A theory of propositional types *

by

L. Henkin** (Berkley, Calif.)

§ 1. Let \( \mathcal{D}_0 \) be the set of the two truth values, \( T \) and \( F \), and consider the operation of passing from any two sets \( \mathcal{D}_a \) and \( \mathcal{D}_b \) to the set \( \mathcal{D}_{ab} \) of all functions which map \( \mathcal{D}_b \) into \( \mathcal{D}_a \). The family \( \mathfrak{P}_L \) of all sets generated from \( \mathcal{D}_0 \) by repeated application of this operation we call the family of propositional types. Thus \( \mathfrak{P}_L \) is the least class of sets, containing \( \mathcal{D}_0 \) as an element, which is closed under passage from \( \mathcal{D}_a \) and \( \mathcal{D}_b \) to \( \mathcal{D}_{ab} \).

In the ordinary propositional (or sentential) logic we have variables which range over \( \mathcal{D}_0 \), and a constant—the negation sign—which denotes one of the 4 elements of \( \mathcal{D}_0 \). We also have other constants, such as the connective \( \land \) for conjunction, which denote binary operations on \( \mathcal{D}_0 \). Such operations may be identified with elements of \( \mathcal{D}_{2m} \) in a familiar way; for example, the operation \( \land^x \) denoted by the symbol \( \land \) is described by the equations \( (\land^x T) F = T \), \( (\land^x F) F = F \), and \( (\land^x F) x = F \) for all \( x \in \mathcal{D}_0 \). By means of formulas built up from propositional variables and connectives we may refer to particular elements of the propositional types \( \mathcal{D}_2 \), \( \mathcal{D}_{2m} \), \( \mathcal{D}_{2m+1} \), \( \mathcal{D}_{2m+2} \), ...,).

In the present paper we shall construct a theory with a distinct set of variables for each propositional type \( \mathcal{D}_0 \). The theory will be couched in a language which permits these variables to occur bound as well as free.

Theories of this kind were first studied by Leśniewski under the name prototetic. An account of Leśniewski's systems is given by Grzegorczyk in [3].

In the systems of prototetic there is incorporated a rule of definition which allows for the introduction of new symbols as names of arbitrary elements of any propositional type. In the present system we start with names for only a relatively few elements, but we allow for the construction of new names by means of variables and the functional abstructor \( \lambda \). We shall prove that each element possesses a name in our system.

---

* A portion of this work was supported by the U. S. National Science Foundation (Grant No. 0114006).

** This paper was written while the author served as Fellow of the John Simon Guggenheim Memorial Foundation at the Institute for Advanced Study, Princeton, New Jersey.
A theory of types which incorporates the abstractor \( \lambda \) was first given by Church in \([1]\), and our present formulation owes much to his. We depart from Church, however, in taking as our only primitive constants the symbols \( \Omega_{\alpha\beta} \), which correspond, for each \( \alpha \), to the identity relations over the propositional type \( \mathcal{D}_\alpha \). The fact that all of the ordinary propositional connectives can be defined in terms of \( \Omega_{\alpha\beta} \) (the usual biconditional) by means of quantified variables over \( \mathcal{D}_\alpha \) is due to Tarski in \([5]\). The fact that the classical quantifiers themselves can be defined in terms of the symbols \( \Omega_{\alpha\beta} \), with the aid of \( \lambda \), does not seem to be stated explicitly in the literature \([1]\).

Our theory is provided with a deductive apparatus consisting of axioms and formal rules of inference. We have been at pains not merely to translate into our primitive notation one of the usual systems of axioms and rules of inference governing sentential connectives and quantifiers, but to find a deductive basis for our theory which seems to express in a natural manner the fundamental properties of our primitive notions.

Our deductive formalization is complete, in the sense that if a formula \( \varphi \) has the value \( T \) for every assignment of values to its free variables, then \( \varphi \) is provable in our system \([6]\). (Conversely, of course, only such valid formulas can be proved.) The completeness of a theory of types in terms of non-standard models was proved in \([7]\), but this result does not seem to imply our present completeness theorem. It is true that by adding suitably to the earlier proof the present result can be obtained, but such a proof would not have the constructive character possessed by the usual completeness proofs for propositional logic, and we have preferred therefore to indicate another method of proof which seems more appropriate for a theory of types each of which is finite.

Our interest was drawn to a theory of propositional types by the problem of constructing non-standard models of a full theory of types. Since many problems of ordinary predicate logic can be reduced to questions about propositional logic (as in Herbrand's theorem, for example), our hopes have been that insight into the totality of models for a full theory of types could be obtained from a study of all models of

the much simpler propositional type theory. We reserve for a future paper, however, a discussion of the models of our present system other than the standard model \( \mathbb{N} \) of propositional types.

\section{2} We now describe the symbolism which underlies our theory.

2.1. The primitive symbols of our system consist of three symbols which are called improper, namely the left and right parentheses and the lower case Greek lambda, and an infinite number of other symbols (proper symbols), each of which is associated with one of the propositional types \( \mathcal{D}_\alpha \). In fact, corresponding to each \( \mathcal{D}_\alpha \) we provide a sequence of proper symbols \( \varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \ldots \), called variables of type \( \mathcal{D}_\alpha \) (or simply of type \( \alpha \)), and a single proper symbol \( Q_{\alpha\beta} \), called a constant of type \( (\alpha\beta) \).

2.2. Certain strings of primitive symbols are called formulas, and each formula is associated with a unique type \( \mathcal{D}_\alpha \). In any formula each occurrence of a variable is distinguished as free or bound. The rules for constructing formulas and distinguishing between free and bound occurrences of variables are given inductively below with the aid of the symbols \( 'X' \), \( 'Y' \), \( 'Z' \), which we use henceforth as metamathematical variables ranging over the set of variables of type \( \alpha \), and the symbols \( 'A_\alpha', 'B_\alpha', 'C_\alpha' \), which are used to range over arbitrary formulas of type \( \alpha \).

(i) A string consisting only of a proper symbol of type \( \alpha \) is a formula of type \( \alpha \); and if it is a variable, this occurrence is free in the formula.

(ii) If \( A_\alpha \) and \( B_\beta \) are formulas of type \( \alpha \beta \) and \( \beta \) respectively, then the string \( (A_\alpha B_\beta) \) is a formula of type \( \alpha \); and an occurrence of a variable in this formula is free or bound according as it is free or bound in that one of the formulas \( A_\alpha \) or \( B_\beta \) in which it occurs.

(iii) If \( A_\alpha \) is any formula of type \( \alpha \) and \( X_\beta \) is any variable of type \( \beta \) then the string \( (\lambda X_\beta A_\alpha) \) is a formula of type \( \alpha \beta \); every occurrence of the variable \( X_\beta \) is bound in this formula, and an occurrence of any other variables is free or bound according as it is free or bound in \( A_\alpha \).

