9/8/81

The proof of the Church-Rosser theorem is on pp. 8–11.

Best regards,

Per M.-L.

A THEORY OF TYPES

Per Martin-Löf

February 1971, revised October 1971

After the recent proof theoretical investigations of simple type theory, it would seem natural to make an attempt at set theory as formalized by Zermelo and Fraenkel. However, mainly because of the form of the replacement axiom, it does not seem as if set theory lends itself very well to a proof theoretical analysis.

Instead, I have formulated an intuitionistic theory of types in which (the intuitionistic and intentional version of) the simple theory of finite types is imbedded as a subsystem. The imbedding is such that the Hauptsatz for simple type theory proved by Girard 1970 and Martin-Löf 1970 follows combinatorially from the Hauptsatz for the full theory which constitutes the principal technical result of this paper. I do not yet know how the proof theoretic strength of the present theory compares with that of Zermelo and Fraenkel's set theory.

In a forthcoming paper I plan to show that, unlike set theory, the present theory of types is adequate for a straightforward formalization of category theory. This is so because the basic axiom that there is a type of types introduces precisely that kind of selfreference which is needed in order to construct, for instance, functor categories and the category of all categories.

1 INTUITIVE EXPLANATIONS OF THE BASIC CONCEPTS

1.1 In what follows mathematical objects will be regarded as our own constructions. Every mathematical object is of a certain kind or type which is uniquely associated with the object in question. A type is defined by prescribing how we are allowed to construct objects of that type. The types themselves are mathematical objects, namely, those objects whose type is the type of types. In other words, a type is the same as an object of the type of types. I shall denote the type of types by the symbol

V.

Note that V is itself a type, namely, the type of types, and hence an object of type V.

The idea of logical types is due to Russell 1903 and 1908. He defined a type as the range of significance of a propositional function. This notion of type is compatible with the one that will be formalized in the present paper, because the domain of a propositional function will always be a type and, conversely, every type will occur as the domain of a propositional function.

1.2 A proposition is defined by prescribing what we have to do in order to prove it. For example,

971 is a non prime number

is a proposition which we prove by exhibiting two natural numbers

greater than one and a computation which shows that their product equals 971. The similarity between the notion of proposition and the notion of type described above is not accidental. Indeed, a proposition may always be regarded as a type, namely, the type of proofs of that proposition, and, conversely, a type always determines a proposition, namely, the proposition which we prove by exhibiting an object of that type. This explains why I shall treat the notion of type and the notion of proposition as one and the same notion, thereby taking seriously the analogy between types (or categories in the terminology of Curry and Feys 1958) and propositions discovered by Curry and Feys 1958 in the case of the positive implicational calculus and extended to Heyting arithmetic by Howard 1969.

1.3 The relation which an object bears to its type, that is, the relation

a is an object of type A

or, if we think of A as a proposition rather than a type,

a is a proof of the proposition A

will be denoted by

$$a \in A.$$

For example,

$$V \in V$$

expresses that V is an object of type V. In intuitionistic mathematics the relation $a \in A$ is more fundamental than the assertion relation

$\vdash A$

which merely expresses that A holds or is true, that is, that there is a proof a of the proposition A, because half, and, usually, the more valuable half, of the information contained in $a \in A$ is lost when passing to $\vdash A$. I think that $a \in A$ may be regarded as the general form of the theorems of intuitionistic mathematics.

The relation $a \in A$ is decidable. Indeed, a type is well-defined only if we have prescribed how we are allowed to construct objects of that type, and this is to mean that we should be able to check whether or not an arbitrarily given object meets the prescription. In the case when A is thought of as a proposition, Kreisel 1960 has argued for the decidability of the proof relation a A. We recognize a proof of a proposition when we see one, as he says. That is, it is not a proof unless we recognize it as such. On the other hand, it was made clear by Brouwer that there is no reason for us to believe that the assertion relation is decidable, because that belief could only be justified if we thought it would be possible to construct a method which, when applied to an arbitrary mathematical proposition, would yield either a proof or a disproof of that proposition.

1.4 Suppose that we have defined a function, rule or method which associates with every object x of type A a certain type B(x). Then we allow ourselves to form the cartesian product

$$(\prod x \in A)B(x)$$

of the types B(x) when x ranges over A. Alternatively, if we think of B(x) as a proposition for every object x of type A, then $(\prod x \in A)B(x)$ is the logical product or conjunction of the

propositions $B(x)$ obtained by letting x range over A. The cartesian product $(\Pi x \in A)B(x)$ is to be a type and, therefore, I have to prescribe how we are allowed to construct objects of that type. Let b be a function which associates with every object x of type A an object $b(x)$ of type $B(x)$. Then this function b is an object of type $(\Pi x \in A)B(x)$. In symbols,

$$b \in (\Pi x \in A)B(x).$$

Alternatively, if we think of $b(x)$ as a proof

$$\begin{array}{c} \vdots \\ B(x) \end{array}$$

of the proposition $B(x)$ for every object x of type A, then, by joining these proofs together, which is usually indicated by a figure of the form

$$\frac{\begin{array}{c} \vdots \\ B(x) \text{ for all } x \in A \end{array}}{(\Pi x \in A)B(x)}$$

we get the proof b of the universal proposition $(\Pi x \in A)B(x)$.

In the special case when $B(x)$ is defined to be one and the same type B for every object x of type A, $(\Pi x \in A)B(x)$ will be abbreviated

$$A \to B.$$

It is the type of functions whose arguments are of type A and whose values are of type B. Thinking of A and B as propositions, it is the proposition

A implies B.

A proof of the implication $A \rightarrow B$ is a function (rule or method) which takes an arbitrarily given proof of A into a proof of B. Note that this is precisely the intuitionistic explanation of the notion of implication.

1.5 If b is an object of type $(\Pi x \in A)B(x)$, that is, a function which takes an arbitrary object x of type A into an object of type B(x), then we can apply b to an object a of type A. The result

$$b(a)$$

of this <u>application</u> operation is an object of type B(a).

1.6 If two linguistic expressions a and b both denote the same object, we shall say that they are <u>definitionally</u> or <u>intentionally</u> <u>equal</u> and write

$$a = b.$$

Other notions of equality, such as equality (in the usual sense) between real numbers and extensional equality between species of objects of some given type, have to be defined. As for the distinction between intension and extension, see Whitehead and Russell 1910 and Church 1941.

2 FORMALIZATION OF A THEORY OF TYPES

2.1 The formal symbols are

$$\Pi,\ \lambda,\ \epsilon,$$

the constant

$$V,$$

variables

$$x,\ y,\ z,\ \ldots\ ,\ X,\ Y,\ Z,\ \ldots$$

and parantheses

$$(,\ ).$$

The symbol λ stands for functional abstraction as in Church 1932.

2.2 From the formal symbols we build up certain (in general, meaningless) symbol strings to be called formal expressions. Every occurrence of a variable in an expression is either free or bound. I shall denote an expression b by b(x) when I want to indicate explicitly that it may contain some free occurrences of the variable x.

2.2.1 V is an expression.

2.2.2 A variable x is an expression, and the occurrence of x in this expression is free.

2.2.3 If A and B(x) are expressions and the variable x does not occur free in A, then $(\Pi x \epsilon A)B(x)$ is an expression in which every occurrence of x is bound. An occurrence of a variable

other than x in $(\Pi x \in A)B(x)$ is free or bound depending on whether it was free or bound in that one of the expressions A and B(x) in which it occurs.

2.2.4 If A and b(x) are expressions and the variable x does not occur free in A, then $(\lambda x \in A)b(x)$ is an expression. The definition of free and bound variables is the same as in the previous paragraph.

2.2.5 If a and b are expressions, then (ba) is an expression. An occurrence of a variable in (ba) is free or bound depending on whether it was free or bound in that one of the expressions a and b in which it occurs.

Following Schönfinkel 1924, I shall use

$$ba_1 \ldots a_n$$

as an abbreviation for

$$(\ldots(ba_1)\ldots a_n).$$

2.3 Expressions may be simplified by means of the <u>reduction rule</u>

$$(\lambda x \in A)b(x)a \geqslant b(a).$$

Here b(a) denotes the expression which is obtained by substituting the expression a for all free occurrences of the variable x in the expression b(x). Before the substitution can be carried out, however, some bound variables in b(x) may have to be renamed so that no variable which is free in a becomes bound in b(a). In the following I shall tacitly assume that bound variables are

renamed whenever necessary in order to avoid undesired ties. Also, expressions which only differ in the naming of their bound variables are identified.

In an application of the reduction rule the right member is said to be obtained from the left member by contraction. An expression a reduces to an expression b, abbreviated

$$a \geqslant b$$

as in Curry and Feys 1958, if b can be obtained by repeated contractions of parts of the expression a. An expression is irreducible or normal if it cannot be further reduced.

2.4 Two expressions a and b are definitionally or intentionally equal and we write

$$a = b$$

if they reduce to a common expression, that is, if there is an expression c such that $a \geqslant c$ and $b \geqslant c$. The relation = is an equivalence relation. The reflexivity and the symmetry are obvious from the definition, whereas the transitivity is a consequence of the following theorem first proved by Church and Rosser 1936 for the type free calculus of lambda conversion. The proof given below is an adaptation of a proof for the type free combinator calculus shown to me by William Tait.

2.5 Property of Church and Rosser. If $a \geqslant b$ and $a \geqslant c$, then there is an expression d such that $b \geqslant d$ and $c \geqslant d$.

Proof. We shall first prove the theorem for an auxiliary relation $\geqslant_1$.

2.5.1 Definition of the relation $\geqslant_1$.

2.5.1.1 $V \geqslant_1 V$.

2.5.1.2 If x is a variable, then $x \geq_1 x$.

2.5.1.3 If $A \geq_1 C$ and $B(x) \geq_1 D(x)$, then $(\Pi x \in A)B(x) \geq_1 (\Pi x \in C)D(x)$.

2.5.1.4 If $A \geq_1 C$ and $b(x) \geq_1 d(x)$, then $(\lambda x \in A)b(x) \geq_1 (\lambda x \in C)d(x)$.

2.5.1.5 If $a \geq_1 c$ and $b \geq_1 d$, then $ba \geq_1 dc$.

2.5.1.6 This is the crucial case. If $a \geq_1 c$ and $b(x) \geq_1 d(x)$, then $(\lambda x \in A)b(x)a \geq_1 d(c)$.

2.5.2 Lemma. If $a \geq_1 c$ and $b(x) \geq_1 d(x)$, then $b(a) \geq_1 d(c)$.

Proof. By a straightforward induction on the length of the proof of $b(x) \geq_1 d(x)$.

