathcal{F}}_V$,
- a function $\dot{\mathrm{fv}} : (\dot{\mathcal{F}}_V \cup \dot{\mathcal{T}}_V) \to 2^{\mathcal{V}}$ (a proto-free-variable-mapping),
- a function $\dot{\mathrm{Sb}} : (\mathcal{V} \nrightarrow \dot{\mathcal{T}}_V) \to ((\dot{\mathcal{F}}_V \to \dot{\mathcal{F}}_V) \cup (\dot{\mathcal{T}}_V \to \dot{\mathcal{T}}_V))$ (a proto-substitutor),
- a function $\mathrm{tsort} : (\mathcal{V} \nrightarrow S) \times \dot{\mathcal{T}}_V \to (S \cup \{\bot\})$ (a term-sorter), and
- a function $\mathrm{fsort} : (\mathcal{V} \nrightarrow S) \times \dot{\mathcal{F}}_V \to (\{0, 1\})$ (a formula-sorter)

by mutual recursion as follows, letting σ be any partial function $\mathcal{V} \nrightarrow \dot{\mathcal{T}}_V$:

(1) For each $x \in \mathcal{V}$, we let $(0, x) \in \dot{\mathcal{T}}_V$ hold, $\mathrm{tsort}(E, (0, x))$ be

$$\begin{cases} E(x), & \text{if } x \in \mathrm{dom}(E), \\ \bot, & \text{if } x \notin \mathrm{dom}(E), \end{cases}$$

and $\dot{\mathrm{fv}}(0,x)$ be $\{x\}$, and $\dot{\mathrm{Sb}}(\sigma)(0,x)$ be

$$\begin{cases}\sigma(x), & \text{if } x \in \mathrm{dom}(\sigma),\\ (0,x) & \text{if } x \notin \mathrm{dom}(\sigma),\end{cases}$$

(2) For every $f \in F$ and $t_1,\dots,t_{a(f)} \in \dot{\mathcal{T}}_V$, we let $(1,f,t_1,\dots,t_{a(f)}) \in \dot{\mathcal{T}}_V$ hold, $\mathrm{tsort}(E,(1,f,t_1,\dots,t_{a(f)}))$ be

$$\begin{cases}s(f,a(f)), & \text{if } \mathrm{tsort}(E,t_i) \text{ is } s(f,i-1) \text{ for every } i,\\ \bot, & \text{otherwise,}\end{cases}$$

and $\dot{\mathrm{fv}}(1,f,t_1,\dots,t_{a(f)})$ be $\dot{\mathrm{fv}}(t_1)\cup\cdots\cup\dot{\mathrm{fv}}(t_{a(f)})$, and $\dot{\mathrm{Sb}}(\sigma)(1,f,t_1,\dots,t_{a(f)})$ be $(1,f,\dot{\mathrm{Sb}}(\sigma)(t_1),\dots,\dot{\mathrm{Sb}}(\sigma)(t_{a(f)}))$.

(3) For every $p \in \dot{\mathcal{F}}_V$, $\tau \in S$ and $x \in \mathcal{V}$, we let $(2,x,\tau,p) \in \dot{\mathcal{T}}_V$ hold, $\mathrm{tsort}(E,(2,x,\tau,p))$ be

$$\begin{cases}\tau, & \text{if } \mathrm{fsort}(E',p)=1 \text{ holds,}\\ \bot, & \text{otherwise,}\end{cases}$$

where E' is $\{\,(x',\tau') \in \mathcal{V} \nrightarrow S \mid (x'=x \to \tau'=\tau) \wedge (x' \in \mathrm{dom}(E)\setminus \{x\} \to \tau' = E(x)) \wedge x' \in \mathrm{dom}(E)\cup\{x\}\,\}$.

(4) For every $r \in R$ and $t_1,\dots,t_{a(r)} \in \dot{\mathcal{T}}_V$, we let $(3,r,t_1,\dots,t_{a(r)}) \in \dot{\mathcal{F}}_V$ hold, $\mathrm{tsort}(E,(3,r,t_1,\dots,t_{a(r)}))$ be

$$\begin{cases}1, & \text{if } \mathrm{tsort}(E,t_i) \text{ is } s(f,i-1) \text{ for every } i,\\ 0, & \text{otherwise,}\end{cases}$$

and $\dot{\mathrm{fv}}(3,r,t_1,\dots,t_{a(r)})$ be $\dot{\mathrm{fv}}(t_1)\cup\cdots\cup\dot{\mathrm{fv}}(t_{a(r)})$, and $\dot{\mathrm{Sb}}(\sigma)(3,r,t_1,\dots,t_{a(r)})$ be $(3,r,\dot{\mathrm{Sb}}(\sigma)(t_1),\dots,\dot{\mathrm{Sb}}(\sigma)(t_{a(f)}))$.

(5) For every $p \in \dot{\mathcal{F}}_V$, we let $(4,p) \in \dot{\mathcal{F}}_V$ hold, $\mathrm{fsort}(E,(4,p))$ be $\mathrm{fsort}(E,p)$, and $\dot{\mathrm{fv}}(4,p)$ be $\dot{\mathrm{fv}}(p)$ and and $\dot{\mathrm{Sb}}(\sigma)(4,p)$ be $\dot{\mathrm{Sb}}(\sigma)(p)$.

(6) For every $p,q \in \dot{\mathcal{F}}_V$ and for every $n \in \{5,\dots,12\}$, we let $(n,p,q) \in \dot{\mathcal{F}}_V$ hold, $\mathrm{fsort}(E,(n,p,q))$ be equal to the product of $\mathrm{fsort}(E,p)$ and $\mathrm{fsort}(E,q)$, and $\dot{\mathrm{fv}}(n,p,q)$ be $\dot{\mathrm{fv}}(p)\cup\dot{\mathrm{fv}}(q)$, and $\dot{\mathrm{Sb}}(\sigma)(n,p,q)$ be $(n,\dot{\mathrm{Sb}}(\sigma)(p),\dot{\mathrm{Sb}}(\sigma)(q))$.