A formula containing no free occurrence of a variable will be called closed.

In practice we shall often omit parentheses, in describing formulas, in accordance with the following conventions. (a) Parentheses at the beginning or end of a formula may be omitted. (b) A dot may replace a left parenthesis, and its mate will be suppressed, if the mate comes at the end of the formula. (c) Parentheses may be omitted if their use is to indicate association to the left in a sequence of three or more expressions. For example,

\[ \lambda x_1. \lambda y_2. \Omega_{\alpha\beta} \lambda z_3. f_{\alpha\beta}(\lambda p(q_4, z_3)) \]
§ 3. We now describe an interpretation of our symbolism to indicate how the formulas may be used to refer to elements of the types \( \mathcal{D} \). Under this interpretation each closed formula of type \( a \) will denote a unique element of \( \mathcal{D} \), but a formula of type \( a \) which contains free occurrences of variables will only refer to a specific element of \( \mathcal{D} \), when values (of appropriate type) are assigned to these variables.

3.1. By an assignment we mean a function \( \varphi \), having as its domain the set of all variables of our system, such that to each variable \( x_\alpha \) of type \( a \) the function \( \varphi \) assigns as value an element \( \varphi(x_\alpha) \) of \( \mathcal{D} \). With each formula \( A_\alpha \) and assignment \( \varphi \) we associate an element \( V(A_\alpha, \varphi) \) of \( \mathcal{D} \), as follows:

(i) If \( x_\alpha \) is any variable then \( V(x_\alpha, \varphi) = \varphi(x_\alpha) \).

(ii) If \( \varphi \) is the element \( f \) of \( \mathcal{D} \) such that for any \( x, y \in \mathcal{D} \), we have \( (fx)y = T \) if \( x = y \) and \( (fx)y = F \) if \( x \neq y \).

(iii) If \( A_\alpha \) and \( B_\beta \) are any formulas of type \( a \) and \( b \) respectively, then \( V((A_\alpha)B_\beta, \varphi) \) is the element of \( \mathcal{D} \) obtained by operating on the element \( V(A_\alpha, \varphi) \) of \( \mathcal{D} \) with the function \( V(A_\alpha, \varphi) \) of \( \mathcal{D} \).

(iv) If \( A_\alpha \) is any formula of type \( a \) and \( x_\alpha \) is any variable of type \( a \), then \( V(x_\alphaA_\alpha, \varphi) \) is the function of \( \mathcal{D} \) whose value, for any \( x \in \mathcal{D} \), is the element \( V(A_\alpha, \varphi) \) of \( \mathcal{D} \), where \( \varphi_0 \) is the assignment such that \( (\varphi_0(x_\alpha) = x \) and \( (\varphi_0(x_\alpha) = \varphi(x_\alpha) \) for all \( x \neq x_\alpha \).

3.2. It is easy to show by induction that if \( A_\alpha \) is any formula, and if \( x_\alpha \) and \( y_\alpha \) are any assignments such that \( \varphi(x_\alpha) = \varphi(y_\alpha) \) for every variable \( x_\alpha \) which occurs free in \( A_\alpha \), then \( V(A_\alpha, \varphi) = V(A_\alpha, \varphi) \).

In particular, if \( A_\alpha \) is closed then \( V(A_\alpha, \varphi) \) is independent of \( \varphi \); this element of \( \mathcal{D} \) is then called the denotation of \( A_\alpha \) and we shall put \( V(A_\alpha) = V(A_\alpha, \varphi) \) in this case. For example, \( (x_\alpha y_\alpha x_\alpha y_\alpha) \) is the identity function of \( \mathcal{D} \) such that \( (x_\alpha y_\alpha x_\alpha y_\alpha)(x_\alpha y_\alpha x_\alpha y_\alpha) = y_\alpha \) for all \( y_\alpha \in \mathcal{D} \).

A formula \( A_\alpha \) is called valid if and only if \( V(A_\alpha, \varphi) = T \) for all assignments \( \varphi \). In particular, a closed formula \( A_\alpha \) is valid if and only if \( (A_\alpha)^\varphi = T \).

§ 4. We now apply the rules of the preceding section to examine the meaning of certain special formulas under our interpretation. In particular, we shall associate, with each element \( x \) of an arbitrary type \( \mathcal{D} \), a closed formula \( \psi(x) \) of type \( a \) such that \( (\psi(x))^\varphi = x \).

4.1. For any formulas \( A_\alpha \) and \( B_\beta \) of type \( a \) we let \( A_\alpha = B_\beta \) be the formula \( (\mathcal{D}A_\alphaB_\beta) \) of type \( 0 \). Clearly \( V(A_\alpha = B_\beta, \varphi) = T \) if \( \varphi(B_\beta) \) according as \( V(A_\alpha, \varphi) = V(B_\beta, \varphi) \) or \( V(A_\alpha, \varphi) \neq V(B_\beta, \varphi) \).

4.2. We put \( T^a = (\lambda(x_\alpha)) \) and \( F^a = (\lambda(x_\alpha)) \). Clearly \( T^a \) and \( F^a \) are closed formulas of type \( 0 \), and we have \( (T^a)^\varphi = T \) and \( (F^a)^\varphi = F \).
is T or F according as \( \varphi(x_a) = \lambda(x_{a_0}) \) or \( \varphi(x_a) \neq \lambda(x_{a_0}) \). Hence

\[
V(\exists x \in \mathcal{A}, \varphi(x_a) \land \forall x \in \mathcal{A}, \varphi(x_a - x_{a_0} = y_0)) \mid \varphi = \lambda(x_{a_0})
\]

and therefore \( \lambda(x_{a_0}) \varphi \neq \varphi \) for this \( \varphi \).

On the other hand, suppose that \( g \) is any element of \( \mathcal{D}_{a_0} \) for which there is no \( h \in \mathcal{D}_{a_0} \), or more than one \( h \in \mathcal{D}_{a_0} \), such that \( (g,h) = T \). Then for any assignment \( \varphi \) such that \( \mapsto \varphi \varphi = g \) we have

\[
V(\exists x \in \mathcal{A}, \varphi(x_a) \land \forall x \in \mathcal{A}, \varphi(x_a - x_{a_0} = y_0)) \mid \varphi = \lambda(x_{a_0})
\]

Hence

\[
V[[\exists x \in \mathcal{A}, \varphi(x_a) \land \forall x \in \mathcal{A}, \varphi(x_a - x_{a_0} = y_0)) \mid \varphi = \lambda(x_{a_0})]
\]

Using 4.8 we obtain \( \lambda(x_{a_0}) \varphi = a_0 \) for this \( \varphi \).