2.5.3 Lemma. If $a \geq_1 b$ and $a \geq_1 c$, then there is an expression d such that $b \geq_1 d$ and $c \geq_1 d$.

Proof. By induction on the sum (or maximum) of the lengths of the proofs of $a \geq_1 b$ and $a \geq_1 c$.

2.5.3.1 a is the constant V. Trivial.

2.5.3.2 a is a variable. Also trivial.

2.5.3.3 a is of the form $(\Pi x \in A_1)A_2(x)$. Then b and c are necessarily of the forms $(\Pi x \in B_1)B_2(x)$ and $(\Pi x \in C_1)C_2(x)$, respectively, where

$$A_1 \geq_1 B_1, \qquad A_2(x) \geq_1 B_2(x),$$
$$A_1 \geq_1 C_1, \qquad A_2(x) \geq_1 C_2(x).$$

By induction hypothesis we can find D_1 and $D_2(x)$ such that

$$B_1 \geq_1 D_1, \qquad B_2(x) \geq_1 D_2(x),$$
$$C_1 \geq_1 D_1, \qquad C_2(x) \geq_1 D_2(x).$$

Letting d be the expression $(\Pi x \in D_1)D_2(x)$, we have $b \geqslant_1 d$ and $c \geqslant_1 d$ as desired.

2.5.3.4 a is of the form $(\lambda x \in A_1)a_2(x)$. This case is completely analogous to the previous one.

2.5.3.5 a is of the form a_2a_1 and b and c are of the forms b_2b_1 and c_2c_1, respectively, where

$$a_1 \geqslant_1 b_1, \qquad a_2 \geqslant_1 b_2,$$
$$a_1 \geqslant_1 c_1, \qquad a_2 \geqslant_1 c_2.$$

By induction hypothesis we can find d_1 and d_2 such that

$$b_1 \geqslant_1 d_1, \qquad b_2 \geqslant_1 d_2,$$
$$c_1 \geqslant_1 d_1, \qquad c_2 \geqslant_1 d_2.$$

We can now take d to be the expression d_2d_1.

2.5.3.6 a is of the form $(\lambda x \in A_1)a_2(x)a_1$ and b and c are of the forms $b_2(b_1)$ and $(\lambda x \in C_1)c_2(x)c_1$, respectively, where

$$a_1 \geqslant_1 b_1, \qquad a_2(x) \geqslant_1 b_2(x),$$
$$a_1 \geqslant_1 c_1, \qquad (\lambda x \in A_1)a_2(x) \geqslant_1 (\lambda x \in C_1)c_2(x).$$

By induction hypothesis we can find d_1 and $d_2(x)$ such that

$$b_1 \geqslant_1 d_1, \qquad b_2(x) \geqslant_1 d_2(x),$$
$$c_1 \geqslant_1 d_1, \qquad c_2(x) \geqslant_1 d_2(x).$$

Using lemma 2.5.2 and definition 2.5.1.6, we can conclude that $b_2(b_1) \geqslant_1 d_2(d_1)$ and $(\lambda x \in C_1)c_2(x)c_1 \geqslant_1 d_2(d_1)$, respectively, so that d can be taken to be $d_2(d_1)$.

2.5.3.7 a is of the form $(\lambda x \in A_1)a_2(x)a_1$ and b and c are of the forms $(\lambda x \in B_1)b_2(x)b_1$ and $c_2(c_1)$, respectively. This case is completely symmetric to the previous one.

2.5.3.8 a is of the form $(\lambda x \in A_1)a_2(x)a_1$ and b and c are of the forms $b_2(b_1)$ and $c_2(c_1)$, respectively, where

$$a_1 \geqslant_1 b_1, \qquad a_2(x) \geqslant_1 b_2(x),$$
$$a_1 \geqslant_1 c_1, \qquad a_2(x) \geqslant_1 c_2(x).$$

By induction hypothesis we can find d_1 and $d_2(x)$ such that

$$b_1 \geqslant_1 d_1, \qquad b_2(x) \geqslant_1 d_2(x),$$
$$c_1 \geqslant_1 d_1, \qquad c_2(x) \geqslant_1 d_2(x).$$

Using lemma 2.5.2 twice, we can conclude that $b_2(b_1) \geqslant_1 d_2(d_1)$ and $c_2(c_1) \geqslant_1 d_2(d_1)$ so that d can be taken to be $d_2(d_1)$.

2.5.4 If we define the relation $a \geqslant_n b$ for arbitrary n by putting $a \geqslant_0 a$ and letting $a \geqslant_{n+1} c$ mean that $a \geqslant_n b$ and $b \geqslant_1 c$ for some b, then it is clear that $a \geqslant b$ if and only if $a \geqslant_n b$ for some n.

2.5.5 Suppose now that $a \geqslant b$ and $a \geqslant c$. Then $a \geqslant_m b$ and $a \geqslant_n c$ for some m and n. By repeatedly applying lemma 2.5.3, we can find d such that $b \geqslant_n d$ and $c \geqslant_m d$ and a fortiori $b \geqslant d$ and $c \geqslant d$. The proof of the property of Church and Rosser is now complete.

2.6 Uniqueness of normal form. <u>If an expression has a normal form, its normal form is unique</u>.

Proof. Let a be an arbitrary expression and suppose that $a \geqslant b$ and $a \geqslant c$ where b and c are both normal. According to the property of Church and Rosser, there is an expression d such that $b \geqslant d$ and $c \geqslant d$. Since b and c are both normal, they must be identical with d and hence with each other. Remember that identity means syntactical identity neglecting differences in the naming of bound variables.

2.7 I shall now set up a formal deductive system which will codify the principles of reasoning that were described informally in the beginning of this paper. The derivations in this system are to be certain tree like arrangements of symbol strings of the form

$$a \in A$$

where a and A are formal expressions. When $a \in A$ has been derived, it may be read

a denotes an object whose type is denoted by A

or, for short,

a is a term with type symbol A.

An expression a is a term if $a \in A$ can be derived for some expression A, and an expression A is a type symbol if there is a derivation of $A \in V$.

A derivation is started by appealing to an axiom or making an assumption and proceeds downwards by means of the rules of inference. Some of the rules of inference have the

property that, when passing from the premises to the conclusion, certain assumptions (all identical in form) are cancelled. Just as Gentzen 1934 did in his system of natural deduction for first order logic, I shall indicate this by enclosing the assumptions in question within square brackets. A derivation is closed if all of its assumptions have been cancelled. Otherwise, it is open.

The axiom, assumptions and rules of inference are as follows.

2.7.1

$$V \in V$$

This is the basic axiom with which every derivation begins. There are no other axioms.

2.7.2 If we have a (possibly open) derivation of $A \in V$, then we may introduce a new variable x and use

$$x \in A$$

as assumption. The type symbol A is to be uniquely associated with x, that is, having made the assumption $x \in A$, we are not allowed to use $x \in B$ as assumption unless B is identical with A.

2.7.3

$$\Pi\text{rule} \qquad \frac{A \in V \qquad \begin{matrix}[x \in A] \\ B(x) \in V\end{matrix}}{(\Pi x \in A)B(x) \in V}$$

This rule of inference is subjected to the restriction that the variable x must not occur in any assumption of the derivation of $B(x) \in V$ other than the indicated $x \in A$. Note that, when passing

from $B(x) \in V$ to $(\Pi x \in A)B(x) \in V$, all assumptions of the form $x \in A$ are cancelled, but, instead, the conclusion becomes dependent on the assumptions of the derivation of $A \in V$ which allowed us to use $x \in A$ as assumption.

2.7.4

$$\lambda\text{rule} \qquad \frac{A \in V \qquad \begin{matrix}[x \in A] \\ b(x) \in B(x)\end{matrix}}{(\lambda x \in A)b(x) \in (\Pi x \in A)B(x)}$$

Similar remarks as for the previous rule of inference.

2.7.5

$$\text{application rule} \qquad \frac{b \in (\Pi x \in A)B(x) \qquad a \in A}{ba \in B(a)}$$

Here, of course, $B(a)$ denotes the result of substituting a for all free occurrences of x in $B(x)$.

2.7.6

$$\text{equality rule} \qquad \frac{a \in A \qquad B \in V}{a \in B} \quad \text{if} \quad A = B$$

An application of the equality rule with A and B identical is redundant and can be removed. Also, whenever convenient, we can assume that there are no two successive applications of the equality rule, because, clearly, two such applications can be made into one application of the same rule.

Example.

$$\dfrac{V\in V \qquad \dfrac{[x\in V] \qquad [x\in X]}{(\lambda x\in X)x\in(\Pi x\in X)X}}{(\lambda X\in V)(\lambda x\in X)x\in(\Pi X\in V)(\Pi x\in X)X}$$

This derivation is closed. The term $(\lambda X\in V)(\lambda x\in X)x$ denotes the function which, when applied to a type, yields the identity function on that type.

Note that an open derivation

$$\begin{array}{ccc} x_1\in A_1 & \cdots & x_n\in A_n(x_1,\ldots,x_{n-1}) \\ & \ddots & \vdots \\ & a(x_1,\ldots,x_n)\in A(x_1,\ldots,x_n) & \end{array}$$

is always accompanied by certain other derivations

$$\begin{array}{ccccc} \vdots & x_1\in A_1 & & x_1\in A_1 \quad \cdots \quad x_{n-1}\in A_{n-1}(x_1,\ldots,x_{n-2}) \\ & \vdots & \cdots & \ddots \quad\quad\quad \vdots \\ A_1\in V & A_2(x_1)\in V & & A_n(x_1,\ldots x_{n-1})\in V \end{array}$$

namely, those derivations that allowed us to introduce successively the assumptions $x_1\in A_1$, ..., $x_n\in A_n(x_1,\ldots,x_{n-1})$.

2.8 Theorem. From a derivation of $a\in A$ we can find a derivation of $A\in V$. In other words, $a\in A$ cannot be derived unless A is a type symbol.

Proof. By induction on the length of the given derivation of $a\in A$. Six cases have to be distinguished.

2.8.1 $V\in V$. Clear.

2.8.2 $x\in A$. Clear, because we can only introduce an assump-

tion of this form if we have previously derived $A \in V$.

2.8.3

$$\dfrac{\begin{array}{cc} & [x \in A] \\ \vdots & \vdots \\ A \in V & B(x) \in V \end{array}}{(\Pi x \in A)B(x) \in V}$$

Clear, because $V \in V$ is an axiom.

2.8.4

$$\dfrac{\begin{array}{cc} & [x \in A] \\ \vdots & \vdots \\ A \in V & b(x) \in B(x) \end{array}}{(\lambda x \in A)b(x) \in (\Pi x \in A)B(x)}$$

By induction hypothesis we get

$$\begin{array}{c} x \in A \\ \vdots \\ B(x) \in V \end{array}$$

which, taken together with the given derivation of $A \in V$, allows us to conclude $(\Pi x \in A)B(x) \in V$ by the Πrule.

