Archiv math. Logik 19 (1978), 139—156

A NEW TYPE ASSIGNMENT FOR λ-TERMS*

M. Coppo and M. Dezani-Ciancaglini

Abstract

In the present paper we propose a new type assignment for λ-terms whose motivation is to introduce a system with simple inferential rules to study termination (i.e. the property of having a normal form) of λ-terms. The main results that will be proved in this paper are:

a) all λ-terms in normal form possess a type,
b) all λ-terms which possess a type reduce to normal form.

1. Introduction

In recent years many type assignments to terms of the λ-calculus have been studied. An exhaustive review of these theories, in the context of the logical type theory [7], is given in [12, pp. 1—14]. In all these theories the functional character of λ-terms is generalized by representing types with more powerful objects than that ones introduced in the basic theory of functionality [7, Chapter 9].
In our theory no such object is required but an equivalence and a partial order relation between types are introduced with the assumption that each λ-term which possesses a type τ possesses also all types equivalent or lower than τ.
The aim of the present type assignment is to introduce a powerful method (suitable also for mechanical implementation) for studying the property for a λ-term to possess a normal form (n.f.). In fact the basic result of this paper is that the property of having a type for a λ-term implies that this term and all its subterms possess n.f. Moreover the set of types that can be assigned to a λ-term in n.f. is proved to be always decidable.
The set of types is built recursively from two atomic elements, 0 and 1. Type 0 represents the property for a λ-term to have a n.f., while type 1 represents the property for a λ-term to be such that its applications to an arbitrary number of λ-terms possessing n.f., possesses a n.f. too. The other types are built from 0 and 1 by the usual operation of composition (represented by object F in [7]). We observe that the functional character represented by the fundamental types 0 and 1 (and, consequently, by all other types) can be interpreted only in terms of

* Eingegangen am 27.7.1977.

termination properties. So, if a λ-term X possesses type (0)0[1] this means that, if Y is a λ-term which possesses type 0 (i.e. Y possesses a n.f.) then XY too possesses type 0, i.e. XY reduces to a n.f.
The informal description of the properties represented by types 0 and 1 suggests some considerations about their relations.

First we notice that if X is a λ-term whose applications to arbitrary λ-terms possessing n.f. possesses a n.f. too and Y is a λ-term which possesses a n.f., then XY has the same property as X. It turns out that if X is a λ-term with type 1 and Y is a λ-term with type 0, XY too must be a λ-term with type 1. This fact can be formalized by postulating that types 1 and (0)1 are, in effect, equivalent, i.e. they represent the same property. In a similar way, as it will be shown in the paper, we can justify the assumption that (1)0 is equivalent to 0.

Secondly, it is evident that the property represented by type 1 is much more restrictive than that one represented by type 0. Then we can assume that each λ-term which possesses type 1 possesses also type 0. This property, which will be formalized by the relation $0 \sqsubset 1$, can be generalized in a natural way and leads to the introduction of a partial order relation between types. We can so postulate that, if a given λ-term possesses a type τ, it possesses also all types which are lower than τ. In the paper it will be proved that all these assumptions lead to a consistent system.

An advantage of "despoiling" types of any meaning not related to termination properties is that we can obtain, in this way, a system in which, with a very simple structure of types and type assignment rules, we can assign types also to λ-terms for which this is impossible (or quite troublesome) in most other theories.

For example in the basic theory of functionality [7] it is impossible to assign a type to the combinator $\boldsymbol{W}_* \equiv \lambda x(xx)$ because of the difficulties produced by the application of a variable x to itself. But, with respect to termination properties, it is enough, for example, that x in $\lambda x(xx)$ is replaced by a λ-term which has type (0)0 (i.e. it has a n.f. and when applied to an arbitrary λ-term which has a n.f. produces a λ-term which has a n.f.). So we can deduce immediately that $\boldsymbol{W}_*$ has type ((0)0)0.