Thus for every \( g \in \mathcal{D}_{a_0} \) we see that \( \lambda(x_{a_0}) \varphi = \varphi \), which completes the demonstration that \( \lambda(x_{a_0}) \varphi \) has the required property.

4.10. We are now ready to assign to each element \( x \) of any type \( \mathcal{D} \) a name, i.e., a closed formula \( x^* \) of type \( a \) such that \( (x^*)^a = x \). Indeed, if \( x \) is either of the elements of \( \mathcal{D} \), this has already been done in 4.2. Hence we may proceed by induction.

Suppose that \( y_1, \ldots, y_4 \) are distinct and are all of the elements of \( \mathcal{D}_y \), and let us make the induction hypothesis that to every \( x \) of \( \mathcal{D}_y \) or \( \mathcal{D}_a \) we have already assigned a name \( x^* \). Let \( f \) be any element of \( \mathcal{D}_{xy} \). Then we take

\[
f^* = [\lambda x \in \mathcal{A}_x, \varphi(x_a = y_a) \land \varphi(x_b = y_b) \land \varphi(x_c = y_c) \land \varphi(x_d = y_d)]
\]

Now consider any assignment \( \varphi \), and say \( \varphi(x_a) = y_0 \). Clearly

\[
V(\exists x \in \mathcal{A}_x, \varphi(x_a = y_a) \land \varphi(x_b = y_b) \land \varphi(x_c = y_c) \land \varphi(x_d = y_d)) \mid \varphi = \lambda(x_{a_0})
\]

so that \( (f^*)^a = y_0 \). Since this is true for each \( i = 1, \ldots, 4 \) we get

\[
f^* = f, \text{ as claimed (11).}
\]

Section 5. We turn now to the formulation of a formal deductive system, based upon the symbolism of section 3 above, by providing axioms and rules of inference.

5.1. Certain formulas of type 0 are called axioms. These are described under seven headings, below, some of which comprise single axioms and

(1) In place of \( f^* \) we could have used the simpler formula \( a_0 = (k_{a_0} \lambda_{a_1} y_{a_1}^a) = ((y_0) = (y_0)) \), but this would not be convenient for use in § 8 below.
others infinitely many axioms grouped in a single schema. In formulating schemes we use 'A', 'B', and 'C' to refer to arbitrary types, 'A', 'B', 'C' for arbitrary formulas whose type is indicated by means of a subscript, and 'X', 'Y', and 'Z' for arbitrary variables (with a similar indication of type).

5.1.1. Axiom Schema 1. \( A_A = A_A \).

5.1.2. Axiom Schema 2. \( (A_A = B) = A_A \).

5.1.3. Axiom 3. \( (T' \land E') = F' \).

5.1.4. Axiom Schema 4. \( (g_A T' \land g_A F') = \forall X_A (g_A X_A) \).

5.1.5. Axiom 5.

\[ (x_A = y_A) \rightarrow \] \( (f_A = g_A) \rightarrow \) \( (f_A X_A) = (g_A X_A) \).


\[ (\forall X_A (f_A X_A = g_A X_A)) \rightarrow (f_A = g_A) \].

5.1.7. Axiom Schema 7. \( (\lambda X_B A_B) A_A = C_A \) where \( C_A \) is obtained from \( B \) by replacing each free occurrence of \( X_B \) in \( C_A \) by an occurrence of \( A_A \), providing no such occurrence of \( X_B \) is within a part of \( B \) which is a formula beginning ‘\( \lambda X_B \)’ where \( X_B \) is a variable free in \( A_A \).

5.2. By the Rule of Replacement we refer to the ternary relation on formulas of type \( \theta \) which holds for \( (A_A, C_B, D_B) \) if and only if \( A_A = (B_1, C_B, D_B) \) for some formulas \( A_A, B_1, C_B, D_B \). We shall say that \( B_1 \) is obtained by Rule R from \( A_A = B_1, C_B = C_B \), and \( D_B \) is obtained from \( C_B \) by replacing one occurrence of \( A_A \) by an occurrence of \( B_1 \). When this relation holds for \( (A_A = B_1, C_B, D_B) \) we shall say that \( D_B \) is obtained by Rule R from \( A_A = B_1, C_B = C_B \).

5.3. By a formal proof we mean a finite column of formulas each of which is either an axiom or else is obtained by Rule R from two earlier formulas of the column. By a formal theorem we mean a formula which is the last line of some formal proof. We put \( \vdash A_A \) if and only if \( A_A \) is a formal theorem.

5.4. Without altering the class of formal theorems, we may replace the above list of axioms and Rule R by a longer list of axioms and rules having a somewhat simpler character. These possibilities are described below without proof of their equivalence.

5.4.1. Axiom 5 may be replaced by:

AXIOM 5.1. \( (x_A = y_A) \rightarrow (f_A X_A = f_A X_A) \).

AXIOM 5.2. \( (f_A = g_A) \rightarrow (f_A X_A = g_A X_A) \).

5.4.2. Axiom Schema 7 may be replaced by:

AXIOM SCHEMA 7.1. \( (\lambda X_B A_B) A_A = A_A \).

AXIOM SCHEMA 7.2. \( (\lambda X_B Y_A) A_A = Y_A \) if \( X_A \neq Y_A \).

§ 6. To justify consideration of the system of axioms and rules of § 5 we wish to show that every formal theorem is valid. By § 3 a simple inductive argument reduces the problem to that of showing that every axiom is valid, and that Rule R preserves validity.

6.1. With the aid of 3.1, 4.4, 4.5, and 4.6 it is a trivial matter to verify the validity of all axioms falling under headings 1 through 6. In particular it will be observed that: Axiom Schema 1 expresses the most basic law of equality; Axiom Schema 2 is a simple identity involving the biconditional operation on truth values; Axiom 3 is an entry from the usual table of values for \( \land \); Axiom 4 is a way of expressing that \( D_B \) contains the elements \( T, F \), and no others; Axiom 5 expresses the substitutivity property of the identity relation; and Axiom 6 states the principle of extensionality.

Axiom Schema 7 gives the fundamental property of the functional abstractor, \( \lambda \). Because of the relative complexity of its formulation, verification of the validity of its axioms by 3.1 is not so simple as in the preceding cases. The simplest way to proceed is to show (by induction on the length of \( B_1 \)) that any instance of Axiom Schema 7 can be obtained by a succession of applications of Rule R to instances of Schemata 7.1-7.5 in 5.4.2 above. The validity of each instance of these schemata is a simple matter to establish by 3.1 and 3.2.