2.8.5

$$\dfrac{\begin{array}{cc} \vdots & \vdots \\ b \in (\Pi x \in A)B(x) & a \in A \end{array}}{ba \in B(a)}$$

By induction hypothesis we get a derivation of $(\Pi x \in A)B(x) \in V$. After deleting possible redundant applications of the equality

rule in the end of this derivation, it must be of the form

$$\dfrac{\begin{array}{cc} & [x\in A] \\ \vdots & \vdots \\ A\in V & B(x)\in V \end{array}}{(\prod x\in A)B(x)\in V}.$$

We get the desired derivation

$$\begin{array}{c} \vdots \\ a\in A \\ \vdots \\ B(a)\in V \end{array}$$

of $B(a)\in V$ by substituting a for all free occurrences of x throughout the derivation of $B(x)\in V$ from $x\in A$ and attaching the given derivation of $a\in A$ to the derivation obtained after the substitution.

2.8.6

$$\dfrac{\begin{array}{cc} \vdots & \vdots \\ a\in A & B\in V \end{array}}{a\in B}$$

where A = B. Clear, because the derivation of the right premise is a derivation of $B\in V$ as desired.

2.9 Theorem. *If we have a derivation of* $a\in A$ *and* $a \geqslant b$, *then we can find a derivation of* $b\in A$.

Proof. It will suffice to handle the contraction of a single subterm. By deleting, if necessary, redundant applications

of the equality rule, we can assume that the given derivation has the form

$$\dfrac{\dfrac{\dfrac{\vdots}{A\in V}\qquad\dfrac{\begin{matrix}[x\in A]\\ \vdots\\ b(x)\in B(x)\end{matrix}}{(\lambda x\in A)b(x)\in(\Pi x\in A)B(x)}\qquad\dfrac{\dfrac{\vdots}{C\in V}\qquad\dfrac{\begin{matrix}[x\in C]\\ \vdots\\ D(x)\in V\end{matrix}}{(\Pi x\in C)D(x)\in V}}{(\lambda x\in A)b(x)\in(\Pi x\in C)D(x)}\qquad\dfrac{\vdots}{a\in C}}{(\lambda x\in A)b(x)a\in D(a)}}{\vdots}$$

where $(\Pi x\in A)B(x) = (\Pi x\in C)D(x)$, that is, $A = C$ and $B(x) = D(x)$. If this derivation is rebuilt as follows

$$\dfrac{\dfrac{\dfrac{\vdots}{a\in C}\qquad\dfrac{\vdots}{A\in V}}{a\in A}\qquad\dfrac{\vdots}{a\in C}}{\dfrac{\vdots}{b(a)\in B(a)}\qquad\dfrac{\vdots}{D(a)\in V}}{b(a)\in D(a)}}{\vdots}$$

we obtain the desired derivation in which $(\lambda x\in A)b(x)a$ has been contracted to $b(a)$.

2.10 Theorem. *If* $a\in A$ *and* $b\in B$ *can both be derived and* $a = b$, *then* $A = B$.

This is the formal counterpart of the idea that every mathematical object is of a uniquely determined type.

Proof. That $a = b$ means that there is an expression c such that $a \geqslant c$ and $b \geqslant c$. Using the previous theorem, we can

find derivations of $c \in A$ and $c \in B$. The proof that $A = B$ is by induction on the sum of the lengths of these derivations.

If the last inference of one of the derivations, say the second, is an application of the equality rule

$$\begin{array}{cc} \vdots & \vdots \quad \vdots \\ c \in A & \dfrac{c \in C \qquad B \in V}{c \in B} \end{array}$$

we know by induction hypothesis that $A = C$. Since $B = C$, we can conclude $A = B$ as desired. It remains to consider the five cases that arise when none of the derivations ends with an application of the equality rule. They must then both end with applications of one and the same rule of inference which is determined by the form of c.

2.10.1 $V \in V$. Trivial.

2.10.2 Suppose $x \in A$ and $x \in B$ are both assumptions. Then A and B are identical so that a fortiori $A = B$.

2.10.3 If both derivations end with applications of the Π rule, then A and B are both V so that a fortiori $A = B$.

2.10.4 If both derivations end with applications of the λ rule, they must have the forms

$$\dfrac{\begin{array}{cc} \vdots & \begin{bmatrix} x \in A \end{bmatrix} \\ A \in V & \vdots \\ & b(x) \in B(x) \end{array}}{(\lambda x \in A)b(x) \in (\Pi x \in A)B(x)} \qquad \dfrac{\begin{array}{cc} \vdots & \begin{bmatrix} x \in A \end{bmatrix} \\ A \in V & \vdots \\ & b(x) \in D(x) \end{array}}{(\lambda x \in A)b(x) \in (\Pi x \in A)D(x)}$$

By induction hypothesis $B(x) = D(x)$ and, consequently, $(\Pi x \in A)B(x) = (\Pi x \in A)D(x)$ as desired.

2.10.5 If both derivations end with applications of the rule of application, they must have the forms

$$\frac{\begin{array}{cc}\vdots & \vdots \\ b\in(\prod x\in A)B(x) & a\in A\end{array}}{ba\in B(a)} \qquad \frac{\begin{array}{cc}\vdots & \vdots \\ b\in(\prod x\in C)D(x) & a\in C\end{array}}{ba\in D(a)}$$

By induction hypothesis A = C and B(x) = D(x) so that B(a) = D(a) as desired.

2.11 Theorem. *If we have a derivation of $a\in A$ and b is a term such that a = b, then we can find a derivation of $b\in A$.*

Proof. That b is a term means that we have a derivation of $b\in B$. From the previous theorem we can conclude that A = B. Also, theorem 2.8 allows us to find a derivation of $A\in V$. An application of the equality rule

$$\frac{\begin{array}{cc}\vdots & \vdots \\ b\in B & A\in V\end{array}}{b\in A}$$

now yields the desired derivation.

2.12 Theorem. *If we have a derivation of $a\in A$ and B is a term such that A = B, then we can find a derivation of $a\in B$.*

Proof. First, use theorem 2.8 to obtain a derivation of $A\in V$, and, second, use the previous theorem to obtain a derivation of $B\in V$. We then get the desired derivation

$$\frac{\begin{array}{cc}\vdots & \vdots \\ a\in A & B\in V\end{array}}{a\in B}$$

by applying the equality rule.

3 REPRESENTATION OF SOME LOGICAL AND MATHEMATICAL CONCEPTS IN THE FORMAL THEORY

3.1 Implication and the type of functions from one type to another. For two type symbols A and B we put

$$A \to B = (\Pi x \in A)B$$

where the variable x occurs free neither in A nor in B. In order to avoid an excessive number of parantheses,

$$A_1 \to (\ldots \to (A_{n-1} \to A_n)\ldots)$$

will be abbreviated to

$$A_1 \to \ldots \to A_{n-1} \to A_n.$$

Clearly, $A \to B$ is a type symbol, and, since

$$K = (\lambda x \in A)(\lambda x \in B)x \in A \to B \to A$$

and

$$S = (\lambda z \in A \to B \to C)(\lambda y \in A \to B)(\lambda x \in A)(zx(yx))$$
$$\in (A \to B \to C) \to (A \to B) \to A \to C,$$

the usual positive implicational axioms are satisfied. Except for the type restrictions, K and S are the combinators introduced by Schönfinkel 1924. Modus ponens

$$\frac{b \in A \to B \qquad a \in A}{ba \in B}$$

is a special case of the rule of application.

3.2 Falsehood and the empty type. Prawitz's 1965 definition for second order logic can be carried over to the present formalism, that is, we put

$$\bot = (\overline{\Pi}X \in V)X.$$

Clearly, $\bot$ is a type symbol, and

$$(\lambda X \in V)(\lambda x \in \bot)(xX) \in (\overline{\Pi}X \in V)(\bot \to X)$$

so that the intuitionistic axiom of absurdity is satisfied.

3.3 Negation. If A is a type symbol, we put as usual

$$-A = A \to \bot.$$

Clearly, $-A$ is again a type symbol. Russell's 1903 definition of the negation of A as $(\overline{\Pi}X \in V)(A \to X)$ would do just as well.

3.4 Conjunction and the cartesian product of two types. For two type symbols A and B we put with Russell 1903

$$A \times B = (\overline{\Pi}X \in V)((A \to B \to X) \to X).$$

It is readily verified that $A \times B$ is a type symbol. Define the pairing operation by

$$(a,b) = (\lambda X \in V)(\lambda z \in A \to B \to X)(zab) \in A \times B$$

for a and b with type symbols A and B, respectively, so that

$$(\lambda x \in A)(\lambda y \in B)(x,y) \in A \to B \to A \times B,$$

and introduce the corresponding projections

$$p = (\lambda z \in A \times B)(zA(\lambda x \in A)(\lambda y \in B)x) \in A \times B \to A$$

and

$$q = (\lambda z \in A \times B)(zB(\lambda x \in A)(\lambda y \in B)y) \in A \times B \to B.$$

Then we see that the usual conjunctive axioms are satisfied, and

$$p(a,b) = a \quad \text{and} \quad q(a,b) = b$$

as desired.

3.5 Disjunction and the disjoint union of two types. For two type symbols A and B we put with Russell 1903

$$A + B = (\Pi X \in V)((A \to X) \to (B \to X) \to X).$$

Clearly, $A + B$ is a type symbol. If we define the associated injections by

$$i = (\lambda x \in A)(\lambda X \in V)(\lambda f \in A \to X)(\lambda g \in B \to X)(fx) \in A \to A + B$$

and

$$j = (\lambda y \in B)(\lambda X \in V)(\lambda f \in A \to X)(\lambda g \in B \to X)(gy) \in B \to A + B$$

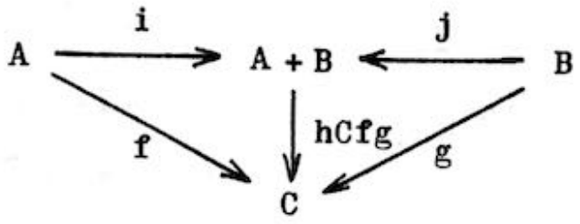
and put

$$h = (\lambda X \in V)(\lambda f \in A \to X)(\lambda g \in B \to X)(\lambda z \in A + B)(zXfg)$$
$$\in (\Pi X \in V)((A \to X) \to (B \to X) \to A + B \to X),$$

we see that the usual axioms for intuitionistic disjunction are satisfied. Also,

$$hCfg(ia) = fa \quad \text{and} \quad hCfg(jb) = gb$$

so that the diagram



commutes.