(7) For every $x \in \mathcal{V}$, $\tau \in S$, $p \in \dot{\mathcal{F}}_V$ and $n \in \{13,14\}$, we let $(n,x,\tau,p) \in \dot{\mathcal{F}}_V$ hold, and $\mathrm{fsort}(E,(n,x,\tau,p))$ be equal to

$$\begin{cases}\tau, & \text{if } \mathrm{fsort}(E',p)=1 \text{ holds,}\\ \bot, & \text{otherwise,}\end{cases}$$

where E' is $\{\,(x',\tau') \in \mathcal{V} \nrightarrow S \mid (x'=x \to \tau'=\tau) \wedge (x' \in \mathrm{dom}(E)\setminus \{x\} \to \tau' = E(x)) \wedge x' \in \mathrm{dom}(E)\cup\{x\}\,\}$

(8) For every $x \in \mathcal{V}$, $\tau \in S$, $p \in \dot{\mathcal{F}}_V$ and $n \in \{2,13,14\}$, let $\dot{\mathrm{fv}}(n,x,\tau,p)$ be $\dot{\mathrm{fv}}(p)\setminus\{x\}$ and $\dot{\mathrm{Sb}}(\sigma)(n,x,\tau,p)$ be $(n,y,\tau,\dot{\mathrm{Sb}}(\sigma')(p))$, where $y \in \mathcal{V}$ is chosen so that $y \notin \dot{\mathrm{fv}}(t)$ holds for all $t \in \mathrm{ran}(\sigma)$,

and where σ' is $\{\,(z,t) \in \mathcal{V} \times \dot{\mathcal{T}}_V \mid z \in \mathrm{dom}(\sigma) \setminus \{x\} \wedge t = \sigma(z)\,\} \cup \{(x,(0,y))\}$.

Further, we define the sets of *well-formed terms* $\mathcal{T}_V$ and *well-formed formulae* $\mathcal{F}_V$ as follows:

$$\mathcal{T}_V = \{\,(E,t) \in (\mathcal{V} \twoheadrightarrow S) \times \dot{\mathcal{T}}_V \mid \mathrm{tsort}(E,t) \neq \bot \wedge \mathrm{dom}(E) = \dot{\mathrm{fv}}(t)\,\}$$

$$\mathcal{F}_V = \{\,(E,p) \in (\mathcal{V} \twoheadrightarrow S) \times \dot{\mathcal{F}}_V \mid \mathrm{fsort}(E,p) = 1 \wedge \mathrm{dom}(E) = \dot{\mathrm{fv}}(p)\,\}$$

We also define a new function $\mathrm{fv} : (\mathcal{T}_V \cup \mathcal{F}_V) \to (\mathcal{V} \twoheadrightarrow S)$ as $\mathrm{fv}(E,f) = E$ for all $(E,f) \in \mathcal{F}_V \cup \mathcal{T}_V$, and another new function $\mathrm{Sb} : (\mathcal{V} \twoheadrightarrow \mathcal{T}_V) \to ((\mathcal{F}_V \twoheadrightarrow \mathcal{F}_V) \cup (\mathcal{T}_V \twoheadrightarrow \mathcal{T}_V))$ as $\mathrm{Sb}(\sigma) = \{\,((E,t),(E',t')) \in \mathcal{T}_V^2 \cup \mathcal{F}_V^2 \mid t' = \dot{\mathrm{Sb}}(\sigma)(t) \wedge E' = \{\,(z,\tau) \in \mathcal{V} \twoheadrightarrow S \mid z \in \mathrm{dom}(E) \setminus \mathrm{dom}(\sigma) \wedge \tau = E(z)\,\} \cup \{\,(z,\tau) \in \mathcal{V} \twoheadrightarrow S \mid \exists z' \in \mathrm{dom}(\sigma) : z \in \dot{\mathrm{fv}}(\sigma(z')) \wedge \tau = \mathrm{tsort}(\sigma(z'))\,\} \wedge \forall (x,u) \in \sigma : x \in \mathrm{dom}(\mathrm{fv}(t)) \to \mathrm{fv}(t)(x) = \mathrm{tsort}(\sigma)(u)\,\}$.

Now, the quadruple $\mathcal{L}_V = (\mathcal{T}_V, \mathcal{F}_V, \mathrm{fv}, \mathrm{Sb})$ is the *many-sorted first-order language* of the alphabet V.

A definition of an unsorted first-order language is omitted here, since it would be tediously repetitious: it can be obtained from the above definition by removing all references to sorts. As before, the unsorted case will be identified with the many-sorted case with a single sort. We will, for that reason, drop the qualifier "many-sorted" from now on.

Note that the notion of a well-formed term and a well-formed formula presumes that every free variable is given a sort; this is imitated by having each well-formed term and formula contain a mapping giving a sort for each free variable along with the actual term or formula. The function fv just extracts that mapping from the well-formed term or formula. It is worth noting that $\mathrm{dom}(\mathrm{fv}(f))$ denotes the set of free variables in a well-formed term or formula f. The function Sb denotes the operation of applying a substitution to a formula or term; it is defined in all cases where the substitution does not create an ill-formed term or formula. We will informally identify a well-formed formula and a well-formed term with their second elements, proto-formula and proto-term, respectively, taking the sort-giving mapping as evident from the context.

