Chapter 5

Type Theory

§50. Introduction

So far we have been concerned with first-order logic, and its subsystem propositional calculus, which we might regard as zeroth-order logic. We now wish to discuss higher-order logics.

It seems very natural to extend the system $\mathcal{F}$ of first-order logic by permitting quantification on predicate, propositional, and function variables as well as individual variables. One thus obtains the wffs of a system of *second-order logic*. One might then introduce predicate and function variables of higher type to denote relations and functions whose arguments may be relations and functions of individuals as well as individuals. Thus one would be led to a system of *third-order logic*, and if one permitted quantification with respect to these new variables, one would obtain a system of *fourth-order logic*. This process can be continued indefinitely to obtain logics of arbitrarily high order. Of course, after a while one runs out of different types of letters to use for different types of variables, but this problem can be solved by introducing *type symbols* to indicate the types of variables, and using a letter with type symbol α as subscript for a variable of type α.

To make these ideas precise, we shall briefly describe a system $\mathcal{F}^\omega$ of ω-order logic which has all finite order logics as subsystems. We shall not discuss $\mathcal{F}^\omega$ very extensively, since we prefer to devote most of our attention to an improved formulation of higher-order logic called $\mathcal{Q}_0$ which we shall soon introduce. Nevertheless, $\mathcal{F}^\omega$ provides a convenient starting point for a discussion of higher-order logics. For the sake of simplicity, $\mathcal{F}^\omega$ will have no function symbols. It is well known that assertions about functions can be expressed in terms of relations, since to every n-ary function f there

corresponds an $(n+1)$-ary relation R such that for all $x_1, \ldots, x_n$, and y, $Rx_1 \ldots x_n y$ iff $fx_1 \ldots x_n = y$.

A system of logic which includes logics of all finite orders (but no infinite orders) is known as *ω-order logic*, or *finite type theory*. (There are also systems of transfinite type theory, such as that in [Andrews, 1965], but we shall not discuss them here.) Type theory was invented by Bertrand Russell [Russell, 1908]. In [Whitehead and Russell, 1913], which had extraordinary influence on the development of logic in the early twentieth century, he and Alfred North Whitehead demonstrated that various fundamental parts of mathematics could indeed be formalized in this system. Russell was concerned with providing secure foundations for mathematics, and wished to enforce the *vicious-circle principle* that no totality can contain members defined in terms of itself. He therefore originally advocated using what is now called *ramified* type theory,[1] which is still important in various proof-theoretic studies (such as [Feferman, 1964]) of the consistency of restricted portions of mathematics. However, ramified type theory did not prove adequate for formalizing mathematics in general, and we shall devote our attention to a simplified version of it which is known as *simple* type theory.

Additional information about type theory may be obtained from sources such as [Andrews, 2001], [van Benthem and Doets, 1983], [Hatcher, 1968], [Hindley, 1997], [Jacobs, 1999], [Leivant, 1994], [Mendelson, 1987], [Quine, 1963], [Shapiro, 1991], [Wolfram, 1993], and references cited therein.

The *type symbols* of $\mathcal{F}^\omega$ and their *orders* are defined inductively as follows:

(a) ι is a type symbol (denoting the type of individuals) of order 0.
(b) o is a type symbol (denoting the type of truth values) of order 1.
(c) If $\tau_1, \ldots, \tau_n$ are type symbols (with $n \geq 1$), then $(\tau_1 \ldots \tau_n)$ is a type symbol (denoting the type of n-ary relations with arguments of types $\tau_1, \ldots, \tau_n$, respectively), and its order is 1+ the maximum of the orders of $\tau_1, \ldots, \tau_n$. (One may think of "o" as "( )".)

The primitive symbols of $\mathcal{F}^\omega$ are the improper symbols [,], $\sim$, $\vee$, and $\forall$, a denumerable list of variables of each type, and (optionally) constants of certain types. Variables of type o are called *propositional variables*, and variables of type ι are called *individual variables*.

The formation rules of $\mathcal{F}^\omega$ are the following:

[1] See [Hazen, 1983] or [Church, 1956, p. 347] for more details.

(a) Every propositional variable or constant is a wff.
(b) If $\mathbf{u}_{\tau_1 \ldots \tau_n}, \mathbf{v}^1_{\tau_1} \ldots, \mathbf{v}^n_{\tau_n}$ are variables or constants of the types indicated by their subscripts, then $\mathbf{u}_{\tau_1 \ldots \tau_n} \mathbf{v}^1_{\tau_1} \ldots \mathbf{v}^n_{\tau_n}$ is a wff.
(c) If $\mathbf{A}$ and $\mathbf{B}$ are wffs and $\mathbf{u}$ is a variable of any type, then $\sim \mathbf{A}$, $[\mathbf{A} \vee \mathbf{B}]$, and $\forall \mathbf{u} \mathbf{A}$ are wffs.