3.6 Existence and disjoint union. We can use Prawitz's 1965 definition

$$(\Sigma x \in A)B(x) = (\Pi X \in V)((\Pi x \in A)(B(x) \to X) \to X).$$

Here, of course, A is a type symbol and B(x) is a type symbol under the assumption that x is a variable with type symbol A. This guarantees that $(\Sigma x \in A)B(x)$ is a type symbol as well. If we introduce the associated injection

$$\begin{aligned} i &= (\lambda x \in A)(\lambda y \in B(x))(\lambda X \in V)(\lambda f \in (\Pi x \in A)(B(x) \to X))(fxy) \\ &\in (\Pi x \in A)(B(x) \to (\Sigma x \in A)B(x)) \end{aligned}$$

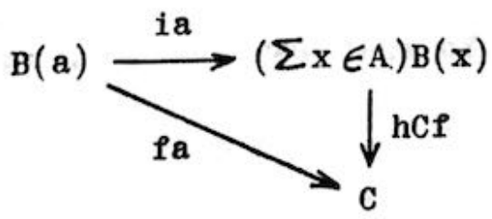
and put

$$\begin{aligned} h &= (\lambda X \in V)(\lambda f \in (\Pi x \in A)(B(x) \to X))(\lambda z \in (\Sigma x \in A)B(x))(zXf) \\ &\in (\Pi X \in V)((\Pi x \in A)(B(x) \to X) \to (\Sigma x \in A)B(x) \to X), \end{aligned}$$

we see that the usual axioms for intuitionistic existence are satisfied. Also,

$$hCf(iab) = fab$$

so that the diagram



commutes. Finally,

$$p = (\lambda z \in (\Sigma x \in A)B(x))(zA(\lambda x \in A)(\lambda y \in B(x))x)$$
$$\in (\Sigma x \in A)B(x) \to A$$

has the property that

$$p(iab) = a.$$

This means that, when p is applied to a proof of $(\Sigma x \in A)B(x)$, we obtain the object of type A which exists by virtue of that proof. Note that, when B does not contain the variable x free, $(\Sigma x \in A)B$ is identical with $A \times B$.

3.7 Power type. Put

$$P = (\lambda X \in V)(X \to V) \in V \to V.$$

Clearly, if A is a type symbol, then PA is a type symbol which denotes the type of propositional functions defined on the type denoted by A. An object of type PA is called a <u>species</u> (or <u>class</u> in the terminology of Principia Mathematica) of objects of type A. Note that a species is always a species of objects of a certain type. The definitions of inclusion, extensional equality, complementation, intersection, and so on, are all standard.

3.8 Identity. Let A be a type symbol. Following Russell 1903, we define the identity relation between objects of type A by putting

$$I = (\lambda x \in A)(\lambda y \in A)(\Pi X \in A \to V)(Xx \to Xy) \in A \to A \to V.$$

It is then easily verified that the laws of identity,

$$(\Pi x \in A)(Ixx),$$

$$(\Pi x \in A)(\Pi y \in A)(Ixy \to Iyx),$$

$$(\Pi x \in A)(\Pi y \in A)(\Pi z \in A)(Ixy \to Iyz \to Ixz),$$

become satisfied. Note that, because of the equality rule, Iab holds provided a and b are definitionally equal terms of type A. The converse will be a consequence of the Hauptsatz.

3.9 Finite types. For every natural number n we put

$$M_n = (\Pi X \in V)(\underbrace{X \to \ldots \to X}_{n} \to X)$$

with the understanding that

$$M_0 = \bot = (\Pi X \in V)X.$$

It is natural to use M_1 to represent not only the one element type but also the logical constant truth $\top$. A simple combinatorial argument shows that

$$m_1 = (\lambda X \in V)(\lambda x_1 \in X)\ldots(\lambda x_n \in X)x_1 \in M_n,$$

$$\vdots$$

$$m_n = (\lambda X \in V)(\lambda x_1 \in X)\ldots(\lambda x_n \in X)x_n \in M_n$$

are the only closed irreducible terms of type M_n and, in particular, that there is no closed irreducible term of type $M_0 = \bot$. Now, let A be an arbitrarily given type symbol and $a_1, \ldots, a_n$ terms with type symbol A. We can then define a function

$$f = (\lambda z \in M_n)(zAa_1\ldots a_n) \in M_n \to A$$

with the property that

$$fm_1 = a_1, \ \ldots, \ fm_n = a_n.$$

3.10 Natural numbers. As in Martin-Löf 1970a we put

$$M = (\Pi X \in V)(X \to (X \to X) \to X) \in V,$$
$$0 = (\lambda X \in V)(\lambda x \in X)(\lambda y \in X \to X)x \in M,$$
$$s = (\lambda z \in M)(\lambda X \in V)(\lambda x \in X)(\lambda y \in X \to X)(y(zXxy)) \in M \to M,$$
$$R = (\lambda X \in V)(\lambda x \in X)(\lambda y \in X \to X)(\lambda z \in M)(zXxy)$$
$$\in (\Pi X \in V)(X \to (X \to X) \to M \to X).$$

Then, if A is a type symbol and a, b and c have type symbols A, $A \to A$ and M, respectively,

$$\begin{cases} RAab0 = a, \\ RAab(sc) = b(RAabc), \end{cases}$$

so that we can define functions by recursion. A simple combinatorial argument shows that the numerals

$$0 = (\lambda X \in V)(\lambda x \in X)(\lambda y \in X \to X)x \in M,$$
$$1 = (\lambda X \in V)(\lambda x \in X)(\lambda y \in X \to X)(yx) \in M,$$
$$2 = (\lambda X \in V)(\lambda x \in X)(\lambda y \in X \to X)(y(yx)) \in M,$$
$$\vdots$$

are the only closed irreducible terms with type symbol M. Except for the type restrictions, this is the representation of the natural numbers invented by Church 1933. Using the definition of the property of being a natural number given by Frege and Dedekind,

$$N = (\lambda x \in M)(\Pi X \in M \to V)(X0 \to (\Pi x \in M)(Xx \to X(sx)) \to Xx),$$

the relativized versions of the Peano axioms,

$$N0,$$

$$(\Pi x \in M)(Nx \to N(sx)),$$

$$(\Pi x \in M)(Nx \to \neg\, I0(sx)),$$

$$(\Pi x \in M)(\Pi y \in M)(Nx \to Ny \to I(sx)(sy) \to Ixy),$$

$$(\Pi X \in M \to V)(X0 \to (\Pi x \in M)(Nx \to Xx \to X(sx)) \to (\Pi x \in M)(Nx \to Xx)),$$

become provable. I have not succeeded in defining the type of natural numbers in such a way that the axiom of induction becomes provable without relativization.

Example. The term RVMP with type symbol $M \to V$ denotes the function which, when applied to a natural number m, yields the mth power of the type of natural numbers. Note that the existence of this function cannot be established in Zermelo's set theory.

4 REDUCTION OF SOME OTHER FORMAL THEORIES TO THE THEORY OF TYPES

4.1 Gödel's primitive recursive functionals of finite type. As is clear from sections 3.1 and 3.10, they can be constructed by means of M, $\rightarrow$, 0, s, R and lambda abstraction.

4.2 Girard's theory. As was pointed out by Martin-Löf 1970a, the theory of Girard 1970 can be given the following simplified form provided we make the inessential change of excluding the zero constant of each type from his theory.

4.2.1 Types.

4.2.1.1 The type variables X, Y, ... are types.

4.2.1.2 If A and B are types, then so is $A \rightarrow B$.

4.2.1.3 If B(X) is a type, then so is $\Pi XB(X)$.

4.2.2 Terms.

4.2.2.1 The variables x, y, ... of type A are terms of type A.

4.2.2.2 If x is a variable of type A and b(x) is a term of type B, then $\lambda xb(x)$ is a term of type $A \rightarrow B$.

4.2.2.3 If b(X) is a term of type B(X) and X does not occur free in the type of a free variable in b(X), then $\lambda Xb(X)$ is a term of type $\Pi XB(X)$.

4.2.2.4 If a and b are terms of types A and $A \rightarrow B$, respectively, then ba is a term of type B.

4.2.2.5 If A is a type and b is a term of type $\Pi XB(X)$, then bA is a term of type B(A).

4.2.3 When formulated in this way, the reduction of Girard's theory to the theory of types becomes trivial. Indeed, the type variables are represented by variables with type symbol V, and $\rightarrow$, Π, lambda abstraction and application are represented by the entities which are denoted and named likewise in the theory of types.

Another way of arguing is this. Girard's theory in the above formulation is isomorphic to that part of intuitionistic second order logic which is obtained by excluding everything that has to do with individuals. Therefore, being a subsystem of second order logic and a fortiori of simple type theory, its reduction to the theory of types follows from the reduction of simple type theory which we now proceed to carry out.

4.3 Simple type theory. We shall consider the following intuitionistic and intentional version of the theory of simple types. The classical and extensional version can be reduced to the intuitionistic and intentional version by means of the double negation interpretation and Takeuti's 1953 restriction theory.

4.3.1 Types.

4.3.1.1 0 is a type.

4.3.1.2 If $n \geqslant 0$ and $\tau_1, \ldots, \tau_n$ are types, then so is $(\tau_1, \ldots, \tau_n)$.

4.3.2 Terms and formulae. A formula is the same as a term of type ().

4.3.2.1 A variable of type τ is a term of type τ.

4.3.2.2 If f is an nary function symbol and $t_1, \ldots, t_n$ are terms of type 0, then $ft_1 \ldots t_n$ is a term of type 0.

4.3.2.3 If t is a term of type $(\tau_1,\ldots,\tau_n)$ and $t_1, \ldots, t_n$ are terms of types $\tau_1, \ldots, \tau_n$, respectively, then $tt_1\ldots t_n$ is a formula.

4.3.2.4 If A and B are formulae, then so is $A \to B$.

4.3.2.5 If x is a variable of type τ and B(x) is a formula, then $\forall xB(x)$ is a formula.

4.3.2.6 If $x_1, \ldots, x_n$ are variables of types $\tau_1, \ldots, \tau_n$, respectively, and $B(x_1,\ldots,x_n)$ is a formula, then $\lambda x_1\ldots x_nB(x_1,\ldots,x_n)$ is a term of type $(\tau_1,\ldots,\tau_n)$.

4.3.3 Reduction of terms. One term reduces to another if the latter can be obtained from the former by repeated contractions of subformulae according to the reduction rule

$$\lambda x_1\ldots x_nB(x_1,\ldots,x_n)t_1\ldots t_n \geqslant B(t_1,\ldots,t_n).$$

The property of Church and Rosser is easily established for this kind of reduction. Two terms are definitionally or intentionally equal if they reduce to a common term. Intentional equality will be denoted by =. A simple combinatorial argument, very similar to the proof of normalization for first order logic, shows that every term and, in particular, every formula can be reduced to normal form.