The definition given above gives the abstract syntax of terms and formulae. Essentially, each term and formula is a tree with one or more label (the numbers and in some cases the predicate or function symbol) and zero or more subtrees. Such a definition is less work than a traditional string-based definition, and it is closer to the reality of how terms and formulae are represented on a computer. To make it easier to read and write terms and formulae of a first-order language, we set up the informal convention given in Table 1 (note that there are several different notational conventions in the literature). We will use parentheses liberally to indicate the proper

When we write…	…we mean…	…and we call it…
x	$(0, x)$	variable term
$f(t_1, \ldots, t_n)$	$(1, f, t_1, \ldots, t_n)$	function application
tfu	$(1, f, t, u)$	infix function application
f	$(1, f)$	constant term
$(\epsilon x : \tau)p$	$(2, x, \tau, p)$	committed choice
$(\epsilon x)p$	$(2, x, \tau, p)$	unsorted committed choice
$r(t_1, \ldots, t_n)$	$(3, r, t_1, \ldots, t_n)$	atomic formula
tru	$(3, r, t, u)$	infix atomic formula
$\neg p$	$(4, p)$	negation
$p \wedge q$	$(5, p, q)$	conjunction
$p \vee q$	$(6, p, q)$	disjunction
$p \rightarrow q$	$(7, p, q)$	conditional
$p \leftarrow q$	$(8, p, q)$	reverse conditional
$p \leftrightarrow q$	$(9, p, q)$	biconditional
$p \mid q$	$(10, p, q)$	Sheffer stroke (NAND)
$p \downarrow q$	$(11, p, q)$	Peirce's arrow (NOR)
$p \veebar q$	$(12, p, q)$	exclusive disjunction
$(\forall x : \tau)p$	$(13, x, \tau, p)$	universal quantification
$(\exists x : \tau)p$	$(14, x, \tau, p)$	existential quantification
$(\forall x)p$	$(13, x, \tau, p)$	unsorted universal quantification
$(\exists x)p$	$(14, x, \tau, p)$	unsorted existential quantification
$p[t/x]$	$\mathrm{Sb}(\{(x, t)\})(p)$	simple term substitution

Here, each line holds for all $x \in \mathcal{V}$, and $f \in F$, and $t, t_i, u \in \mathcal{T}_V$, and $p, q \in \mathcal{F}_V$, and $r \in R$, and $\tau \in S$. Unsorted committed choice, unsorted universal quantification and unsorted existential quantification may only be used in languages with a single sort. In those cases, the implicit τ is uniquely determined by the vocabulary.

TABLE 1. An informal convention for writing terms and formulae of a first-order language

abstract-syntax structure of each term and formula. We will generally assume, unless otherwise indicated, that x, y and z and their primed, uppercased or subscripted variants, when used in a formula or term, stand for some unspecified but *a priori* fixed variables distinct from each other. All other letters will be introduced as they are used.

The committed choice term (also known as ϵ-term) may be unfamiliar to many readers. The intuitive denotation of a term $(\epsilon x : \tau)p$ is "an x of sort τ for which p holds". It chooses some value from the extension of p (the subset of the range of τ for which p holds); this choice is demonic in the sense that we have no information about which element of the extension is chosen, only that it is an element; the choice is committed in the sense that two terms $(\epsilon x : \tau)p$ and

$(\epsilon y : \tau)q$ are equal at least in all cases where p and $q[x/y]$ are logically equivalent. A rigorous definition is given below.

METADEFINITION 2.12 (Component relation). Let $\mathcal{L}_V = (\mathcal{T}_V, \mathcal{F}_V, \mathrm{fv}, \mathrm{Sb})$ be a first-order language over a vocabulary $V = (F, R, a, S, s)$. We define a function $f : (\mathcal{T}_V \cup \mathcal{F}_V) \to 2^{\mathcal{T}_V \cup \mathcal{F}_V}$, case by case recursively as follows:

(1) $f(E, (1, g, t_1, \ldots, t_{a(f)})) = \{(E, t_1), \ldots, (E, t_{a(f)})\}$ holds for all $g \in F$, and $t_i \in \mathcal{T}_V$, and $i \in \{1, \ldots, a(f)\}$, and $E \in \mathcal{V} \nrightarrow S$ such that $(E, (1, g, t_1, \ldots, t_{a(f)}))$ is a well-formed term;
(2) $f(E, (n, x, \tau, p)) = \{(E' \cup \{(x, \tau)\}, p)\}$ holds for all $p \in \mathcal{F}_V$, and $x \in \mathcal{V}$, and $\tau \in S$, and $n \in \{2, 13, 14\}$, and $E \in \mathcal{V} \nrightarrow S$ such that $(E, (n, x, \tau, p))$ is a well-formed term or formula, and $E' = \{(z, \tau') \mid z \in \mathrm{dom}(E) \setminus \{x\} \wedge \tau' = E(z)\}$;
(3) $f(E, (3, r, t_1, \ldots, t_{a(r)})) = \{(E, t_1), \ldots, (E, t_n)\}$ holds for all $r \in R$, and $t_i \in \mathcal{T}_V$ and $i \in \{1, \ldots, a(r)\}$, and $E \in \mathcal{V} \nrightarrow S$ such that $(E, (3, r, t_1, \ldots, t_{a(f)}))$ is a well-formed formula;
(4) $f(E, (4, p)) = \{(E, p)\}$ holds for all $p \in \mathcal{F}_V$, and $E \in \mathcal{V} \nrightarrow S$ such that $(E, (4, p))$ is a well-formed formula; and
(5) $f(E, (n, p, q)) = \{p, q\}$ holds for all $p, q \in \mathcal{F}_V$, and $n \in \{5, \ldots, 12\}$, and $E \in \mathcal{V} \nrightarrow S$ such that $(E, (n, p, q))$ is a well-formed formula.

We further define a binary relation $(\prec) \subseteq (\mathcal{T}_V \cup \mathcal{F}_V) \times (\mathcal{T}_V \cup \mathcal{F}_V)$, the *component relation*, so that, for all $g, g' \in (\mathcal{F}_V \cup \mathcal{T}_V)$, it is the case that $g \prec g'$ holds if and only if there are $n \in \mathbb{N}$ and $g_1, \ldots, g_n \in (\mathcal{F}_V \cup \mathcal{T}_V)$ such that $g_1 = g$ and $f_n = g'$ hold and for all $i = 1, \ldots, n-1$, the assertion $g_i \in f(g_{i+1})$ holds.