The *order* of a variable or constant is the order of its type symbol. For any positive integer m, the wffs of $2m$th-order logic are the wffs in which no variable or constant of order greater than m occurs, and the wffs of $(2m-1)$th-order logic are the wffs of $2m$th-order logic in which no variable of order m is quantified. (Note that first-order logic, as here defined, is just the result of dropping function symbols from the system $\mathcal{F}$ of §20.)

The definitions of connectives and quantifiers in §10 and §20 can be extended to apply to $\mathcal{F}^\omega$. If $\mathbf{x}_\tau$ and $\mathbf{y}_\tau$ are variables or constants of the same type τ, $\mathbf{x}_\tau = \mathbf{y}_\tau$ is defined to be an abbreviation for

$$\forall z_{(\tau)} [z_{(\tau)} \mathbf{x}_\tau \supset z_{(\tau)} \mathbf{y}_\tau] \tag{*1}$$

where $z_{(\tau)}$ is a variable of type (τ). An alternative possibility is to define $\mathbf{x}_\tau = \mathbf{y}_\tau$ as

$$\forall p_{(\tau\tau)} [\forall z_\tau p_{(\tau\tau)} z_\tau z_\tau \supset p_{(\tau\tau)} \mathbf{x}_\tau \mathbf{y}_\tau]. \tag{*2}$$

(See Exercise X5007.) Note that the two basic properties of equality which we took as axioms in §26 are reflexivity and substitutivity. *1 defines equality in terms of substitutivity, and *2 defines equality in terms of reflexivity (indeed, as the intersection of all reflexive relations). Thus, all the properties of equality follow from either of the basic properties of substitutivity or reflexivity when these are expressed in a sufficiently powerful way using the expressiveness of higher-order logic.

The rules of inference of $\mathcal{F}^\omega$ are:

(1) **Modus Ponens**. From $\mathbf{A}$ and $\mathbf{A} \supset \mathbf{B}$ to infer $\mathbf{B}$.
(2) **Generalization**. From $\mathbf{A}$ to infer $\forall \mathbf{x} \mathbf{A}$, where $\mathbf{x}$ is a variable of any type.

The axiom schemata of $\mathcal{F}^\omega$ are:

(1) $\mathbf{A} \vee \mathbf{A} \supset \mathbf{A}$
(2) $\mathbf{A} \supset . \mathbf{B} \vee \mathbf{A}$
(3) $\mathbf{A} \supset \mathbf{B} \supset . \mathbf{C} \vee \mathbf{A} \supset \mathbf{B} \vee \mathbf{C}$

(4) $\forall \mathbf{x}_\tau \mathbf{A} \supset S^{\mathbf{x}_\tau}_{\mathbf{y}_\tau}\mathbf{A}$, where $\mathbf{y}_\tau$ is a variable or constant of the same type as the variable $\mathbf{x}_\tau$, and $\mathbf{y}_\tau$ is free for $\mathbf{x}_\tau$ in $\mathbf{A}$.

(5) $\forall \mathbf{x}[\mathbf{A} \vee \mathbf{B}] \supset \ \centerdot\, \mathbf{A} \vee \forall \mathbf{x}\mathbf{B}$, where $\mathbf{x}$ is any variable not free in $\mathbf{A}$.

(6) **(Comprehension Axioms)**

$\exists \mathbf{u}_\circ[\mathbf{u}_\circ \equiv \mathbf{A}]$, where $\mathbf{u}_\circ$ does not occur free in $\mathbf{A}$.

$\exists \mathbf{u}_{(\tau_1 \ldots \tau_n)} \forall \mathbf{v}^1_{\tau_1} \ldots \forall \mathbf{v}^n_{\tau_n} [\mathbf{u}_{(\tau_1 \ldots \tau_n)} \mathbf{v}^1_{\tau_1} \ldots \mathbf{v}^n_{\tau_n} \equiv \mathbf{A}]$, where $\mathbf{u}_{(\tau_1 \ldots \tau_n)}$ does not occur free in $\mathbf{A}$, and $\mathbf{v}^1_{\tau_1}, \ldots, \mathbf{v}^n_{\tau_n}$ are distinct variables.

(7) **(Axioms of Extensionality)**

$[x_\circ \equiv y_\circ] \supset \ \centerdot\, x_\circ = y_\circ$

$\forall w^1_{\tau_1} \ldots \forall w^n_{\tau_n} [x_{(\tau_1 \ldots \tau_n)} w^1_{\tau_1} \ldots w^n_{\tau_n} \equiv y_{(\tau_1 \ldots \tau_n)} w^1_{\tau_1} \ldots w^n_{\tau_n}] \supset \ \centerdot\, x_{(\tau_1 \ldots \tau_n)} = y_{(\tau_1 \ldots \tau_n)}$

The axioms of kth-order logic are those axioms of $\mathcal{F}^\omega$ which are wffs of kth-order logic. Note that Comprehension Axioms first appear in second-order logic, and Axioms of Extensionality first appear in fourth-order logic.