6.2. The preservation of validity by Rule R is most easily established by first showing that any application of Rule R can be effected by a succession of applications of Rules R1-R4. (This is shown by induction on the length of \( C_B \) in the application of Rule R.) That Rules R1-R4 preserve validity may be shown directly from 3.1 and 3.2. These rules (and indeed Rule R itself) express a form of the well-known substitutivity principle for the identity relation. Despite the fact that Rule R and Axiom 5 largely overlap in their intuitive meanings, neither one seems to be dispensable in our deductive system.
§ 7. We now show how the usual theorems and rules involving propositional connectives and quantifiers may be derived within our deductive system. As in the preceding section, $A_1$, $B_1$, and $C_1$ (resp., $X_1$, $X_2$, $Z_1$) are understood to be arbitrary formulas (resp., variables) of indicated type.

7.1. Rule of Biconditional (Rule B): If $\vdash A_1$ and $\vdash A_2 = B_2$, then $\vdash B_1$. This is immediate by Rule B (5.2) and the definition of $\vdash$ (5.3).

7.2. Equivalence Rules (E-Rules): (i) If $\vdash A_1 = B_1$, then $\vdash A_1 = A_2$, and (ii) If $\vdash A_1 = B_1$ and $\vdash B_1 = C_1$, then $\vdash A_1 = C_1$. These are derived in a familiar way by Axiom 1 (5.1.1) and Rule R.

7.3. Rule T: $\vdash A_1$ if and only if $\vdash A_1 = T$. The proof is by Axiom 2 (5.1.2), Rule B, and the E-rules.

7.4. Rule of Generalization (Rule G): If $\vdash A_1$ then $\vdash (\forall X_1 A_1)$.

Proof.
1. Suppose $\vdash A_1$.
2. $\vdash A_1 = T$; by T-Rule (7.3).
3. $\vdash (\forall X_1 A_1) = (\forall X_1 A_1)$; Axiom 1.
4. $\vdash (\forall X_1 A_1) = (\forall X_1 T)$; by Rule R applied to lines 2 and 3.
5. $\vdash (\forall X_1 A_1)$; by line 4 and the definition of $\forall$ (4.6).

7.5. Rule of Specialization (Rule S): If $\vdash (\forall X_1 A_1)$ then $\vdash C_1$, where $C_1$ results from substituting some formula $A_1$ for all free occurrences of $X_1$ in $A_1$, providing no such occurrence of $X_1$ is in a part of $B_1$ which is a formula beginning with the symbols $(\forall X_1)$, where $X_1$ is a variable occurring free in $A_1$.

Proof.
1. Suppose that $B_1$, $X_1$, $A_1$, and $C_1$ are related as above, and that $\vdash (\forall X_1 B_1)$.
2. $\vdash (\exists X_1 B_1) = (\forall X_1 T)$; by line 1 and definition of $\forall$ (4.6).
3. $\vdash (\exists X_1 B_1) = (\forall X_1 A_1) = (\forall X_1 A_1)$; Axiom 1.
4. $\vdash (\exists X_1 B_1) = (\forall X_1 T)$; by Rule R from lines 2 and 3.
5. $\vdash (\exists X_1 T)$; by Axiom 7 and definition of $\exists$ (4.2).
6. $\vdash (\exists X_1 B_1) = (\forall X_1 T)$; by E-Rules from lines 4 and 5.
7. $\vdash (\exists X_1 B_1) = (\forall X_1 T)$; by T-Rule from line 6.
8. $\vdash (\exists X_1 B_1) = (\forall X_1 = C_1)$; by line 1 and Axiom 7.
9. $\vdash C_1$; by Rule B (7.1) from lines 7 and 8.

7.6. Rule of Substitution for Free Variables (Rule Sub): If $B_1$, $X_1$, $A_1$, and $C_1$ are related as in the hypothesis of Rule S (7.5), and if $\vdash B_1$, then $\vdash C_1$. This is proved by combining Rule G with Rule S.

7.7. Theorem. $\vdash (T \wedge T) = T$.

Proof.
1. $\vdash (T \wedge T) = (T \wedge T)$; by definitions of $T$ (4.2) and $\wedge$ (4.4), Axioms 1 and 7 and Rule R.
2. $\vdash (T \wedge T) = (T \wedge T)$; Axiom 2.
3. $\vdash (T \wedge T) = (T \wedge T)$; by E-Rules (7.2) from lines 1 and 2.
4. $\vdash (T \wedge T) = (T \wedge T)$; by Axiom 1 and Rule T.
5. $\vdash (T \wedge T) = T$; by E-Rules from lines 3 and 4.

7.8. Rule of Conjunction (Rule C): If $\vdash A_1$ and $\vdash B_1$ then $\vdash (A_1 \wedge B_1)$.

Proof.
1. Suppose $\vdash A_1$ and $\vdash B_1$.
2. $\vdash A_1 = T$ and $\vdash B_1 = T$; by Rule T from line 1.
3. $\vdash (A_1 \wedge B_1) = (A_1 \wedge B_1)$; Axiom 1.
4. $\vdash (A_1 \wedge B_1) = (T \wedge T)$; by Rule R from lines 2 and 3.
5. $\vdash (A_1 \wedge B_1) = (T \wedge T)$; by E-Rules from line 4 and Theorem 7.7.
6. $\vdash (A_1 \wedge B_1) = (T \wedge T)$; by Rule T from line 5.

7.9. Rule of (Propositional) Cases: If $A_1$ is any formula of type 0, if $X_1$ is any variable of type 0, if $A_1$ and $A_1'$ are obtained from $A_1$ by substituting $T$ and $F$ respectively for all free occurrences of $X_1$ in $A_1$, and if $\vdash A_1$ and $\vdash A_1'$, then also $\vdash A_1$.

Proof.
1. Suppose that $A_1$, $X_1$, $A_1'$, and $A_1''$ are related as above, and that $\vdash A_1$ and $\vdash A_1'$.
2. $\vdash (\forall X_1 A_1) = A_1' = A_1''$; by Axiom 7 and line 1, then Rule R.
3. $\vdash ((\exists X_1 A_1) \wedge (\exists X_1 A_1'')) = (\forall X_1 A_1)$; by Rule C (7.8) and line 1, then Rule R and line 2.
4. $\vdash ((\exists X_1 A_1) \wedge (\exists X_1 A_1'')) = (\forall X_1 A_1)$; by Rule Sub (7.6) from Axiom Schema 4.
5. $\vdash (\exists X_1 A_1'')) = X_1$; by Rule M (7.1) from lines 4 and 3.
6. $\vdash (\exists X_1 A_1'')) = A_1''$; by Axiom 7.
7. $\vdash (\exists X_1 A_1'')) = R$; by Rule M from lines 6 and 5.
8. $\vdash A_1''$; by Rule S (7.5) from line 7.