METADEFINITION 2.13 (Restricted formulae). Let $\mathcal{L}_V = (\mathcal{T}_V, \mathcal{F}_V, \mathrm{fv}, \mathrm{Sb})$ be a first-order language over a vocabulary $V = (F, R, a, S, s)$, and let $t \in \mathcal{T}_V$ and $p \in \mathcal{F}_V$ hold. Then,

(1) t and p are a *ground term* and a *ground formula*, respectively, if there is no $x \in \mathcal{V}$ such that $x \prec t$ or $x \prec p$, respectively, holds;
(2) t and p are *ϵ-free*, if there are no $x \in \mathcal{V}$, $\tau \in S$ and $q \in \mathcal{F}_V$ such that $(\epsilon x : \tau)q \prec p$ and $(\epsilon x : \tau)q \prec t$, hold, respectively;
(3) t is a *(complex) constant* and p is a *sentence* if $\mathrm{fv}(t) = \emptyset$ and $\mathrm{fv}(p) = \emptyset$ hold, respectively.

We further define a set G_V as the set of all ground formulae and ground terms over the vocabulary V, and $\mathcal{S}_V$ as the set of all sentences over the vocabulary V.

4. Semantics

METADEFINITION 2.14 (Semantics of first-order languages). Let $\mathcal{L}_V = (\mathcal{T}_V, \mathcal{F}_V, \mathrm{fv}, \mathrm{Sb})$ be a first-order language over a vocabulary $V =$

(F, R, a, S, s) and let $\mathrm{St} = (U, m, p)$ be a structure over the vocabulary V. Further, let ch be a function from 2^U to U for which it holds that $\mathrm{ch}(X) \in X$ for every $X \subseteq U, X \neq \emptyset$. Then the pair $M_{\mathrm{ch}} = (\mathrm{St}, \mathrm{ch})$ is a *choice model* of the language $\mathcal{L}_V$.

Let $\mathcal{U}$ be $\mathcal{V} \to U$. We define, for each choice model M_{ch},

- a function $[\![\cdot]\!]_{M_{\mathrm{ch}}} : \mathcal{T}_V \to (\mathcal{V} \nrightarrow U) \nrightarrow U$ (the *denotation function*), and
- a function $[\![\cdot]\!]_{M_{\mathrm{ch}}} : \mathcal{F}_V \to \mathcal{V} \to U$ (the *extension function*), and

recursively as follows:

(1) For all $f \in \mathcal{V} \nrightarrow U$ and all $x \in \mathcal{V}$, the assertion $[\![x]\!]_{M_{\mathrm{ch}}}(f) = f(x)$ holds if and only if $x \in \mathrm{dom}(f)$ holds.

(2) For $f \in \mathcal{V} \nrightarrow U$ and all $g(t_1, \ldots, t_{a(g)}) \in \mathcal{T}_V$, the assertion

$$[\![g(t_1, \ldots, t_n)]\!]_{M_{\mathrm{ch}}}(f) = m(g)([\![t_1]\!]_{M_{\mathrm{ch}}}(f), \ldots, [\![t_{a(g)}]\!]_{M_{\mathrm{ch}}})$$

holds if and only if $[\![t_i]\!]_{M_{\mathrm{ch}}}(f)$ is defined for all i .

(3) For all $f \in \mathcal{V} \nrightarrow U$, and all $(\epsilon x : \tau)p \in \mathcal{T}_V$, the assertion $[\![(\epsilon x : \tau)p]\!]_{M_{\mathrm{ch}}}(f) = \mathrm{ch}(e)$ holds, where e is

$$\{\, y \in U \mid [\![p]\!]_{M_{\mathrm{ch}}} = \emptyset \vee \exists g \in [\![p]\!]_{M_{\mathrm{ch}}} : y = g(x) \wedge \forall z \in \mathrm{dom}(f) \setminus \{x\} : f(z) = g(z) \,\}$$

(4) For all $p = r(t_1, \ldots, t_{a(r)}) \in \mathcal{F}_V$, we define $[\![p]\!]_{M_{\mathrm{ch}}}$ as

$$\{\, g \in \mathcal{U} \mid \exists f \in \mathrm{dom}(\mathrm{fv}(p)) \to \mathrm{ran}(\mathrm{fv}(p)) : m(r)\left([\![t_1]\!]_{M_{\mathrm{ch}}}(f), \ldots, [\![t_{a(r)}]\!]_{M_{\mathrm{ch}}}(f)\right) \wedge \forall x \in \mathrm{dom}(f) : f(x) \in \mathrm{fv}(p)(x) \wedge g(x) = f(x) \,\}$$

(5) For all $p = \neg q \in \mathcal{F}_V$, we define $[\![p]\!]_{M_{\mathrm{ch}}}$ as $\mathcal{U} \setminus [\![q]\!]_{M_{\mathrm{ch}}}$.

(6) For all $p = q \wedge r \in \mathcal{F}_V$, we define $[\![p]\!]_{M_{\mathrm{ch}}}$ as $[\![q]\!]_{M_{\mathrm{ch}}} \cap [\![r]\!]_{M_{\mathrm{ch}}}$.

(7) For all $p = q \vee r \in \mathcal{F}_V$, we define $[\![p]\!]_{M_{\mathrm{ch}}}$ as $[\![q]\!]_{M_{\mathrm{ch}}} \cup [\![r]\!]_{M_{\mathrm{ch}}}$.

(8) For all $p = q \to r \in \mathcal{F}_V$, we define $[\![p]\!]_{M_{\mathrm{ch}}}$ as $(\mathcal{U} \setminus [\![q]\!]_{M_{\mathrm{ch}}}) \cup [\![r]\!]_{M_{\mathrm{ch}}}$.

(9) For all $p = q \leftarrow r \in \mathcal{F}_V$, we define $[\![p]\!]_{M_{\mathrm{ch}}}$ as $[\![q]\!]_{M_{\mathrm{ch}}} \cup (\mathcal{U} \setminus [\![r]\!]_{M_{\mathrm{ch}}})$.

(10) For all $p = q \leftrightarrow r \in \mathcal{F}_V$, we define $[\![p]\!]_{M_{\mathrm{ch}}}$ as $(\mathcal{U} \setminus ([\![p]\!]_{M_{\mathrm{ch}}} \cup [\![q]\!]_{M_{\mathrm{ch}}})) \cup ([\![p]\!]_{M_{\mathrm{ch}}} \cap [\![q]\!]_{M_{\mathrm{ch}}})$.