Moreover since the termination properties of an application XY depend not only on X itself but also on Y, we can expect that a given λ-term X possesses, in general, more than one type. So, for example, it will be shown that $\boldsymbol{W}_*$ possesses (among infinitely many others) also type (1)1. The intuitive meaning of this type is straightforward. We notice here that if this functional characterization requires a very strong limitation on the possible arguments of $\boldsymbol{W}_*$ (the subset of λ-terms which have type 1 is properly included in subsets of λ-terms which have each other type) it gives also the strongest characterization to the result of the application. Actually termination properties represented by types ((0)0)0 and (1)1 (which are incomparable with respect to the equivalence and partial order rela-

[1] For example λxx has type (0)0.

tions) are quite different and give two different items of information about the λ-term $\lambda x(xx)$.
We have so that all different types that can be assigned to the same λ-term represent different properties of it and give, in a certain sense, the "spectre" of its termination properties.
In this sense our theory "stands" between theories of restricted functionality [7, Chapter 9] (in which a λ-term has only one principal F-scheme [8, pp. 296 and 300]) and the ones of generalized functionality [12] (in which the type of an application XY becomes an explicit function of Y).
The present type assignment to λ-terms is realized by means of a "natural deduction" system as in [7, p. 315]. Two new deduction rules, which are related to the equivalence and partial order relations between types, have been adjoined to the classical rules and axioms [7, p. 317].
We observe that one interest in studying termination properties of λ-terms lies in the fact that λ-calculus can be interpreted as a programming language in which data, programs and results are represented by λ-terms. In this context termination properties of programs can be interpreted as termination properties of λ-terms which represent them. This justifies the interest in a functional characterization of λ-terms with respect to termination, powerful enough to give meaningful results and suitable, at the same time, for a mechanical implementation. In Theorem 3, for example, there is given implicitly an algorithm to determine if a λ-term in n.f. possesses a given type. The complex type structure introduced in the generalized functionality theories [12] would make quite difficult such an implementation.
In Section 7 of this paper we will sketch an application of the present type assignment to determine decidable subsets of n.f.s which build semigroups with respect to the composition combinator $\boldsymbol{B}$[2]. Properties of these semigroups will be a field of further research by the authors.
The introduction of two new deduction rules would not be required if we would develop our type assignment inside Combinatory Logic instead of λ-calculus. In fact, as has been suggested to us by Prof. J. P. Seldin, such a theory could be obtained from $\mathscr{F}_1^b(\boldsymbol{S}, \boldsymbol{K})$[2] of [7, p. 279] by postulating a subject-reduction rule and by adjoining some axioms to represent inclusion and equivalence between types (these axioms will assign types to the identity combinator $\boldsymbol{I}$[2]). As pointed out by the referee, this formulation can be applied to λ-terms as well. Such an approach will be sketched in the last section of the present paper.
The main reason to prefer λ-calculus as the underlaying system for developing our theory is that it is then possible to utilize some previous technical results [3, 4] that could be hardly proved inside Combinatory Logic. Moreover, since

[2] $\boldsymbol{K} \equiv \lambda x \lambda y x$

$\boldsymbol{S} \equiv \lambda x \lambda y \lambda z\, (xz(yz))$

$\boldsymbol{I} \equiv \lambda x x$

$\boldsymbol{B} \equiv \lambda x \lambda y \lambda z\, (x(yz))$.

the authors are mainly interested in the applications of λ-calculus as a model of programming languages, this formulation seems more suitable for studying termination properties of programs.
A classification of n.f.s, with respect to the property of preserving n.f. when applied to suitable n.f.s, has been first introduced, with a quite different methodology, in [3] and then further refined in [9] and [4]. Some properties, proved in these papers, have been used in the proofs of lemmas of the present paper.

2. The Set of Types and Its Properties

The set of types is defined inductively from the atomic types 0 and 1 and the operation of composition.

Definition 1. The set T of T of types is defined by:
a) 0, 1 are types *(atomic types)*,
b) if σ, τ are types then[3] $(\sigma)\tau$ is a type.

In Curry's notation the composition of σ and τ would be written $F\sigma\tau$.
$(\tau_1)\ldots(\tau_{n-1})\tau_n$ will abbreviate $(\tau_1)((\tau_2)(\ldots((\tau_{n-1})\tau_n)\ldots))$.

Example 1. $\tau\equiv(0)(((0)(0)0)0)1$ is a type.

We introduce the following equivalence axioms between types:

$$E_0: 0=(1)0$$
$$E_1: 1=(0)1 .$$

Two types σ and τ are then *equivalent* ($\sigma=\tau$) iff they can be reduced to the same type with a finite number of applications of E_0 and E_1.

Example 2. τ as given in Example 1 and $\sigma\equiv((1)0)((((1)0)(0)0)0)(0)1$ are equivalent. In fact σ can be reduced to τ by applying two times E_0 and one time E_1.