4.3.4 Rules of inference. The following is an extension of Gentzen's 1934 calculus of natural deduction for first order logic to higher types.

$$\to\text{introduction} \qquad \frac{\begin{array}{c}[A]\\ B\end{array}}{A \to B}$$

$\rightarrow$elimination modus ponens	$\dfrac{A \rightarrow B \quad A}{B}$
$\forall$introduction	$\dfrac{B(x)}{\forall x B(x)}$
$\forall$elimination	$\dfrac{\forall x B(x)}{B(t)}$
equality	$\dfrac{A}{B}$ if $A = B$

4.3.5 Translation into the theory of types. I shall denote the translation operation by a star.

4.3.5.1 Types.

4.3.5.1.1 0^* is taken to be some fixed variable with type symbol V.

4.3.5.1.2 $(\tau_1, \ldots, \tau_n)^*$ is $\tau_1^* \rightarrow \ldots \rightarrow \tau_n^* \rightarrow V$. In particular, $()^*$ is V.

4.3.5.2 Terms and formulae.

4.3.5.2.1 A variable of type τ is translated into a variable with type symbol τ^*.

4.3.5.2.2 If f is an nary function symbol, then f^* is some fixed variable with type symbol

$$\underbrace{0^* \rightarrow \ldots \rightarrow 0^*}_{n} \rightarrow 0^*$$

and, if $t_1, \ldots, t_n$ are terms of type 0, we let $(ft_1 \ldots t_n)^*$ be $f^* t_1^* \ldots t_n^*$.

4.3.5.2.3 $(tt_1 \ldots t_n)^*$ is $t^* t_1^* \ldots t_n^*$.

4.3.5.2.4 $(A \to B)^*$ is $A^* \to B^*$.

4.3.5.2.5 If x is a variable of type τ, we let $(\forall x B(x))^*$ be $(\prod x \in \tau^*) B^*(x)$.

4.3.5.2.6 If $x_1, \ldots, x_n$ are variables of types $\tau_1, \ldots, \tau_n$, respectively, we let $(\lambda x_1 \ldots x_n B(x_1, \ldots, x_n))^*$ be $(\lambda x_1 \in \tau_1^*) \ldots (\lambda x_n \in \tau_n^*) B^*(x_1, \ldots, x_n)$.

4.3.5.3 Derivations. By induction on the length of a derivation a of a formula A in simple type theory I shall construct a term a^* and a derivation of $a^* \in A^*$ in the theory of types.

4.3.5.3.1 Corresponding to an assumption A in simple type theory we make an assumption of the form $x \in A^*$ in the theory of types.

4.3.5.3.2 $\to$introduction.

$$\dfrac{\begin{array}{c}[A] \\ \vdots \\ B\end{array}}{A \to B}$$

By induction hypothesis we have constructed $b^*(x)$ and a derivation

$$\begin{array}{c} x \in A^* \\ \vdots \\ b^*(x) \in B^* \end{array}$$

Define the translation of the derivation of $A \to B$ to be $(\lambda x \in A^*) b^*(x)$. The λ rule allows us then to infer

$$\dfrac{\begin{array}{c}[x \in A^*] \\ \vdots \\ b^*(x) \in B^*\end{array}}{(\lambda x \in A^*) b^*(x) \in A^* \to B^*}$$

as desired.

4.3.5.3.3 →elimination.

$$\frac{\begin{array}{cc}\vdots & \vdots \\ A \to B & A\end{array}}{B}$$

By induction hypothesis we have constructed a^* and b^* and derivations

$$\begin{array}{cc}\vdots & \vdots \\ b^* \in (A \to B)^* & a^* \in A^*\end{array}$$

Define the translation of the derivation of B to be b^*a^*. The rule of application allows us then to infer

$$\frac{\begin{array}{cc}\vdots & \vdots \\ b^* \in (A \to B)^* & a^* \in A^*\end{array}}{b^*a^* \in B^*}$$

as desired.

4.3.5.3.4 $\forall$introduction.

$$\frac{\begin{array}{c}\vdots \\ B(x)\end{array}}{\forall x B(x)}$$

Let τ be the type of the variable x. By induction hypothesis we have constructed $b^*(x)$ and a derivation

$$\begin{array}{c} x \in \tau^* \\ \vdots \\ b^*(x) \in B^*(x)\end{array}$$

Define the translation of the derivation of $\forall xB(x)$ to be $(\lambda x \in \tau^*)b^*(x)$. The λrule allows us then to infer

$$\dfrac{\begin{array}{c}[x \in \tau^*] \\ \vdots \\ b^*(x) \in B^*(x)\end{array}}{(\lambda x \in \tau^*)b^*(x) \in (\forall xB(x))^*}$$

as desired.

4.3.5.3.5 $\forall$ elimination.

$$\dfrac{\begin{array}{c}\vdots \\ \forall xB(x)\end{array}}{B(t)}$$

By induction hypothesis we have constructed b^* and a derivation

$$\begin{array}{c}\vdots \\ b^* \in (\forall xB(x))^*\end{array}$$

Also, if τ is the type of the term t, we can find a derivation

$$\begin{array}{c}\vdots \\ t^* \in \tau^*\end{array}$$

Define the translation of the derivation of B(t) to be b^*t^*. The rule of application allows us then to infer

$$\dfrac{\begin{array}{c}\vdots \\ b^* \in (\forall xB(x))^*\end{array} \qquad \begin{array}{c}\vdots \\ t^* \in \tau^*\end{array}}{b^*t^* \in (B(t))^*}$$

as desired. Here we have tacitly used the fact that $(\forall x B(x))^*$ is $(\Pi x \in \tau^*) B^*(x)$ and that $(B(t))^*$ is $B^*(t^*)$.

4.3.5.3.6 Equality.

$$\begin{array}{c} \vdots \\ A \\ \hline B \end{array}$$

where A = B. By induction hypothesis we have constructed a^* and a derivation

$$\begin{array}{c} \vdots \\ a^* \in A^* \end{array}$$

Also, we can find a derivation

$$\begin{array}{c} \vdots \\ B^* \in V \end{array}$$

Define the translation of the derivation of B to be a^*. The equality rule allows us then to infer

$$\frac{\begin{array}{c} \vdots \\ a^* \in A^* \end{array} \qquad \begin{array}{c} \vdots \\ B^* \in V \end{array}}{a^* \in B^*}$$

as desired. Here we have tacitly used the fact that A = B implies $A^* = B^*$. The translation of simple type theory is now complete.

5 THE HAUPTSATZ AND ITS CONSEQUENCES

5.1 **Hauptsatz.** Every term reduces to a normal term.

Proof. In the course of the proof Roman letters will just as before denote formal expressions, whereas Greek letters will denote abstract objects such as types, functions and propositional functions.

The idea of the proof is this. With every term $a(x_1,\ldots,x_n)$ all of whose free variables occur among $x_1, \ldots, x_n$ I shall associate an abstract object

$$\alpha_{a(x_1,\ldots,x_n)}(\xi_1,\ldots,\xi_n)$$

which depends on certain other abstract objects $\xi_1, \ldots, \xi_n$ which are associated with the variables $x_1, \ldots, x_n$. For the sake of notational simplicity, I shall write α_a without exhibiting explicitly the free variables and the abstract objects which are associated with them.

When A is a type symbol, α_A is to be a triple $(\tau_A, \omega_A, \theta_A)$ where τ_A is a type, ω_A an object of type τ_A (which we can forget about at this stage) and θ_A a binary propositional function whose first argument is a formal expression and whose second argument is an object of type τ_A. $\theta_A(a,\xi)$ may be thought of as expressing that ξ is a computation of a. The predicate θ_A will be constructed so as to have the crucial property

$$\theta_A(a,\xi) \text{ implies that } a \text{ is normalizable.}$$

Finally, it will be shown that if a is a term, that is, if there

is a derivation of $a \in A$, then α_a is an object of type τ_A such that $\theta_A(c, \alpha_a)$ where c is a term to which a reduces. Because of the property of θ_A just stated, c is normalizable. But a reduces to c and so a is normalizable as well. The proof of the Hauptsatz will then be complete.

5.1.1 The definition of α_a is by induction on the construction of a.

5.1.1.1 α_V is the triple $(\tau_V, \omega_V, \theta_V)$ where

τ_V is the type of triples (τ, ω, θ) where τ is a type, ω an object of that type, and θ a binary propositional function whose first argument is a formal expression and whose second argument is an object of type τ,

ω_V is the triple (τ, ω, θ) where τ is some fixed non empty type, ω some fixed object of that type, and $\theta(a, \xi)$ the proposition that a is normalizable, and

θ_V is the binary propositional function whose value for an expression A and an object (τ, ω, θ) of type τ_V is the conjunction of (the universal closures of) the following three propositions.

5.1.1.1.1 A is normal.

5.1.1.1.2 $\theta(a, \xi)$ implies that a is normalizable.

5.1.1.1.3 $\theta(a, \omega)$ provided a is a normal expression of the form $ya_1 \ldots a_n$ where y is a variable.

5.1.1.2 $\alpha_x(\xi)$ is ξ. This definition makes sense, of course, no matter what the type of the object ξ is.

5.1.1.3 $\alpha_{(\Pi x \in A)B(x)}$ is the triple $(\tau_{(\Pi x \in A)B(x)}, \omega_{(\Pi x \in A)B(x)}, \theta_{(\Pi x \in A)B(x)})$ where

$\tau_{(\Pi x \in A)B(x)}$ is the type of functions η which to an object ξ of type τ_A assigns an object $\eta(\xi)$ of type $\tau_{B(x)}(\xi)$,

$\omega_{(\Pi x \in A)B(x)}$ is the function which to an object ξ of type τ_A assigns $\omega_{B(x)}(\xi)$, and

$\theta_{(\Pi x \in A)B(x)}(b, \eta)$ is the proposition that, for all expressions a and objects ξ of type τ_A, if $\theta_A(a, \xi)$ then ba reduces to an expression d such that $\theta_{B(x)}(\xi)(d, \eta(\xi))$, and, furthermore, if b reduces to $(\lambda x \in C)d(x)$ then C is normalizable.

This definition makes sense provided α_A is of type τ_V and $\alpha_{B(x)}(\xi)$ is of type τ_V for ξ of type τ_A.

5.1.1.4 $\alpha_{(\lambda x \in A)b(x)}$ is the function which to an object ξ of type τ_A assigns $\alpha_{b(x)}(\xi)$. This definition makes sense provided α_A is a welldefined object of type τ_V and $\alpha_{b(x)}(\xi)$ is welldefined for ξ of type τ_A.