(11) For all $p = q \mid r \in \mathcal{F}_V$, we define $[\![p]\!]_{M_{\mathrm{ch}}}$ as $\mathcal{U} \setminus ([\![q]\!]_{M_{\mathrm{ch}}} \cap [\![r]\!]_{M_{\mathrm{ch}}})$.

(12) For all $p = q \downarrow r \in \mathcal{F}_V$, we define $[\![p]\!]_{M_{\mathrm{ch}}}$ as $\mathcal{U} \setminus ([\![q]\!]_{M_{\mathrm{ch}}} \cup [\![r]\!]_{M_{\mathrm{ch}}})$.

(13) For all $p = q \veebar r \in \mathcal{F}_V$, we define $[\![p]\!]_{M_{\mathrm{ch}}}$ as $([\![q]\!]_{M_{\mathrm{ch}}} \cup [\![r]\!]_{M_{\mathrm{ch}}}) \cap (\mathcal{U} \setminus ([\![q]\!]_{M_{\mathrm{ch}}} \cap [\![r]\!]_{M_{\mathrm{ch}}}))$.

(14) For all $p = (\forall x : \tau)q \in \mathcal{F}_V$, we define $[\![p]\!]_{M_{\mathrm{ch}}}$ as

$$\{\, f \in \mathcal{U} \mid \forall g \in \mathcal{U} : (\forall y \in \mathcal{V} \setminus \{x\} : f(y) = g(y)) \to g \in [\![p]\!]_{M_{\mathrm{ch}}} \,\}$$

(15) For all $p = (\exists x : \tau)q \in \mathcal{F}_V$, we define $[\![p]\!]_{M_{\text{ch}}}$ as

$$\{ f \in \mathcal{U} \mid \exists g \in \mathcal{U} : \big(\forall y \in \mathcal{V} \setminus \{x\} : f(y) = g(y)\big) \to g \in [\![p]\!]_{M_{\text{ch}}} \}$$

We write $M \vDash p$ as a shorthand for "there is some ch such that $[\![p]\!]_{M_{\text{ch}}} = \mathcal{U}$ holds"; if $M \vDash p$ holds, we say that p is *true* in the model M. The converse, "there is no ch such that $[\![p]\!]_{M_{\text{ch}}} = \mathcal{U}$ holds" is abbreviated as $M \nvDash p$; if it holds, we say that p is *false* in the model M. We call any structure paired with an implicitly existentially-quantified choice function a *model*; it is a model for any language that shares the vocabulary with the structure. It is very important to note that the concept of truth *is meaningless* except relative to a particular model.

There is an important categorization of well-formed formulae based on how they behave across the totality of all models:

Valid formulae: A formula p is valid if and only if it is true in all models (i.e. $M \vDash p$ holds for all models M); we abbreviate this as $\vDash p$.

Invalid formulae: A formula p is invalid if and only if it is false in all models (i.e. $M \nvDash p$ holds for all models M).

Contingent formulae: A formula p is contingent if and only if it is true in some model and false in another model (i.e. there are models M and M' such that both $M \vDash p$ and $M' \nvDash p$ hold).

Satisfiable formulae: A formula p is satisfiable if and only if it is true in some model (i.e. there is a model M such that $M \vDash p$ holds).

Refutable formulae: A formula p is refutable if and only if if it is false some model (i.e. there is a model M such that $M \vDash p$ holds). We abbreviate this as $\nvDash p$.

Here, "all models" means "all models of the language being considered".

We say that a formula q is a *logical conclusion* from a countable set of formulae P, denoted $P \vDash q$, if $M \vDash q$ holds for every model M for which $M \vDash p$ holds for every $p \in P$. We say that two formulae p and q are *logically equivalent*, $p \equiv q$, if for each model M, either both or neither of $M \vDash p$ and $M \vDash q$ hold.

METAPROPOSITION 2.15. *For every well-formed formula p, the assertion $\emptyset \vDash p$ holds if and only if $\vDash p$ holds.*

PROOF. Trivial. □

METAPROPOSITION 2.16. *Logical equivalence is an equivalence relation: for all well-formed formulae p, q, and r,*

(1) *$p \equiv p$ holds (*reflexivity*),*
(2) *if $p \equiv q$ holds, so does $q \equiv p$ (*symmetry*), and*
(3) *if $p \equiv q$ and $q \equiv r$ hold, so does $p \equiv r$ (*transitivity*).*

PROOF. Omitted. □

METAPROPOSITION 2.17. *Let $p_1, \ldots, p_n$ and q be any well-formed formulae (n being finite). Now, $\{p_1, \ldots, p_n\} \vDash q$ holds if and only if $\vDash (p_1 \wedge \cdots \wedge p_n) \rightarrow q$ holds.*

PROOF. Omitted. □

The semantics of a sentence is easily understood, but the same does not hold for well-formed formulae with free variables. However, the following arguments make it clear: any free variables are implicitly universally quantified.

METADEFINITION 2.18. Let $\mathcal{L}_V = (\mathcal{T}_V, \mathcal{F}_V, \mathrm{fv}, \mathrm{Sb})$ be a first-order language over a vocabulary $V = (F, R, a, S, s)$ and let $p \in \mathcal{F}_V$ be a formula. Let $x_1, \ldots, x_n \in \mathcal{V}$ be the free variables of p (i.e. $\{x_1, \ldots, x_n\} = \mathrm{dom}(\mathrm{fv}(p))$ holds). Then the formula

$$(\forall x_1 : \mathrm{fv}(p)(x_1)) \ldots (\forall x_n : \mathrm{fv}(p)(x_n))(p)$$

is the *universal closure* of p.