7.10. Theorem. $\vdash (T \wedge T) = T$.

Proof. By Rule of Cases (7.9) from Theorem 7.7 and Axiom 3.

7.11. Theorem Schema: $\vdash (T \rightarrow R_1) = R_1$.

Proof.
1. $\vdash (T \rightarrow R_1) = (T \rightarrow R_1)$; Axiom 1.
2. \( \vdash (T'' \to R_b) = (\lambda x_a (\lambda y_b (x_a \land y_b) = x_a)) \) \( T'' R_b \); from line 1 by definition of \( \to (4.5) \).
3. \( \vdash (T'' \to R_b) = (T'' \land R_b) = T'' \); by E-Rules from line 2 and Axiom Schema 7.
4. \( \vdash (T'' \to R_b) = R_b \); by Rule T and E-Rules from line 3 and Theorem 7.10.

7.12. RULE OF MODUS PONENS (RULE MP). If \( \vdash A_a \) and \( \vdash (A_a \to R_b) \), then \( \vdash R_b \).

Proof.
1. Suppose \( \vdash A_a \) and \( \vdash (A_a \to R_b) \).
2. \( \vdash (A_a = T_a) \); by Rule T from line 1.
3. \( \vdash (T'' \to R_b) \); by Rule R from lines 2 and 1.
4. \( \vdash (T'' \to R_b) = R_b \); Theorem Schema 7.11.
5. \( \vdash R_b \); by Rule B from lines 4 and 3.

7.13. THEOREM SCHEMA. \( \vdash (F'' \to A_a) \).

Proof.
1. \( \vdash (A_a = A_a) \to ((\lambda x_a x_a) = (\lambda x_a x_a)) \to ((\lambda x_a x_a) A_a) = ((\lambda x_a T_a) A_a) \); by Rule Sub from Axiom 5.
2. \( \vdash ((\lambda x_a x_a) = (\lambda x_a T_a)) \to ((\lambda x_a x_a) A_a) = ((\lambda x_a T_a) A_a) \); by Rule MP (7.11) from Axiom 1 and line 1.
3. \( \vdash F'' \to (A_a = T_a) \); from line 2 by definition of \( F'' (4.2) \), Axiom 7, and Rule R.
4. \( \vdash F'' \to A_a \); by Axiom 2 and Rule R from line 3.

7.14. THEOREM. \( \vdash (T'' \to T''') = T'', \vdash (T'' \to F'') = F'', \vdash (T'' \to T''') = T'' \), and \( \vdash (F'' \to F''') = F'' \).

Proof. By Theorems 7.11 and 7.13 and Rule T.

7.15. THEOREM. \( \vdash (\gamma T'') = F'' \) and \( \vdash (\gamma F'') = T'' \); also \( \vdash (T'' \lor F'') = T'' \), \( \vdash (T'' \lor F'') = T'' \), \( \vdash (F'' \lor F'') = F'' \). 

Proof. Using the definitions of \( \gamma (4.3) \) and of \( \lor (4.5) \), these results follow easily by Axioms 1 and 7, Rules R and T, and Theorem 7.14.

7.16. THEOREM. \( \vdash (F'' \lor T'') = F'' \), \( \vdash (F'' \lor F'') = F'' \), \( \vdash (T'' \lor F'') = T'' \), and \( \vdash (T'' \lor F'') = F'' \).

Proof. We have \( \vdash (F'' \lor T'') = (F'' \lor T'') = F'' \); by Axioms 1 and 7 and definition of \( \lor (4.5) \). Hence \( \vdash (F'' \lor T'') = F'' \) by Theorem 7.14, E-Rules, and Rule T. Similarly \( \vdash (F'' \lor F'') = (F'' \lor F'') = F'' \), and hence \( \vdash (F'' \lor F'') = F'' \). The remaining results of the theorem come from 7.16.

7.17. THEOREM. \( \vdash (T'' = T'') = T'' \), \( \vdash (F'' = T'') = F'' \), \( \vdash (F'' = F'') = T'' \), and \( \vdash (T'' = F'') = F'' \).
By induction hypothesis, therefore, we obtain \( \vdash A' \) and \( \vdash A'' \). But then \( \vdash A' \) by the Rule of Cases (7.9).

7.20. Theorem. If \( A \) is a tautology and \( B_i \) is obtained from \( A \) by simultaneous substitution of arbitrary formulas of type \( 0 \) for the free variables of \( A \), then \( B_i \) is called a tautological formula. Clearly by Rule Sub and 7.19 we have \( \vdash B_i \) for each such \( B_i \).

Having shown how the ordinary theorems of propositional logic are included among ours we turn to theorems of predicate logic (with equality).

7.21. Theorem on Change of Bound Variables. Suppose that \( A_i \) and \( B_i \) are formulas, and \( X_i \) and \( Y_j \) are variables, such that \( B_i \) is obtained by replacing all free occurrences of \( X_i \) in \( A_i \), by \( Y_j \), and \( A_i \) is obtained by replacing all free occurrences of \( Y_j \) in \( B_i \), by \( X_i \). Then \( \vdash (\lambda X_i A_i \lambda Y_j B_i) \).

Proof. By Axiom 7 and our hypothesis we have \( \vdash (\lambda X_i A_i \lambda Y_j B_i) = A_i \) and \( \vdash (\lambda X_i A_i \lambda Y_j B_i) = A_i \), so that \( \vdash \forall X_i, (\lambda X_i A_i \lambda Y_j B_i) = (\lambda X_i A_i \lambda Y_j B_i) \) by E-Rules and Rule G (7.4). We then obtain \( \vdash (\lambda X_i A_i \lambda Y_j B_i) = (\lambda X_i A_i \lambda Y_j B_i) \) by applying Rule Sub to Axiom 6 and using Rule MP.

7.22. Suppose that \( C_i \) results from substituting some formula \( A_i \) for all free occurrences of \( X_i \) in \( B_i \), and that no such occurrence of \( X_i \) is in a part of \( B_i \) which is a formula beginning with the symbols \( (\lambda X_i \lambda Y_j) \), where \( Y_j \) is a variable occurring free in \( A_i \). Then \( \vdash (\lambda X_i A_i) \) by Rule Sub applied to Axiom 5, followed by Rule MP with Axiom 1.

From this we obtain \( \vdash (\lambda X_i A_i) \) by definition of \( \forall \) (4.6), Axiom 7, E-Rules, and Rule B.