5.1.1.5 α_{ba} is the result $\alpha_b(\alpha_a)$ of applying α_b to α_a. This definition makes sense provided we know that α_b is a function and that α_a is an object whose type is the domain of that function.

5.1.2 Substitution property. $\alpha_{b(x)}(\alpha_a) = \alpha_{b(a)}$ <u>in the sense that if one member is welldefined so is the other and the two are definitionally equal</u>.

Proof. Immediate by induction on the construction of b(x) and using the following three obvious properties.

5.1.2.1 If $\alpha_A = \alpha_C$ and $\alpha_{B(x)}(\xi) = \alpha_{D(x)}(\xi)$ for ξ of type $\tau_A = \tau_C$, then $\alpha_{(\Pi x \in A)B(x)} = \alpha_{(\Pi x \in C)D(x)}$. Here as before and in the following the equality sign denotes definitional equality.

5.1.2.2 If $\alpha_A = \alpha_C$ and $\alpha_{b(x)}(\xi) = \alpha_{d(x)}(\xi)$ for ξ of type $\tau_A = \tau_C$, then $\alpha_{(\lambda x \in A)b(x)} = \alpha_{(\lambda x \in C)d(x)}$.

5.1.2.3 If $\alpha_a = \alpha_c$ and $\alpha_b = \alpha_d$, then $\alpha_{ba} = \alpha_{dc}$.

5.1.3 Compatibility with equality. *If* α_a *and* α_b *are both* welldefined *and* $a = b$, *then* $\alpha_a = \alpha_b$.

Proof. It suffices to show that, if α_a is welldefined and a reduces to b, then α_b is welldefined and $\alpha_a = \alpha_b$. Because of the properties 5.1.2.1, 5.1.2.2 and 5.1.2.3 it even suffices to show that, if $\alpha_{(\lambda x \in A)b(x)a}$ is welldefined, then so is $\alpha_{b(a)}$ and the two are definitionally equal. Now, this is indeed so, because

$$\alpha_{(\lambda x \in A)b(x)a} = \alpha_{(\lambda x \in A)b(x)}(\alpha_a) = \alpha_{b(x)}(\alpha_a) = \alpha_{b(a)}.$$

Here the first two equalities hold by definition and the third because of the substitution property.

5.1.4 By induction on the length of a derivation

$$\begin{array}{ccc} x_1 \in A_1 & \ldots & x_n \in A_n(x_1,\ldots,x_{n-1}) \\ \ddots & & \ddots \\ & a(x_1,\ldots,x_n) \in A(x_1,\ldots,x_n) & \end{array}$$

I shall prove that,

if α_{A_1} is a welldefined object of type τ_V,

$\vdots$

if $\alpha_{A_n(x_1,\ldots,x_{n-1})}(\xi_1,\ldots,\xi_{n-1})$ is a welldefined object of type τ_V for ξ_1 of type τ_{A_1}, ..., ξ_{n-1} of type $\tau_{A_{n-1}(x_1,\ldots,x_{n-2})}(\xi_1,\ldots,\xi_{n-2})$,
then $\alpha_{A(x_1,\ldots,x_n)}(\xi_1,\ldots,\xi_n)$ and $\alpha_{a(x_1,\ldots,x_n)}(\xi_1,\ldots,\xi_n)$ are welldefined objects of types τ_V and $\tau_{A(x_1,\ldots,x_n)}(\xi_1,\ldots,\xi_n)$,

respectively, such that, if

$$\theta_{A_1}(a_1, \xi_1), \ldots, \theta_{A_n(x_1,\ldots,x_{n-1})}(\xi_1,\ldots,\xi_{n-1})(a_n, \xi_n),$$

then $a(a_1,\ldots,a_n)$ reduces to an expression c such that

$$\theta_{A(x_1,\ldots,x_n)}(\xi_1,\ldots,\xi_n)(c, \alpha_{a(x_1,\ldots,x_n)}(\xi_1,\ldots,\xi_n)).$$

Six cases have to be distinguished depending on the form of the last inference of the derivation. In order to alleviate the notation, I shall not exhibit explicitly any assumptions except the one which is possibly involved in the particular inference under consideration.

5.1.4.1

$$V \in V$$

We have to show that α_V, that is, the triple $(\tau_V, \omega_V, \theta_V)$, is an object of type τ_V, which is immediately clear, and that the proposition $\theta_V(V, \alpha_V)$ holds. $\theta_V(V, \alpha_V)$ is the conjunction of the following three propositions which we proceed to verify.

5.1.4.1.1 V is normal. Clear.

5.1.4.1.2 $\theta_V(A, \alpha)$ implies that A is normalizable. Clear, because $\theta_V(A, \alpha)$ is the conjunction of three propositions, the first of which says that A is normal.

5.1.4.1.3 $\theta_V(A, \omega_V)$ provided A is a normal expression of the form $ya_1\ldots a_n$. Clear, because in that case A is normal, and, owing to the definition of the triple (τ, ω, θ) which is ω_V, it has the property that $\theta(a, \xi)$ implies that a is normalizable as well as the property that $\theta(a, \omega)$ holds provided a is a normal expression of the form $ya_1\ldots a_n$.

5.1.4.2

$$x \in A$$

We have to show that, if α_A is a welldefined object of type τ_V, then α_A is a welldefined object of type τ_V and, for ξ of type τ_A, $\alpha_x(\xi)$ is a welldefined object of type τ_A such that if $\theta_A(a, \xi)$ then a reduces to an expression c such that $\theta_A(c, \alpha_x(\xi))$. This is trivial because $\alpha_x(\xi)$ is ξ and c can be taken to be a itself.

5.1.4.3

$$\frac{\begin{array}{cc} & [x \in A] \\ \vdots & \vdots \\ A \in V & B(x) \in V \end{array}}{(\Pi x \in A) B(x) \in V}$$

By induction hypothesis we know that α_A is a welldefined object of type τ_V and that $A \geqslant C$ such that $\theta_V(C, \alpha_A)$. Also, for ξ of type τ_A, $\alpha_{B(x)}(\xi)$ is a welldefined object of type τ_V such that $\theta_A(a, \xi)$ implies that $B(a) \geqslant D$ such that $\theta_V(D, \alpha_{B(x)}(\xi))$. Consequently, $\alpha_{(\Pi x \in A) B(x)}$ is welldefined by 5.1.1.3. It remains only to verify that $(\Pi x \in A) B(x) \geqslant E$ such that $\theta_V(E, \alpha_{(\Pi x \in A) B(x)})$, that is, that the following three conditions are fullfilled.

5.1.4.3.1 E is normal. We know already that $\theta_V(C, \alpha_A)$ so that C is normal and $\theta_A(x, \omega_A)$. Also, $\theta_A(x, \omega_A)$ implies that $B(x) \geqslant D(x)$ such that $\theta_V(D(x), \alpha_{B(x)}(\omega_A))$ so that D(x) is normal as well. If we take E to be $(\Pi x \in C) D(x)$, then, clearly, $(\Pi x \in A) B(x) \geqslant E$ and E is normal.

5.1.4.3.2 $\theta_{(\Pi x\in A)B(x)}(b,\eta)$ implies that b is normalizable. By definition $\theta_{(\Pi x\in A)B(x)}(b,\eta)$ is the proposition

> for all a and ξ of type τ_A, if $\theta_A(a,\xi)$ then $ba \geqslant d$ such that $\theta_{B(x)}(\xi)(d,\eta(\xi))$, and, furthermore, if $b \geqslant (\lambda x\in C)d(x)$ then C is normalizable.

ω_A is an object of type τ_A such that $\theta_A(x,\omega_A)$. Hence $bx \geqslant d$ such that $\theta_{B(x)}(\omega_A)(d,\eta(\omega_A))$ which implies that d is normalizable since $B(x) \geqslant D(x)$ such that $\theta_V(D(x),\alpha_{B(x)}(\omega_A))$. Finally, if d is normalizable, then so is bx, and, if bx is normalizable, then so is b, because b cannot reduce to $(\lambda x\in C)d(x)$ unless C is normalizable.

5.1.4.3.3 $\theta_{(\Pi x\in A)B(x)}(b,\omega_{(\Pi x\in A)B(x)})$ holds provided b is a normal expression of the form $ya_1\ldots a_n$. Remember that $\omega_{(\Pi x\in A)B(x)}(\xi)$ was defined to be $\omega_{B(x)}(\xi)$ for ξ of type τ_A. Therefore, we have to verify that if $\theta_A(a,\xi)$ then $ba \geqslant d$ such that $\theta_{B(x)}(\xi)(d,\omega_{B(x)}(\xi))$. Now, since $A \geqslant C$ such that $\theta_V(C,\alpha_A)$, $\theta_A(a,\xi)$ implies that a is normalizable. Let d be the normal expression $ya_1\ldots a_n a_{n+1}$ where a_{n+1} is the normal form of a. Then $\theta_{B(x)}(\xi)(d,\omega_{B(x)}(\xi))$ because $B(a) \geqslant D$ such that $\theta_V(D,\alpha_{B(x)}(\xi))$.

5.1.4.4

$$\frac{\begin{array}{cc} & [x\in A] \\ \vdots & \vdots \\ A\in V & b(x)\in B(x)\end{array}}{(\lambda x\in A)b(x)\in(\Pi x\in A)B(x)}$$

By induction hypothesis we know that α_A is a welldefined object of type τ_V and that $A \geqslant C$ such that $\theta_V(C,\alpha_A)$. Also, for ξ of

type τ_A, $\alpha_{B(x)}(\xi)$ and $\alpha_{b(x)}(\xi)$ are welldefined objects of type τ_V and $\tau_{B(x)}(\xi)$, respectively, such that if $\theta_A(a,\xi)$ then $b(a) \geqslant d$ such that $\theta_{B(x)}(\xi)(d,\alpha_{b(x)}(\xi))$. We have to show that $\alpha_{(\Pi x\in A)B(x)}$ and $\alpha_{(\lambda x\in A)b(x)}$ are welldefined objects of type τ_V and $\tau_{(\Pi x\in A)B(x)}$, respectively, which is clear from 5.1.1.3 and 5.1.1.4. The treatment of this case will be completed by showing that

$$\theta_{(\Pi x\in A)B(x)}((\lambda x\in A)b(x),\alpha_{(\lambda x\in A)b(x)}),$$

that is, that, for all a and ξ of type τ_A, if $\theta_A(a,\xi)$ then $(\lambda x\in A)b(x)a \geqslant d$ such that

$$\theta_{B(x)}(\xi)(d,\alpha_{(\lambda x\in A)b(x)}(\xi)),$$

and, furthermore, if $(\lambda x\in A)b(x) \geqslant (\lambda x\in C)d(x)$ then C is normalizable. Now, this is indeed so, because $(\lambda x\in A)b(x)a \geqslant b(a) \geqslant d$ and $\alpha_{(\lambda x\in A)b(x)}(\xi)$ was defined to be $\alpha_{b(x)}(\xi)$ for ξ of type τ_A, and, furthermore, A is normalizable by induction hypothesis. Here I have tacitly used the fact that, if $A \geqslant C$ and A is normalizable, then so is C because of the property of Church and Rosser.