This equivalence relation determines a partition of T into equivalence classes.
It is easy to see that the following facts on type equivalence are true:
1) it is decidable if two types are equivalent.
2) given a type τ, there exists one unique type $\bar{\tau}$ such that $\bar{\tau}=\tau$ and $\bar{\tau}$ has a minimum number of symbols.
Some consequences of previous facts are:
3) each equivalence class in T contains one unique type with a minimum number of symbols (*minimal type* of the class),
4) given a type τ and an integer n, there exist always n types $\tau_1,\ldots,\tau_n$ such that $\tau=(\tau_1)\ldots(\tau_{n-1})\tau_n$.
The length $\|\tau\|$ of an arbitrary type τ is defined as the number of occurrences of atomic types in the minimal type equivalent to τ.

---

[3] The syntax of types is the same as in [6], but the meaning of 0 and 1 in the present paper is quite different from the meaning usually assigned to them.

Example 3. $\|((0)0)1\| = 3$ and $\|((0)1)0\| = 1$ since $((0)1)0 = 0$.

We notice that if $\tau = (\tau_1)\tau_2$ then either τ is equivalent to an atomic type (i.e. $\|\tau\| = 1$) or $\|\tau_1\| < \|\tau\|$ and $\|\tau_2\| < \|\tau\|$.
In our theory all equivalent types define the same object. So, when no particular need arises, we shall represent this object with the minimal type of the corresponding equivalence class.
We introduce now a partial order relation between types defined by $\sqsubseteq$ [4].

Definition 2.
a) $0 \sqsubseteq 1$,
b) let $\tau = (\tau_1)\tau_2$ and $\sigma = (\sigma_1)\sigma_2$:

$$\tau \sqsubseteq \sigma \quad \text{iff} \quad \sigma_1 \sqsubseteq \tau_1 \quad \text{and} \quad \tau_2 \sqsubseteq \sigma_2 .$$

Let us note the inversion, with respect to the order relation, between a type and the leftmost type which builds it. (This inversion is consistent with [7, p. 370] as it will be pointed-out later.)
As usual, $\tau \sqsubset \sigma$ iff $\tau \sqsubseteq \sigma$ and $\tau \neq \sigma$.
From these definitions, moreover, it is clear that it is decidable, for two arbitrary types σ and τ, if $\sigma = \tau$, $\sigma \sqsubset \tau$, $\tau \sqsubset \sigma$ or σ and τ are incomparable.

Example 4. $((0)0)((0)0)0 \sqsubset (0)((0)0)1$.

It may be easily proved that $\sqsubseteq$ is an order relation, i.e. it possesses the reflexive, antisymmetric and transitive properties.
Moreover from Definition 2 we obtain easily the following relations between types[5] which will be widely used later:

$$\forall \tau \in T \quad \text{such that} \quad \tau \neq 0 \;\; 0 \sqsubset \tau ,$$
$$\forall \tau \in T \quad \text{such that} \quad \tau \neq 1 \;\; \tau \sqsubset 1 .$$

3. Type Assignments

In this section we introduce the rules to assign (if it is possible) types to arbitrary λ-terms. This assignment is realized by means of a "natural deduction" system [10, p. 74] with one axiom scheme and four inference rules.
Following [12, p. 35]

(1) $$\sigma X$$

will mean that σ is a type of X.

---

[4] $\sigma \sqsubseteq \tau$ means that τ is included in σ in the sense of [8, p. 453].

[5] As a general result it could be easily proved that the set of types (with the above equivalence and partial order relations) builds a complete lattice [1] in which 0 and 1 are respectively the Bottom and Top elements. The l.u.b. and g.l.b. of two types are so inductively defined: l.u.b. $[\sigma, \tau] = ($g.l.b. $[\sigma_1, \tau_1])$ l.u.b. $[\sigma_2, \tau_2]$, g.l.b. $[\sigma, \tau] = ($l.u.b. $[\sigma_1, \tau_1])$ g.l.b. $[\sigma_2, \tau_2]$, where $\sigma = (\sigma_1)\sigma_2$ and $\tau = (\tau_1)\tau_2$.