7.23. Theorem. If the variable \( X_i \) does not occur free in \( B_i \), then \( \vdash (\lambda X_i A_i) \) by Rule Sub applied to Axiom 5, followed by Rule MP with Axiom 1.

From this we obtain \( \vdash (\lambda X_i A_i) \) by definition of \( \forall \) (4.6), Axiom 7, E-Rules, and Rule B.

7.24. Theorem. Suppose that \( B_i \) is obtained from \( A_i \) by replacing one free occurrence of \( X_i \) by a free occurrence of \( Y_j \). Then \( \vdash (\lambda X_i A_i \lambda Y_j B_i) \).

The proof is by induction on the length of \( A_i \). If \( A_i \) is a variable (namely, \( X_i \)), the formula involved is tautological. If \( A_i \) has the form \( (\lambda X_i A''_i) \), we carry through the induction step with the aid of Axiom 5 and Rules Sub and MP. Thus it remains only to consider the case where \( A_i = (\lambda X_i C_i) \), with \( X_i \) free in \( C_i \). From the induction hypothesis

\[
\vdash (\lambda X_i Y_j) \Rightarrow (C_i \Rightarrow (\lambda X_i C_i))
\]

where \( C_i \) arises from \( C_i \) by replacing one free occurrence of \( X_i \) with a free occurrence of \( Y_j \), and we then obtain

\[
\vdash (\lambda X_i Y_j) \Rightarrow (\lambda X_i C_i) \Rightarrow (\lambda X_i C_i)
\]

by Rule G and Theorem 7.23. But

\[
\vdash (\lambda X_i C_i) \Rightarrow (\lambda X_i C_i)
\]

is obtained by Rule Sub and a suitable tautological formula, we now combine the above results to obtain

\[
\vdash (\lambda X_i Y_j) \Rightarrow (\lambda X_i C_i) \Rightarrow (\lambda X_i C_i)
\]

which completes our inductive proof.

7.25. Theorems 7.20-7.24, together with Rules G and MP, show that all of the usual formal theorems involving propositional connectives, quantifiers, and the equality sign are available in our system (1). In the sequel we shall make free use of such theorems with the simple reference "by predicate logic".

7.26. By induction we define for our system a relation of formal consequence which holds between a finite set \( \Gamma \) of formulas of type \( 0 \), and a single formula \( A \), as follows. If \( \Gamma \) is empty then \( \Gamma \vdash A \) if and only if \( \vdash A \). If \( \Gamma \) is non-empty then \( \Gamma \vdash (\forall X_i A_i) \) if and only if, for every \( B_i \in \Gamma \), we have \( \Gamma \vdash (B_i \Rightarrow (B_i \vdash A_i)) \), where \( B_i \) is the set obtained from \( \Gamma \) by removing the element \( B_i \). We list below some basic properties of this relation, all of which are easily proved by induction on the size of \( \Gamma \) (with the aid of the observation of 7.25).

(i) If \( \Gamma \vdash A \) and \( \Gamma \subseteq A \) then \( A \vdash A \).

(ii) If \( \Gamma \vdash A \) and \( \Gamma \vdash (A_i \Rightarrow B_i) \) then \( \Gamma \vdash B_i \).

(iii) If \( \Gamma \vdash A_i \) and \( X_i \) is not free in any formula of \( \Gamma \) then \( \Gamma \vdash (\forall X_i A_i) \).

(iv) If \( \Gamma \vdash (A_i = B_i) \) and \( \Gamma \vdash C_i, D_i \) is obtained by replacing one occurrence of \( A_i \) in \( C_i \) by an occurrence of \( B_i \), and if no part of \( C_i \) contains this occurrence of \( A_i \) in a formula beginning with the symbols \( (\lambda X_i \lambda Y_j) \), where \( Y_j \) is a variable free in \( (A_i = B_i) \) and in some formula of \( \Gamma \), then \( \Gamma \vdash D_i \).

We can paraphrase the content of (ii)-(iv) by saying that Rule MP, Rule G, and Rule B are valid for deductions from a set \( \Gamma \), providing certain restrictions on the binding of variables are observed. The property of the consequence relation that whenever \( \Gamma \vdash A \) we have also \( \Gamma \vdash B \), which is immediate from our definition of this relation, is known in the literature as the Deduction Theorem.

§ 8. In this section we develop our completeness result for the theory of propositional types by generalizing our method of proof of Theorem 7.19.

---

(1) Compare Sections 30 and 48 of [2].
8.1. Consider the following four closed formulas of type (00):
\[ E' = (\lambda x_a (g_a E')), \quad E = (\lambda x_a (F = a)), \quad E' = (\lambda x_a (F a)), \quad \text{and} \quad E = (\lambda x_a (F')). \]
We first establish some simple formal theorems about these.

8.1.1. Theorem. We have \[ \vdash (E' E') = E', \quad \vdash (E F) = F, \quad \vdash (E' F) = F, \quad \text{and} \quad \vdash (E' E') = F. \]
More precisely, for each \( i = 1, \ldots, 4 \) and each \( x \in \mathcal{D}_a \) we have \[ \vdash (E' x) = (f x) \]
where \( f, \ldots, f' \) are the elements of \( \mathcal{D}_a \) specified in 4.4.

Each part of this theorem is established with the aid of Axiom 7.

In the case of \( E_a \) an additional step is needed, using Axiom 1 or Axiom 2.

8.1.2. Theorem. For \( i \neq j \) we have \[ \vdash (E' = E') \]
for any formula \( E \).

To indicate the proof for one part of this theorem, take the case
\( i = 1, j = 4 \). From 8.1.1 we have \[ \vdash (E' E') = E' \] and \[ \vdash (E' E) = E' \]. Hence by predicate logic (in fact, by propositional logic), we get \[ \vdash (E' = E') \]
and the desired result, \[ \vdash (E' E') = E' \], now follows by predicate logic from Axiom 6.

8.1.3. Theorem. \[ \vdash \forall g_a E = E' \] or \( \forall g_a E = F' \) or \( \forall g_a E = E \).

8.2. Theorem. We turn now to the formulas \( \omega_{ab} \), introduced in 4.9, and shall show by induction on a \( \vdash \)

\[ \vdash \forall g_a \omega_{ab} (\exists x_a (g_a x_a) \to (g_a \omega_{ab} g_a x_a)) \]

We remark that once we have established this theorem for some \( a \) we may employ predicate logic, the definition (4.10) of \( (\lambda X_a A)_a \), and Axiom 7 to obtain

\[ \vdash \exists 1 (\lambda X_a A_a) \to (\lambda X_a A_a) \]

for any formula \( A_a \).