5.1.4.5

$$\frac{\begin{array}{cc}\vdots & \vdots\\ b\in(\Pi x\in A)B(x) & a\in A\end{array}}{ba\in B(a)}$$

By induction hypothesis we know that α_A and $\alpha_{(\Pi x\in A)B(x)}$ are welldefined objects of type τ_V. Also, α_a and α_b are welldefined objects of type τ_A and $\tau_{(\Pi x\in A)B(x)}$, respectively, and $a \geqslant c$ and $b \geqslant d$ such that $\theta_A(c,\alpha_a)$ and $\theta_{(\Pi x\in A)B(x)}(d,\alpha_b)$. Remembering the definition of the latter predicate, we can conclude that $dc \geqslant f$ such that

$$\theta_{B(x)}(\alpha_a)(f,\alpha_b(\alpha_a)).$$

since α_{ba} was defined to be $\alpha_b(\alpha_a)$ and $\alpha_{B(x)}(\alpha_a) = \alpha_{B(a)}$ because of the substitution property, we can conclude

$$\theta_{B(a)}(f, \alpha_{ba})$$

and $ba \geqslant dc \geqslant f$ as desired.

5.1.4.6

$$\frac{\begin{array}{cc}\vdots & \vdots \\ a \in A & B \in V\end{array}}{a \in B}$$

where A = B. By induction hypothesis α_A and α_a are welldefined objects of type τ_V and τ_A, respectively, and $a \geqslant c$ such that $\theta_A(c, \alpha_a)$. Also, α_B is a welldefined object of type τ_V. Since A = B, we can conclude that $\alpha_A = \alpha_B$ by using 5.1.3. Consequently,

$$\theta_B(c, \alpha_a)$$

where $a \geqslant c$ as desired.

5.1.5 We are now almost done. Suppose that we are given an arbitrary derivation

$$\begin{array}{ccc} x_1 \in A_1 & \cdots & x_n \in A_n(x_1, \ldots, x_{n-1}) \\ & \ddots & \vdots \\ & a(x_1, \ldots, x_n) \in A(x_1, \ldots, x_n) & \end{array}$$

with accompanying derivations

$$\begin{array}{ccccc} \vdots & x_1 \in A_1 & & x_1 \in A_1 & \cdots \quad x_{n-1} \in A_{n-1}(x_1, \ldots, x_{n-2}) \\ & \vdots & \cdots & \ddots & \vdots \\ A_1 \in V & A_2(x_1) \in V & & & A_n(x_1, \ldots, x_{n-1}) \in V \end{array}$$

then we know that

α_{A_1} is a welldefined object of type τ_V and that $A_1 \geqslant C_1$ such that $\theta_V(C_1, \alpha_{A_1})$,

$\vdots$

$\alpha_{A_n(x_1,\ldots,x_{n-1})}(\xi_1,\ldots,\xi_{n-1})$ is a welldefined object of type τ_V for ξ_1 of type τ_{A_1}, ..., ξ_{n-1} of type $\tau_{A_{n-1}(x_1,\ldots,x_{n-2})}(\xi_1,\ldots,\xi_{n-2})$ such that, if $\theta_{A_1}(a_1, \xi_1)$, ..., $\theta_{A_{n-1}(x_1,\ldots,x_{n-2})}(\xi_1,\ldots,\xi_{n-2})(a_{n-1}, \xi_{n-1})$, then $A_n(a_1,\ldots,a_{n-1}) \geqslant C_n$ such that

$$\theta_V(C_n, \alpha_{A_n(x_1,\ldots,x_{n-1})}(\xi_1,\ldots,\xi_{n-1})).$$

Thus, if we put

$$\omega_1 = \omega_{A_1}, \ \ldots, \ \omega_n = \omega_{A_n(x_1,\ldots,x_{n-1})}(\omega_1,\ldots,\omega_{n-1}),$$

then

$$\theta_{A_1}(x_1, \omega_1), \ \ldots, \ \theta_{A_n(x_1,\ldots,x_{n-1})}(\omega_1,\ldots,\omega_{n-1})(x_n, \omega_n)$$

so that we can conclude that $a(x_1,\ldots,x_n)$ reduces to an expression c such that

$$\theta_{A(x_1,\ldots,x_n)}(\omega_1,\ldots,\omega_n)(c, \alpha_{a(x_1,\ldots,x_n)}(\omega_1,\ldots,\omega_n)).$$

On the other hand, by theorem 2.8 there is a derivation

$$\begin{array}{ccc} x_1 \in A_1 & \ldots & x_n \in A_n(x_1,\ldots,x_{n-1}) \\ & \ddots \quad \ddots & \\ & A(x_1,\ldots,x_n) \in V & \end{array}$$

so that $A(x_1,\ldots,x_n)$ reduces to an expression C such that

$$\theta_V(c, \alpha_{A(x_1,\ldots,x_n)}(\omega_1,\ldots,\omega_n)).$$

Hence c is normalizable and, since $a(x_1,\ldots,x_n)$ reduces to c, so is $a(x_1,\ldots,x_n)$. But $a(x_1,\ldots,x_n)$ was an arbitrary term and therefore the proof of the Hauptsatz is complete.

5.2 Corollary. *There is no closed term with type symbol* $\perp$.

This corollary expresses that the theory of types is simply consistent.

Proof. Because of the Hauptsatz it suffices to show that there is no closed normal term with type symbol $\perp$. A normal term is either V, or of the form $(\Pi x \in A)B(x)$ with normal A and B(x), or of the form $(\lambda x \in A)b(x)$ with normal A and b(x), or of the form $ya_1 \ldots a_n$ with normal $a_1, \ldots, a_n$. Hence a closed normal term with type symbol $(\Pi X \in V)X$ would have to be of the form $(\lambda X \in V)b(X)$ where b(X) is normal and has type symbol X. But then b(X) would have to be of the form $ya_1 \ldots a_n$ which is impossible unless n = 0 and y is X. This, however, is also impossible because X has type symbol V and not X.

5.3 Corollary. *A closed term with type symbol* M *reduces to a numeral*.

Proof. This follows immediately from the Hauptsatz, because, as remarked in section 3.11, the closed normal terms with type symbol M are precisely the numerals.

5.4 Corollary. *A number theoretic function which can be constructed in the theory of types is mechanically computable*.

Proof. By saying that a number theoretic function can be constructed in the theory of types I mean that there is

a closed term f with type symbol $M \to M$ which denotes it. Suppose we want to find the value of the function for a certain natural number which is denoted by the numeral m. Then fm denotes the value of the function for this argument. But fm is a closed term with type symbol M so that, according to the previous corollary, it reduces to a numeral n. It only remains to remark that the normal form of a term can be found in a purely mechanical way, that is, by manipulating symbol strings according to rules which do not refer to their meaning.

Of course, the fact that there is a (not necessarily mechanical) rule for computing every function in the present theory of types does not require any proof at all, once it has been recognized that the axiom and rules of inference of the theory are consonant with the constructive notion of function according to which a function is the same as a rule or method.

5.5 Corollary. Let a and b be two closed terms with closed type symbol A. Then there is a closed term with type symbol $(\Pi X \in A \to V)(Xa \to Xb)$ if and only if $a = b$.

Proof. It is readily verified that $(\lambda X \in A \to V)(\lambda x \in Xa)x$ is a closed term which has type symbol $(\Pi X \in A \to V)(Xa \to Xb)$ provided $a = b$. Conversely, suppose there is a closed term with type symbol $(\Pi X \in A \to V)(Xa \to Xb)$ where we can assume without restriction of generality that a and A are both normal. Since neither $A \to V$ nor V is intentionally equal to $Xa \to Xb$, the normal form of this term, which exists according to the Hauptsatz, must be of the form $(\lambda X \in A \to V)(\lambda x \in Xa)c(X,x)$ where the normal term $c(X,x)$ has type symbol Xb. But $c(X,x)$

must be of the form $ya_1 \ldots a_n$ where y is either X or x. Hence the type symbol of c(X,x) must be intentionally equal to $A \rightarrow V$, V or Xa, and this is only possible if c(X,x) is x and a = b.

5.6 Corollary. There is no term of the form aa. In other words, selfapplication is impossible.

Note that it is selfapplication that gives rise to the non normalizable terms like $\lambda x(xx)\lambda x(xx)$ (for other examples see Curry and Feys 1958) in Church's type free calculus of lambda conversion.

Proof. Let A be the normal form of the type symbol of a. For aa to be a term, A would have to be identical with $(\Pi x \in A)B(x)$ for some B(x) which is impossible.

5.7 Corollary. For two terms a and b it can be mechanically decided whether or not a = b.

Proof. The decision procedure is this. Reduce a and b to normal form, which is possible according to the Hauptsatz, and check whether or not their normal forms are identical except possibly for differences in the naming of their bound variables.

5.8 Corollary. It can be mechanically decided whether or not a closed expression is a term.

For expressions that are not necessarily closed the appropriate formulation of the corollary is as follows. Given expressions

$$A_1,\ A_2(x_1),\ \ldots,\ A_n(x_1,\ldots,x_{n-1}) \text{ and } a(x_1,\ldots,x_n)$$

containing free variables in the indicated way, it can be

mechanically decided whether or not there exist derivations

$$\begin{array}{ccccc} & x_1 \in A_1 & & x_1 \in A_1 & \dots \quad x_{n-1} \in A_{n-1}(x_1,\dots,x_{n-2}) \\ \vdots & \vdots & \dots & \ddots & \\ A_1 \in V & A_2(x_1) \in V & & & A_n(x_1,\dots,x_{n-1}) \in V \end{array}$$

and

$$\begin{array}{ccc} x_1 \in A_1 & \dots & x_n \in A_n(x_1,\dots,x_{n-1}) \\ \ddots & & \ddots \\ & a(x_1,\dots,x_n) \in A(x_1,\dots,x_n) & \end{array}$$

In particular, for n = 0 we have the assertion of the corollary.

Proof. We use induction on the sum of the lengths of the expressions A_1, ..., $A_n(x_1,\dots,x_{n-1})$ and $a(x_1,\dots,x_n)$. Five cases have to be distinguished depending on the form of $a(x_1,\dots,x_n)$.