METAPROPOSITION 2.19. *The choice of how the free variables are ordered in the previous definition is immaterial: two universal closures of the same formula obtained using different orderings of free variables are logically equivalent.*

PROOF. Omitted. □

METAPROPOSITION 2.20. *In any first-order language, a formula and its universal closure are logically equivalent.*

PROOF. We use induction on the number of free variables in the formula. The base case of a sentence is trivial, since the universal closure of a sentence is the sentence itself. Now, assume that the proposition holds for any formula with n free variables. Now, take a formula p with $n+1$ free variables. Choose one of its free variables and denote it by x. Now, let q be the formula $\forall x : \mathrm{fv}(p)(x)$. Let the universal closure of q be q'. Clearly, it has n free variables and, by the induction assumption, q is logically equivalent to q'. Now, if p is logically equivalent to q, then, since logical equivalence is an equivalence relation, $p \equiv q'$ also holds. Clearly, q' is also the universal closure of p. It remains to show that p is logically equivalent to q.

Let M be an arbitrary model of the language being considered, and let $f, g \in \mathcal{U}$ be arbitrarily chosen so that $f(y) = g(y)$ holds for all variables y except x. Now, if $M \vDash q$ holds, then, by definition, there is a ch such that $f \in [\![q]\!]_{M_{\mathrm{ch}}}$ implies $f \in [\![p]\!]_{M_{\mathrm{ch}}}$. Therefore, $[\![q]\!]_{M_{\mathrm{ch}}} \subseteq [\![p]\!]_{M_{\mathrm{ch}}}$ holds, but since $[\![q]\!]_{M_{\mathrm{ch}}}$ is $\mathcal{U}$ by assumption, $[\![p]\!]_{M_{\mathrm{ch}}}$ also is $\mathcal{U}$, and hence $M \vDash p$ also holds. However, if $M \vDash p$ holds, then there is a ch such that $f \in [\![p]\!]_{M_{\mathrm{ch}}}$ holds and by definition $f \in [\![q]\!]_{M_{\mathrm{ch}}}$ holds and we also get $M \vDash q$. Hence, p and q are logically equivalent. □

The definition of a first-order language includes quite many connectives and quantifiers. It is a well-known fact that only few are truly necessary; this is embodied in the following propositions.

METAPROPOSITION 2.21. *Let p, q and r be well-formed formulae in an arbitrary first-order language, let τ be an arbitrary sort in the same language, and let x be a variable. The following logical equivalences hold:*

(1) $$\neg p \equiv p \mid p$$
(2) $$\neg p \equiv p \downarrow p$$
(3) $$p \wedge q \equiv \neg(p \mid q)$$
(4) $$p \vee q \equiv \neg((\neg p) \wedge (\neg q))$$
(5) $$p \rightarrow q \equiv (\neg p) \vee q$$
(6) $$p \leftarrow q \equiv q \rightarrow p$$
(7) $$p \leftrightarrow q \equiv (p \rightarrow q) \wedge (p \leftarrow q)$$
(8) $$p \mid q \equiv \neg(p \wedge q)$$
(9) $$p \downarrow q \equiv \neg(p \vee q)$$
(10) $$p \veebar q \equiv (p \vee q) \wedge \neg(p \wedge q)$$
(11) $$(\forall x : \tau)p \equiv \neg(\exists x : \tau)(\neg p)$$
(12) $$(\exists x : \tau)p \equiv \neg(\forall x : \tau)(\neg p)$$

PROOF. A good but tedious exercise. □

METAPROPOSITION 2.22. *In any first-order language, for every formula there is a logically equivalent formula p*

(1) *for which $(E, (n, q, r)) \prec p$ holds only if $n = 10$ and for which $(E, (4, q)) \prec p$ does not hold for any E or q (i.e. p contains no connectives except Sheffer stroke).*
(2) *for which $(E, (n, q, r)) \prec p$ holds only if $n = 11$ and for which $(E, (4, q)) \prec p$ does not hold for any E or q (i.e. p contains no connectives except Peirce's arrow).*
(3) *for which $(E, (n, q, r)) \prec p$ holds only if $n = 5$ (i.e. p contains no connectives except conjunction and negation).*
(4) *for which $(E, (13, x, \tau, q)) \prec p$ never holds (i.e. p contains no universal quantifiers).*
(5) *for which $(E, (14, x, \tau, q)) \prec p$ never holds (i.e. p contains no existential quantifiers).*

PROOF. An easy exercise. □

Committed choice is not usually included in the definition of first-order logic. We include it, because it is so useful in applications. The following proposition shows why it is often left out.

METAPROPOSITION 2.23. *In any first-order language, for every well-formed formula p, there is an ϵ-free formula p' such that $p \equiv p'$*

PROOF. Let n be the number of ϵ-terms in p. Let p_0 be p; for each $i = 1, \ldots, n$, let p_{i-1} be $p_i[(\epsilon x_i : \tau_i)(q_i)/y_i]$, where x_i are suitably chosen variables, τ_i are suitably chosen sorts, y_i are distinct variables not appearing in any of p_i, and q_i are suitably chosen well-formed formulae. Now, let p' be $(\exists y_1 : \tau_i) \ldots (\exists y_n : \tau_n)(p_n)$. Now, the proposition follows nontrivially. □

Finally, we show that many-sorted languages are, with respect to validity, just syntactic sugar over unsorted languages:

METAPROPOSITION 2.24. *Let $V = (F, R, a, S, s)$ be a many-sorted vocabulary and let $V' = (F, R', a')$ be an unsorted vocabulary with R' being $R \cup S$ and a' being $a \cup (S \times \{1\})$. Now, define functions $f : \mathcal{F}_V \to \mathcal{F}_{V'}$ and $t : \mathcal{T}_V \to \mathcal{T}_{V'}$ by mutual recursion as follows:*

$$\begin{aligned}
t(0, x) &= (0, x) \\
t(1, g, u_1, \ldots, u_{a(g)}) &= (1, g, t(u_1), \ldots, t(u_{a(g)})) \\
t(2, x, \tau, p) &= (2, x, (7, (3, \tau, x), f(p))) \\
f(3, r, u_1, \ldots, u_{a(r)}) &= (3, r, t(u_1), \ldots, t(u_{a(r)})) \\
f(4, p) &= (4, f(p)) \\
f(n, p, q) &= (n, f(p), f(q)) && n = 5, \ldots, 12 \\
f(n, x, \tau, p) &= (n, x, (7, (3, \tau, x), f(p))) && n = 13, 14
\end{aligned}$$