8.2.1. Since \( \omega_{ab} = (\lambda x_a (x_a = b)) \), we see by definition (8.1) of \( E^* \) and Axiom 7 that for \( i = 1, 2 \) we have \[ \vdash (\lambda x_a E^*) = (E^*) = E \]. Using 8.1.2 we then obtain \[ \vdash (\omega_{ab} E^*) = E^* \; \text{and} \; \vdash (\omega_{ab} E^*) = F^* \].

We can then infer from Theorems 8.1.4, 8.1.3, and 8.1.1 again, by predicate logic, that \[ \vdash \forall g_a (\exists x_a (g_a x_a) \to (g_a \omega_{ab} g_a x_a)) \], as required.

8.2.2. We now make the induction hypothesis that

\[ \vdash \forall g_a (\exists x_a (g_a x_a) \to (g_a \omega_{ab} g_a x_a)) \]

so that, by the remark following 8.2(1), we have

\[ \vdash (\lambda X_a A_a) \to (\lambda X_a (\lambda X_a (A_a))) \]

for any formula \( A_a \). And we seek to prove that

\[ \vdash \forall g_a (\exists x_a (g_a x_a) \to (g_a \omega_{ab} g_a x_a)) \]

From the definition (4.9) of \( \omega_{ab}(g_a x_a) \) and Axiom 7 we obtain

\[ \vdash (\lambda x_a (\lambda x_a (g_a x_a) \to (g_a \omega_{ab} g_a x_a))) \]

Hence, by predicate logic,

\[ \vdash (\lambda X_a (g_a x_a) \to (\lambda x_a (\lambda x_a (g_a x_a) \to (g_a \omega_{ab} g_a x_a)) \]

However, again by predicate logic,

\[ \vdash (\lambda X_a (g_a x_a) \to (\lambda x_a (\lambda x_a (g_a x_a) \to (g_a \omega_{ab} g_a x_a)))) \]

From this, using (II) above, and Axiom 7, we get

\[ \vdash (\lambda X_a (g_a x_a) \to (\lambda x_a (\lambda x_a (g_a x_a) \to (g_a \omega_{ab} g_a x_a)))) \]

By predicate logic, Axiom 6, and Axiom 7 this, in turn, leads to

\[ \vdash (\lambda X_a (g_a x_a) \to (\lambda x_a (\lambda x_a (g_a x_a) \to (g_a \omega_{ab} g_a x_a)))) \]
When this is combined with (IV), we get
\[ \neg[(\exists x_0 \theta)(\exists x_0 \theta)] \rightarrow \forall z_0 \theta \rightarrow \forall z_0 \theta \rightarrow \forall z_0 \theta = \forall z_0 \theta \]
from which the desired (III)
\[ \neg \forall \theta \exists x_0 \theta \rightarrow [(\exists x_0 \theta)(\exists x_0 \theta)] \rightarrow \forall \theta \exists x_0 \theta \]
follows by elementary predicate logic.

This completes our inductive proof of Theorem 8.2.

8.3. Using the result of the preceding section that
\[ \neg[(\exists x_0 \theta)(\exists x_0 \theta)] \rightarrow [(\exists x \theta)(\exists x \theta)] \]
we turn now to some formal theorems involving the formulas \( \forall \theta \) (for every element \( \theta \) of any propositional type \( \Theta \)) which were defined in 4.10. In fact, if \( \Theta \) is any propositional type and \( x_1, ..., x_p \) are distinct and include all elements of \( \Theta \), we shall show (by induction on \( \gamma \)) that:

1. \[ \neg \gamma (x_1 = x_2) \text{ if } i \neq j, \]
2. \[ \forall \theta (x_1 = x_2) \vee \forall \theta (x_1 = x_2) \]
3. \[ \text{if } \gamma (a_0) \text{ then for any } \gamma \in \Theta \text{ we have } \neg (\forall \theta (x_1 = x_2)). \]

For the case where \( \gamma = 0 \) we have \( p = 2 \) so there is just one formula (1) and this is a tautology; the formula (2) is obtained by Rule G
From another tautology; and (3) holds vacuously.

Proceeding by induction, let us now assume that \( \gamma (a_0) \), and that \( f_1, ..., f_p \) is a list of the distinct elements of \( \Theta \), while \( x_1, ..., x_p \) is a list of the distinct elements of \( \Theta \).

We first give a proof of (3). From the definition (4.10) of \( f_1 \) and Axiom 7 we have
\[ \neg (\forall (x_1 = x_2) = (a_0, (x_1 = x_2)) \wedge \neg \gamma (a_0). \]
Then, using part (1) of our induction hypothesis concerning \( \Theta \), we have
\[ \neg (\forall (x_1 = x_2)) \text{ for } j \neq k, \]
so we obtain
\[ \neg (\forall (x_1 = x_2)) = (a_0, (x_1 = x_2)), \]
by predicate logic we know
\[ \neg (\forall (x_1 = x_2)) = (a_0, (x_1 = x_2)), \]
which in turn yields
\[ \neg (\forall (x_1 = x_2)) = (a_0, (x_1 = x_2)), \]
by Axiom 7. Combining this with (4) above we get the desired
\[ \neg (\forall (x_1 = x_2)) = (a_0, (x_1 = x_2)). \]
\( \vdash A_4 \) must also hold by considering the closure \( \forall X_1, \ldots, X_n. A_4 \), where \( X_1, \ldots, X_n \) are all of the variables occurring freely in \( A_4 \). Thus the completeness proof is completed.

The lemma is proved by induction on the length of \( A_4 \). Indeed, if \( A_4 \) is a variable then \( A_4^{00} = (\forall A_4) \), so \( A_4^{00} = (\forall (A_4)) \) is an instance of Axiom 1.

Next consider the case where \( A_4 \) is a \( Q_{\alpha_1} \), and suppose that \( y_1, \ldots, y_q \) are all of the elements of \( D_\alpha \). For \( 1 \leq i, j \leq q \), we have \( (Q_{\alpha_1} y_i y_j) = F^a \) if \( i \neq j \), by 8.3(1) and definition of \( = \) and \( \vdash \), and we have \( (Q_{\alpha_1} y_i y_i) = F^a \) by Axioms 1 and 2 and definition of \( = \). If \( i \neq j \) and \( (Q_{\alpha_1} y_i y_j) = F^a \) by 8.3(3). Thus we obtain, for each \( i = 1, \ldots, q \),

\[ \vdash (Q_{\alpha_1} y_i y_i) = (f^a y_i y_i) \]

and then

\[ \vdash \forall x, (Q_{\alpha_1} y_i y_i) = (f^a y_i y_i) \]

by using 8.3(2), and finally \( (Q_{\alpha_1} y_i y_i) = (y_i y_i) \) by Axiom 6. Since this holds for each \( i = 1, \ldots, q \), we repeat the same pattern of argument to arrive at the conclusion \( (Q_{\alpha_1}) = F^a \). Since \( Q_{\alpha_1} = Q_{\alpha_1} \) and \( (Q_{\alpha_1} y_i y_i) = f \), the lemma is seen to hold in this case.