Note the principal features which are added to first-order logic in order to obtain this formulation of higher-order logic:

1. Variables of arbitrarily high orders.
2. Quantification on variables of all types.
3. Comprehension Axioms.
4. Axioms of Extensionality.

One may also wish to assume additional axioms, such as the Axiom Schema of Choice, an Axiom of Infinity, and perhaps even the Continuum Hypothesis, though there is room for argument whether these are axioms of pure logic or of mathematics.

It should be remarked that one can restrict the formation rules of $\mathcal{F}^\omega$ by requiring that $n = 1$ in the definitions above, and obtain a still simpler system of ω-order logic which is nevertheless adequate to express most mathematical ideas, since one can define ordered pairs (and, more generally, n-tuples) in various ways, and then represent an n-ary relation B by the monadic relation P such that $Bx^1 \ldots x^n$ iff $P\langle x_1, \ldots, x_n \rangle$. Thus, $P\langle x^1, \ldots, x^n \rangle$ would be an abbreviation for $\exists u[Pu \wedge u = \langle x^1, \ldots, x^n \rangle]$, where $u = \langle x^1, \ldots, x^n \rangle$ is defined in some appropriate way, and suitable types are assigned to u and P.

If one wishes to thus restrict the language so that all predicates are monadic, one may find it congenial to write $\mathbf{v}_\tau \in \mathbf{u}_{(\tau)}$ rather than $\mathbf{u}_{(\tau)}\mathbf{v}_\tau$,

and to say "$\mathbf{v}_\tau$ is in the set $\mathbf{u}_{(\tau)}$" rather than "$\mathbf{v}_\tau$ has the property $\mathbf{u}_{(\tau)}$". Of course, this is a mere notational change. A more radical change is obtained by also dropping the type symbols, so that all variables are of the same type. (Propositional variables are customarily dispensed with.) One thus obtains Naive Axiomatic Set Theory, which can be formalized in first-order logic. $\in$ is regarded as a binary predicate constant, and all atomic wffs are of the form $\mathbf{x} \in \mathbf{y}$, where $\mathbf{x}$ and $\mathbf{y}$ are (individual) variables. $\mathbf{x} \in \mathbf{y}$ is interpreted to mean "$\mathbf{x}$ is a member of the set $\mathbf{y}$", which is of course false if $\mathbf{y}$ is not a set at all. The Comprehension Axiom Schema now takes the simple form $\exists \mathbf{u} \forall \mathbf{v}[\mathbf{v} \in \mathbf{u} \equiv \mathbf{A}]$, where $\mathbf{u}$ is not free in the wff $\mathbf{A}$. This axiom schema asserts that for any wff $\mathbf{A}$ in which $\mathbf{v}$ occurs free, there is a set $\mathbf{u}$ consisting of all those elements $\mathbf{v}$ of which $\mathbf{A}$ is true. Unfortunately, the axiom schema guarantees the existence of too many sets. In particular, $\exists u \forall v[v \in u \equiv \sim v \in v]$, i.e., there is a set whose members are all those elements which are not members of themselves. Choosing such a set u, we conclude that $u \in u \equiv \sim u \in u$, which is a contradiction, so Naive Axiomatic Set Theory is inconsistent! The paradox above was discovered by Bertrand Russell, and is known as Russell's Paradox.

One can restrict the Comprehension Axiom Schema, and take only certain carefully chosen instances of it as axioms, to obtain apparently consistent formulations of Axiomatic Set Theory. Alternatively, one can keep type symbols, so that Russell's paradox (and various other paradoxes) cannot be expressed at all within the formal system. Each approach leads to systems which are adequate for formalizing mathematics, and each approach has advantages for certain purposes. At present, Axiomatic Set Theory is more popular with mathematicians simply because it is much better known. However, most mathematicians do make mental distinctions between different types of mathematical objects, and use different types of letters to denote them, so it might be claimed that type theory provides a more natural formalization of mathematics as it is actually practiced. In addition, explicit syntactic distinctions between expressions denoting intuitively different types of mathematical entities are very useful in computerized systems for exploring and applying mathematics (on a theoretical rather than purely computational level). Of course, type symbols need not be explicitly written as long as one knows the types of the variables and constants.

Therefore we shall turn our attention to finding a formulation of type theory which is as expressive as possible, allowing mathematical ideas to be expressed precisely with a minimum of circumlocutions, and which is as simple and economical as is possible without sacrificing expressiveness. The reader will observe that the formal language we find arises very naturally

http://www.springer.com/978-1-4020-0763-7

An Introduction to Mathematical Logic and Type Theory
To Truth Through Proof
Andrews, P.B.
2002, XVIII, 390 p., Hardcover
ISBN: 978-1-4020-0763-7