where $x \in \mathcal{V}$, and $g \in F$, and $r \in R$, and $\tau \in S$ are arbitrary. Then $\vDash p$ holds if and only if $A \vDash f(p)$ holds, where A comprises the sort axiom

$$(\forall x)(\tau_1(x) \veebar \cdots \veebar \tau_n(x)) \tag{13}$$

$$(\forall x_1) \ldots (\forall x_{a(g)})((s(g, 0)(x_1) \wedge \cdots \wedge s(g, a(g) - 1)(x_{a(g)})) \leftrightarrow s(g, a(g))(x_1, \ldots, x_n)) \tag{14}$$

$$(\forall x_1) \ldots (\forall x_{a(r)})((s(r, 0)(x_1) \wedge \cdots \wedge s(r, a(r) - 1)(x_{a(r)})) \leftarrow r(x_1, \ldots, x_n)) \tag{15}$$

for all $r \in R$ and $g \in F$, where $\{\tau_1, \ldots, \tau_n\} = S$ holds.

PROOF. Omitted for lack of time. □

5. Inference

METADEFINITION 2.25 (Inference system). An *inference system* for a first-order language is any decidable function $f : \mathcal{P} \times \mathcal{F}_V \to \{0, 1, \bot\}$, where $\mathcal{P}$ is the set of countable subsets of $\mathcal{F}_V$. For a countable set of formulae P and a formula p, we write $P \vdash_f p$ if and only if $f(P, p) = 1$, and $P \nvdash_f p$ if and only if $f(P, p) = 0$.

We write $\vdash p$ for $\emptyset \vdash p$ and $\nvdash p$ for $\emptyset \vdash p$. If $\vdash_f p$ holds, we call p an *f-theorem*. If $P \vdash_f p$ holds only when $P \vDash p$ holds, we say that

f is *sound*. If $P \vdash_f p$ holds whenever $P \vDash p$ holds, we say that f is *complete*.

We will not give examples here, since most of this booklet is occupied in studying inference systems. We will only state two important metatheorems about inference systems.

METATHEOREM 2.26 (Kurt Gödel, 1930). *For every first-order language there is a sound and complete inference system.*

PROOF. Omitted. □

The essence of this theorem (often known as *Gödel's completeness theorem* in reference to his more famous incompleteness results) guarantees that we can, if we want to, disregard the question, which inference system we are using. We say that a formula p is a *theorem* ($\vdash p$) if it is an f-theorem for some sound and complete inference system f for the language being considered.

In our abstract definition, the response $\perp$ is intended to model nontermination. Note that an inference system may be sound and complete but still not terminate for some inputs; in fact, every such system does that, as was shown by Church and Turing independently of each other:

METATHEOREM 2.27 (Alonzo Church and Alan Turing, 1936). *For every first-order language and every sound and complete inference system f for it, there is a formula such that $f(p) = \perp$.*

PROOF. Proceed by showing that the negation of this theorem implies that the halting problem has a solution. □

6. First-order theories

METADEFINITION 2.28 (First-order theories). Let $\mathcal{L}_V = (\mathcal{T}_V, \mathcal{F}_V, \mathrm{fv}, \mathrm{Sb})$ be a first-order language over a vocabulary $V = (F, R, a, S, s)$ and let $A \subseteq \mathcal{F}_V$ be a decidable set of well-formed formulae in which $\mathrm{fv}(a) = \emptyset$ holds for all $a \in A$. Then the pair $T = (V, A)$ is a *first-order theory* and the members of the set A are the *(non-logical) axioms* of the theory.

The only requirement for the set of axioms is that it be decidable, that is, it must be possible to write a computer program that takes any formula as input, successfully terminates for all inputs (given enough resources) and correctly determines whether the input formula is a member of the set of axioms. The set can be empty (though usually isn't), finite or even countably infinite.

For a theory $T = (V, A)$, when write $T \vDash p$, we mean $A \vDash p$; if $T \vDash p$ holds, we say that p is *valid in* T. Similarly, we write $T \vdash p$ for $A \vdash p$; if $T \vdash p$ holds, we say that p is a *theorem of* T. In situations, where it is clear that we are working inside a fixed theory T, we may write $\vDash p$ and $\vdash p$ for $T \vDash p$ and $T \vdash p$, respectively. We call any model of

the language over the vocabulary of a theory a model of the theory if the non-logical axioms of the theory are true in that model.

EXAMPLE 2.29. The first-order theory of *Presburger arithmetic* consists of the vocabulary V of Presburger arithmetic (Ex. 2.2) and the universal closures of the following formulae as axioms:

$$x = y \to (x = z \to y = z) \tag{16}$$

$$x = y \to \mathrm{succ}(x) = \mathrm{succ}(y) \tag{17}$$

$$\neg(0 = \mathrm{succ}(x)) \tag{18}$$

$$\mathrm{succ}(x) = \mathrm{succ}(y) \to x = y \tag{19}$$

$$x + 0 = x \tag{20}$$

$$x + \mathrm{succ}(y) = \mathrm{succ}(x + y) \tag{21}$$

$$(p[0/x]) \to ((\forall x)(p \to p[\mathrm{succ}(x)/x])) \to (\forall x)(p) \tag{22}$$

The schema (22) (the *first-order induction schema*) describes an axiom for each $p \in \mathcal{F}_V$ for which $\mathrm{dom}(\mathrm{fv}(p)) = \{x\}$ holds.