Turning to the case where \( A_4 \) has the form \( (R_{\beta_1} C_{\beta_1}) \), we make the induction hypothesis that

\[ \vdash (R_{\beta_1} C_{\beta_1}) = (\forall (R_{\beta_1}) \phi)^a \]

and

\[ \vdash C_{\beta_1} = (\forall (C_{\beta_1} \phi))^a \]

Using 8.3(3) we then conclude that

\[ \vdash (R_{\beta_1} C_{\beta_1}) = (\forall (R_{\beta_1} \phi))^{a a} \]

But by definition of \( A_4^{00} \) we have \( (R_{\beta_1} C_{\beta_1}) = (R_{\beta_1} C_{\beta_1})^{00} \), and by definition of \( \phi \) we have \( (R_{\beta_1} \phi) = (R_{\beta_1} \phi) \), so that \( (R_{\beta_1} C_{\beta_1})^{00} = (\forall (R_{\beta_1} \phi))^a \)

as needed to establish this lemma in the case.

Finally we take up proof of the lemma for the case where \( A_4 \) has the form \( (\lambda x, C_{\beta_1}) \), so that \( x = (f y) \). Let us suppose that \( y_1, \ldots, y_q \) are all of the elements of \( D_\beta \). By induction hypothesis we have \( \vdash C_{\beta_1} = (\forall (C_{\beta_1} \phi))^a \)

for every assignment \( \phi \).

Now given any assignment \( \phi \) and any \( y_i \in D_\beta \), we known from Axiom 7 and the definition of \( (\lambda x, C_{\beta_1})^{00} \) that \( (\lambda x, C_{\beta_1})^{00} = C_{\beta_1}^{00} \), where \( \phi(x_1) = y_1 \) and \( \phi(x_2) = \phi(x_2) \) for every variable \( x_2 \neq x_1 \). But \( C_{\beta_1} = C_{\beta_1} \), by definition of \( \phi \) and the fact (Section 4) that \( (y_1 y_1) = (y_1^a y_1) = y_1 \). Hence our induction hypothesis yields

\[ \vdash (\lambda x, C_{\beta_1})^{00} = (\forall (\lambda x, C_{\beta_1} \phi))^a \]

Using 8.3(3) we therefore get

\[ \vdash (\lambda x, C_{\beta_1})^{00} = (\forall (\lambda x, C_{\beta_1} \phi))^a \]

Since this is true for each \( i = 1, \ldots, q \) we can use 8.3(2) and Axiom 6 to obtain

\[ \vdash (\lambda x, C_{\beta_1}) = (\forall (\lambda x, C_{\beta_1} \phi))^a \]

which completes the inductive proof of our lemma.

**Remark.** It is easy to see that while our completeness proof (§8) depends in an essential way on the restriction of our system to the class of propositional types, \( \mathcal{F}_C \), the essential features of our semantical and syntactical development of propositional and predicate logic (§4—§7) can be carried through for a system of the full theory of types based on the same primitive notions. To obtain such a system, following Church [1], we deal with a class \( \mathcal{F}_C \) of finite types obtained by starting from \( D_0 \) and from an arbitrary set \( D_1 \) (whose elements are called individuoses), and generating all further types obtained by passing from any \( D_0 \) and \( D_1 \) to \( D_2 \).

Now consider a symbol \( \beta \) obtained from our present symbol (§2) simply by adding variables \( x_4 \) and constants \( Q_{\beta_1} \) of the new types to those of the old. The definitions of propositional connectives and quantifiers (4.1—4.7) obviously remain valid for this symbol \( \beta \). Furthermore, if we adopt the same axiom system (§5), it is clear that the derivation of the basis rules of predicate logic, consisting of sections 7.20—7.24, 5.1.1, 7.4, and 7.12, will all continue to hold in the new system.

However, the system obtained in this way does not seem to be a really adequate formulation of type-theoretic predicate logic, since it does not seem possible to prove such a formula as

\[ (\forall x_1 = u_1)^{a a} \]

To remedy this defect it is necessary to add to the system \( \beta \) a new primitive constant \( (\forall x_1 = u_1)^{a a} \) to enable us to extend 4.9 by introducing description-formulas \( \zeta_{a n} \) for all \( a \in \mathcal{F}_C \). We must then add to our axiom system an

**Axiom 8.**

\[ (\forall x_1 = u_1)^{a a} \rightarrow (\forall x_1 = u_1)^{a a} \]

which will enable us to extend Theorem 8.3 to the new system. With the help of this it can be easily prove such formulas as (**).
be left, presents three formulations of the theory of propositional types, one of which is based upon equivalence. A proof of completeness is given. However, the systems differ from ours in various ways, principally in a rule of definition allowing the introduction of names for arbitrary elements of the hierarchy of propositional types.

References


Reçu par la Rédaction le 2. 7. 1952

---

A reduction of the axioms for the theory of propositional types

by

P. Andrews* (Princeton, N.J.)

Throughout this paper we shall follow the notation used by Henkin in his paper *A theory of propositional types* (this Volume, pp. 323-342), hereafter referred to as [H]. Reference numbers followed by 'H' refer to sections of that paper. (1)

Henkin's paper is of particular interest in that it takes symbols for the identity relation as the sole primitive constants. That there is ample historical precedent for special interest in such a system is attested by the following passage from Ramsey's article, *The Foundations of Mathematics*:

"The preceding and other considerations led Wittgenstein to the view that mathematics does not consist of tautologies, but of what he called 'equations', for which I should prefer to substitute 'identities'. (H) is interesting to see whether a theory of mathematics could not be constructed with identities for its foundation. I have spent a lot of time developing such a theory, and found that it was faced with what seemed to me insuperable difficulties." (6)

The full beauty of Henkin's theory of propositional types can perhaps best be appreciated when the system of axioms in section 5.1H is simplified somewhat. Therefore let us replace this system of axioms by the following

AXIOMS.

(1) \((\forall x \exists y ((T \land \exists x \exists y)) \Rightarrow \forall x \exists y (T \land \exists x \exists y)).\)

(2) \((a = a) \Rightarrow (\exists x (a = x)).\)

* The author was supported by a National Science Foundation Cooperative Graduate Fellowship while working on this paper.

(1) I wish to express my appreciation to Professor Henkin for the privilege of reading his paper before publication, and for his many helpful suggestions, comments, and criticisms during the writing of the present paper.