5.8.1 A_1, ..., $A_n(x_1,\dots,x_{n-1})$ and V. If n = 0 the answer is yes and the required derivation is

$$V \in V$$

and if $n > 0$ the answer is yes if and only if the answer is yes for A_1, ..., $A_{n-1}(x_1,\dots,x_{n-2})$ and $A_n(x_1,\dots,x_{n-1})$, and, in addition, the type symbol of $A_n(x_1,\dots,x_{n-1})$ is intentionally equal to V, which can be checked mechanically according to the previous corollary.

5.8.2 A_1, ..., $A_n(x_1,\dots,x_{n-1})$ and x_i. The answer is yes if and only if the answer is yes for A_1, ..., $A_{n-1}(x_1,\dots,x_{n-2})$ and $A_n(x_1,\dots,x_{n-1})$, and, in addition, the type symbol of $A_n(x_1,\dots,x_{n-1})$ is intentionally equal to V.

5.8.3 A_1, ..., $A_n(x_1,\ldots,x_{n-1})$ and $(\Pi x \in A(x_1,\ldots,x_n))$ $B(x_1,\ldots,x_n,x)$. The answer is yes if and only if the answer is yes for A_1, ..., $A_n(x_1,\ldots,x_{n-1})$, $A(x_1,\ldots,x_n)$ and $B(x_1,\ldots,x_n,x)$, and, in addition, the type symbol of $B(x_1,\ldots,x_n,x)$ is intentionally equal to V.

5.8.4 A_1, ..., $A_n(x_1,\ldots,x_{n-1})$ and $(\lambda x \in A(x_1,\ldots,x_n))$ $b(x_1,\ldots,x_n,x)$. The answer is yes if and only if the answer is yes for A_1, ..., $A_n(x_1,\ldots,x_{n-1})$, $A(x_1,\ldots,x_n)$ and $b(x_1,\ldots,x_n,x)$.

5.8.5 A_1, ..., $A_n(x_1,\ldots,x_{n-1})$ and $b(x_1,\ldots,x_n)a(x_1,\ldots,x_n)$. The answer is yes if and only if the answer is yes for A_1, ..., $A_n(x_1,\ldots,x_{n-1})$ and $a(x_1,\ldots,x_n)$ as well as for A_1, ..., $A_n(x_1,\ldots,x_{n-1})$ and $b(x_1,\ldots,x_n)$, and, in addition, the normal form of the type symbol of $b(x_1,\ldots,x_n)$ is of the form $(\Pi x \in A(x_1,\ldots,x_n))B(x_1,\ldots,x_n,x)$ where $A(x_1,\ldots,x_n)$ is the normal form of the type symbol of $a(x_1,\ldots,x_n)$.

5.9 Corollary. For two closed terms a and A it can be mechanically decided whether or not $a \in A$ is derivable. In particular, it can be mechanically decided whether or not a closed term is a type symbol.

Proof. $a \in A$ is derivable if and only if the type symbol of a is intentionally equal to A, and, according to corollary 5.7, it can be mechanically decided whether or not this is so.

5.10 The combinatorial model. It follows from the preceding three corollaries that the species of closed normal terms is mechanically decidable and that it forms a model for the theory of types. In this model definitional equality = is interpreted as syntactical identity neglecting differences in the naming of bound variables, which is clearly a mechanically decidable relation, and $a \in A$ is interpreted to mean that it is derivable in the theory which is mechanically decidable by corollary 5.9. Moreover, for every object in the model, that is, for every closed normal term, it can be mechanically decided if it is the interpretation of a type or a function. And, in the latter case, given an arbitrary other object in the model, it can be mechanically decided if it is applicable to that object in which case the result of the application is obtained by juxtaposition followed by reduction to normal form, which is also a mechanical operation.

Although the theory of types that we are considering has a combinatorial model of the kind just described, the insight that this is so cannot be gained without using abstract (non combinatorial) notions. This is in entire agreement with the philosophical position of Gödel's 1958 paper in Dialectica. In our case the abstract notions used are essentially of the kind that are formalized in the theory itself.

5.11 Completeness of first order intuitionistic logic. Consider a first order formula C containing no other logical constants than $\rightarrow$ and $\forall$, and let f, ... and P, ... be the function and predicate symbols which occur in it. Let C^* denote its translation into the theory of types as defined in section 4.3,

thereby choosing the variable X with type symbol V as the translation of the type of individuals.

Theorem. There is a closed term with type symbol

$$(\Pi X \in V)(\Pi f^* \in X \to \ldots \to X \to X)\ldots(\Pi P^* \in X \to \ldots \to X \to V)\ldots C^*$$

if and only if the formula C is provable in first order intuitionistic predicate logic.

Since the type symbol mentioned in the theorem denotes the proposition that the formula C is true in all models, this may be regarded as an intuitionistic completeness theorem for (a fragment of) first order intuitionistic predicate logic. More precisely, the theorem says that first order intuitionistic predicate logic is complete relative to the theory of types. Kreisel 1968 has suggested to call this property faithfulness rather than completeness in order to avoid misunderstanding.

Proof. One way is simple. Suppose namely that C is provable in first order intuitionistic logic and let c denote its proof. In section 4.3.5.3 we saw how to obtain c^* and a derivation of $c^* \in C^*$ from the assumptions $X \in V$, $f^* \in X \to \ldots \to X \to X$, ..., $P^* \in X \to \ldots \to X \to V$, ... Consequently,

$$(\lambda X \in V)(\lambda f^* \in X \to \ldots \to X \to X)\ldots(\lambda P^* \in X \to \ldots \to X \to V)\ldots c^*$$

is a closed term with type symbol

$$(\Pi X \in V)(\Pi f^* \in X \to \ldots \to X \to X)\ldots(\Pi P^* \in X \to \ldots \to X \to V)\ldots C^*.$$

Conversely, suppose there is a closed term with this type symbol. According to the Hauptsatz, it reduces to a normal term. This normal term must be of the form

$$(\lambda X \in V)(\lambda f^* \in X \to \ldots \to X \to X)\ldots(\lambda P^* \in X \to \ldots \to X \to V)\ldots c^*$$

where c^* is a normal term with type symbol C^*. That c^* is the translation of a derivation c of the formula C in first order intuitionistic logic, which is all that there remains for us to prove, is a consequence of the following lemma.

5.11.1 Lemma. Let C be a first order formula and suppose that c^* is a normal term with type symbol C^* which contains no other variables than X with type symbol V, function variables with type symbols of the form $X \to \ldots \to X \to X$, predicate variables with type symbols of the form $X \to \ldots \to X \to V$, individual variables with type symbol X, and, finally, variables with type symbols of the form A^* where A is a first order formula. Then there is a derivation c of the formula C in first order intuitionistic logic whose translation is c^*.

Proof. By induction on the construction of c^*. Three cases have to be distinguished.

5.11.1.1 c^* and C^* are of the form $(\lambda x \in X)b^*(x)$ and $(\Pi x \in X)B^*(x)$, respectively. By induction hypothesis there is a derivation

$$\begin{array}{c} \vdots \\ B(x) \end{array}$$

in first order logic whose translation is $b^*(x)$. Consequently, we can take c to be the derivation

$$\dfrac{\begin{array}{c} \vdots \\ B(x) \end{array}}{\forall x B(x)}$$

5.11.1.2 c^* and C^* are of the form $(\lambda x \in A^*)b^*(x)$ and $A^* \to B^*$, respectively. By induction hypothesis there is a derivation

$$\begin{array}{c} A \\ \vdots \\ B \end{array}$$

in first order logic whose translation is $b^*(x)$. Consequently, we can take c to be the derivation

$$\dfrac{\begin{array}{c}[A] \\ \vdots \\ B\end{array}}{A \to B}$$

5.11.1.3 c^* is of the form $ya_1^* \ldots a_n^*$. This is not possible unless y has type symbol of the form B^* where B is a formula of first order logic. But then the type symbol of a_i^* must be either X or of the form A_i^* where A_i is a first order formula. Being normal, a_i^* must be the translation of, in the first case, an individual term t and, in the second case, a derivation a_i of the formula A_i in first order logic. Consequently, c^* is the translation of the derivation c which is obtained by letting the assumption B be followed by a sequence of elimination inferences, in the first case, a $\forall$elimination with t in the conclusion and, in the second case, an application of modus ponens with A_i as minor premise.

A. Church

1932 A set of postulates for the foundation of logic, Ann. of Math., vol. 33, pp. 346-366.

1933 A set of postulates for the foundation of logic (second paper), Ann. of. Math., vol. 34, pp. 839-864.

1941 The calculi of lambda-conversion, Ann. of Math. Studies, no. 6, Princeton University Press, Princeton, N. J.

A. Church and J. B. Rosser

1936 Some properties of conversion, Trans. Amer. Math. Soc., vol. 39, pp. 472-482.

H. B. Curry and R. Feys

1958 Combinatory Logic, Vol. I, North-Holland, Amsterdam.

G. Gentzen

1934 Untersuchungen über das logische Schliessen, Math. Z., vol. 39, pp. 176-210 and 405-431.

J. Y. Girard

1970 Une extension de l'interprétation de Gödel à l'analyse, et son application à l'élimination des coupures dans l'analyse et la théorie des types, pp. 63-92 in Proceedings of the Second Scandinavian Logic Symposium, edited by J. E. Fenstad, North-Holland, Amsterdam, 1971.

K. Gödel

1958 Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes, Dialectica, vol. 12, pp. 280-287.

W. A. Howard

1969 The formulae-as-types notion of construction, privately circulated notes.

G. Kreisel

1960 Foundations of intuitionistic logic, pp. 198-210 in Logic, Methodology and Philosophy of Science, edited by E. Nagel, P. Suppes and A. Tarski, Stanford University Press, Stanford, California, 1962.

1968 Church's thesis: a kind of reducibility axiom for constructive mathematics, pp. 121-150 in Intuitionism and Proof Theory, edited by J. Myhill, A. Kino and R. E. Vesley, North-Holland, Amsterdam, 1970.

P. Martin-Löf

1970 Hauptsatz for the simple theory of finite types, to appear.

1970a A construction of the provable wellorderings of the theory of species, to appear.

D. Prawitz

1965 Natural Deduction, Almqvist & Wiksell, Stockholm.

B. Russell

1903 The Principles of Mathematics, Vol. I, Cambridge University Press, Cambridge.

1908 Mathematical logic as based on the theory of types, Amer. J. Math., vol. 30, pp. 222-262.

M. Schönfinkel

1924 Über die Bausteine der mathematischen Logik, Math. Ann., vol. 92, pp. 305-316.

G. Takeuti

1953 On a generalized logic calculus, Japan. J. Math., vol. 23, pp. 39-96.

A. N. Whitehead and B. Russell

1910 Pricipia Mathematica, Vol. I, Cambridge University Press, Cambridge.