EXAMPLE 2.30. The first-order theory of *first-order arithmetic* (also known as Peano arithmetic, *PA*) consists of the vocabulary of *PA* (Ex. 2.3), and the axioms of Presburger arithmetic augmented with the universal closures of the following formulae:

$$x \times 0 = 0 \tag{23}$$

$$x \times \mathrm{succ}(y) = (x \times y) + x \tag{24}$$

METATHEOREM 2.31 (Kurt Gödel, 1931). *In the first-order theory of Peano arithmetic, there is a formula p such that $\not\models p$ and $\not\models \neg p$ holds.*

PROOF. By embedding the first-order theory of Peano arithmetic in the first-order theory of Peano arithmetic, one can express the provability of a PA formula inside PA; use this to construct an embedded sentence for which you can demonstrate that no proof can be found for the sentence itself nor for its negation. Then invoke Metatheorem 2.26 translate this unprovability into invalidity. Details of this proof can be found in many books in mathematical logic. □

This is, of course, a special case of Gödel's famous first incompleteness metatheorem[1], which, in full generality, states that any consistent theory capable of expressing Peano arithmetic has at least one formula such that neither it nor its negation is a theorem and therefore valid. The significance of this metatheorem has been overstressed; while it quite effectively destroyed Hilbert's program, it

[1]Kurt Gödel: *On formally undecidable propositions of Principia Mathematica and related systems*, New York, Dover (1992). Originally published as "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme" in *Monatshefte für Mathematik und Physik*, vol. 38 (1931)

does not make formalization or mathematics useless. The real significance is summarized by the following metacorollary:

METACOROLLARY 2.32. *Assuming that the first-order theory of Peano arithmetic has a model, it has at least one nonstandard model.*

By a nonstandard model we mean a model that does not make true the same formulae as the standard model. In PA's case, the standard model was described in Ex. 2.7.

PROOF. We can assume that there is at least one model M of PA. Now, if it were the only model, then by Metatheorem 2.31 there is a formula p such that $M \vDash p$ and $M \nvDash p$ both hold; this is, however, impossible. Therefore, there are at least two models. Only one of them can be the standard model, and therefore the other must be nonstandard. □

METADEFINITION 2.33 (Subtheories). Let T and T' be two theories over the same vocabulary V and let $A \in \mathcal{F}_V$ and $A' \in \mathcal{F}_V$ be the conjunctions of their axioms, respectively. Now, T' is a *subtheory* of T if $\vDash A \to A'$ holds.

EXAMPLE 2.34. The first-order theory of *second-order arithmetic* consists of the vocabulary of second-order arithmentic (Ex. 2.4), and the axioms of first-order arithmetic (so modified that every quantification uses the sort N), not including (22), augmented with the universal closures of the following formulae:

$$(0 \in X \wedge (\forall n : N)(n \in X \to \mathrm{succ}(n) \in X)) \to (\forall n : N)(n \in X) \tag{25}$$

$$(\exists X : C)((\forall n : N)(n \in X \leftrightarrow p)) \tag{26}$$

In the formula (25) (*second-order induction*), the variable X is to be taken as being of sort C; note that the formula is not a schema. The schema (26) (the *second-order comprehension schema*) describes an axiom for each $p \in \mathcal{F}_V$ for which $x \in \mathrm{dom}(\mathrm{fv}(p))$ and $X \notin \mathrm{dom}(\mathrm{fv}(p))$ hold.

Second-oder arithmetic is known to be sufficiently powerful that most of ordinary mathematics may be performed in it (though not always clearly). It is also important here because it has one subsystem that can be finitely axiomatized — ACA_0 — and finite axiomatizability is a big asset for automation of a theory.

EXAMPLE 2.35. The first-order theory ACA_0 is formed from the first-order theory of second-order arithmetic by replacing its second-order comprehension schema (26) with the *arithmetic comprehension schema*, which has the same form as the second-order comprehension schema except that p is further restricted to *arithmetic formulae*, i.e. formulae that do not contain any variables (bound or free) with the sort C.

METADEFINITION 2.36 (First-order theories with equality). Let $\mathcal{L}_V = (\mathcal{T}_V, \mathcal{F}_V, \mathrm{fv}, \mathrm{Sb})$ be a first-order language over an equality vocabulary $V = (F, R, a, S, s)$ with $=_\tau$ denoting the equality predicate symbol for sort τ, and let $T = (V, A)$ be a first-order theory.

We define the set of *equality axioms* $\mathcal{E}_V$ for V as consisting of all axioms of the forms

$$(\forall x : \tau)(x =_\tau x) \tag{27}$$

$$(\forall x_1 : \tau_{f,1}) \cdots (\forall x_{n_f} : \tau_{f,n_f})(\forall y_1 : \tau_{f,1}) \cdots (\forall y_{n_f} : \tau_{f,n_f}) \tag{28}$$
$$((x_1 =_{\tau_{f,1}} y_1) \wedge \cdots \wedge (x_{n_f} =_{\tau_{f,n_f}} y_{n_f})) \rightarrow (f(x_1, \ldots, x_{n_f}) =_{\tau_{f,n_f}} f(y_1, \ldots, y_{n_f}))$$

$$(\forall x_1 : \tau_{f,1}) \cdots (\forall x_{n_r} : \tau_{r,n_r})(\forall y_1 : \tau_{r,1}) \cdots (\forall y_{n_r} : \tau_{r,n_r}) \tag{29}$$
$$((x_1 =_{\tau_{r,1}} y_1) \wedge \cdots \wedge (x_{n_r} =_{\tau_{r,n_r}} y_{n_r})) \rightarrow (r(x_1, \ldots, x_{n_r}) \rightarrow r(y_1, \ldots, y_{n_r}))$$

for all sorts $\tau \in F$, all function symbols $f \in F$ and all predicate symbols $r \in R$, where $\tau_{f,i}$ denotes $s(f, i-1)$ and n_f denotes $a(f)$.

Let there be a set A' such that $A = A' \cup \mathcal{E}_V$ and $A' \cap \mathcal{E}_V = \emptyset$ hold. Now, T is a *first-order theory with equality* and its *(non-logical) axioms* are the elements of the set A'. If R contains only the equality predicate symbols, then T is also called an *equational theory*.

Note that the term *non-logical axioms* is now overloaded. When a first-order theory with equality is viewed as a plain first-order theory, then its non-logical axioms is larger than what it is when it it is viewed as a first-order theory with equality: when a theory is viewed without equality, the equality axioms are considered non-logical, but when a theory is viewed with equality, the equality axioms are considered logical axioms.