In the literature [7, p. 281] a basis is any set of statements of the form (1) which function as axioms for particular deductions. Here we force the statements of a basis to be all of the form σx, where x is a variable. Moreover, in a given basis, no two different types can be assigned to the same variable. $\mathscr{B}\vdash\sigma X$ means that σX is deducible from the statements of $\mathscr{B}$ (in particular, if $\mathscr{B}$ is empty we write $\vdash\sigma X$) by the given axioms and rules. If $\mathscr{B}\vdash\sigma X$ and $\mathscr{B}\subseteq\mathscr{B}'$, then also $\mathscr{B}'\vdash\sigma X$. Obviously $\mathscr{B}\vdash\sigma X$ only if in $\mathscr{B}$ there are type assignments for all variables which occur free in X.
The axiom scheme and rules of our system are the following ones:

AXIOM (Fp). If $\mathscr{B}$ is a basis which contains σx
then $\mathscr{B}\vdash\sigma x$.

RULE (Fe). If $\mathscr{B}\vdash(\sigma)\tau X$
and $\mathscr{B}\vdash\sigma Y$
then $\mathscr{B}\vdash\tau(XY)$.

RULE (Fi). If $\mathscr{B}, \sigma x\vdash\tau X$ and x does not occur in $\mathscr{B}$
then $\mathscr{B}\vdash(\sigma)\tau\ \lambda xX$.

RULE (Ev). If $\mathscr{B}\vdash\sigma X$
and $\tau=\sigma$
then $\mathscr{B}\vdash\tau X$.

RULE (Le). If $\mathscr{B}\vdash\sigma X$
and $\tau\sqsubset\sigma$
then $\mathscr{B}\vdash\tau X$.

Axiom Fp and rules Fe, Fi are usual in the classical theories of types [7, p. 317]. Rule Ev gives a meaning to the equivalence between types introduced by axioms E_0 and E_1. Thanks to this rule an atomic type can reduce to a non atomic one and therefore these types are not strictly atomic in the sense of [7, p. 365]. Through rule Le the type assignment is connected with the relation $\sqsubset$. This rule renders our type theory non-$\boldsymbol{I}$-consistant in the sense of [7, p. 348], i.e. we can derive $\vdash(\sigma)\tau\boldsymbol{I}$ for any types σ, τ such that $\tau\sqsubset\sigma$. This is not surprising since any $\boldsymbol{I}$-consistant theory is unable to express an order relation between types (see [7, p. 349]).
We notice that since all classical inference rules are still valid in our system, all functional characters of λ-terms derivable in the basic system $\mathscr{F}_1^b(\lambda)$ [7, p. 310] are also derivable in our system provided that we interpret type variables as types for our system.
To avoid having infinite deductions, we don't allow consecutive applications of rules Ev and Le. This is not restrictive since we prove easily the following:

Lemma 1. *Two or more consecutive applications of* Ev *and* Le *may always be replaced by one application of* Ev *or* Le.

Proof. The property follows immediately observing that $=$ and $\sqsubset$ are transitive relations and moreover, if σ, ϱ, τ are types:

$$\sigma = \varrho \sqsubset \tau \quad \text{implies} \quad \sigma \sqsubset \tau$$
$$\sigma \sqsubset \varrho = \tau \quad \text{implies} \quad \sigma \sqsubset \tau . \quad \square$$

The type that can be assigned to an arbitrary λ-term is, generally, not unique. In particular, it is possible to assign, to the same λ-term, also types incomparable with respect to $\sqsubseteq$. Moreover, as it will be shown, to each n.f. can be assigned at least one type.

In the deduction $\mathscr{B} \vdash \tau X$ we will assume that the labels of the bound variables of X are all different and they are also different from the labels of the variables which occur in $\mathscr{B}$. This condition may be always satisfied by α-reducing X.

Example 5. We show one deduction of $\vdash (1)0\lambda xx$.

$$\not a . 1x$$
$$\dfrac{\dfrac{\dfrac{a}{1x}\,\mathrm{Fp}}{0x}\,\mathrm{Le}}{(1)0\lambda xx}\,\text{Fi-}a\,.$$

We notice that $\vdash (1)0\boldsymbol{I}$ would be an axiom in the formulation of our type system inside Combinatory Logic (see Section 8).

Example 6. We show one deduction of $\vdash ((0)0)0\,\lambda x(xx)$.

$$\not a \cdot (0)0x$$
$$\dfrac{\dfrac{\dfrac{a}{(0)0x}\,\mathrm{Fp} \qquad \dfrac{\dfrac{a}{(0)0x}\,\mathrm{Fp}}{0x}\,\mathrm{Le}}{0(xx)}\,\mathrm{Fe}\,.}{((0)0)0\,\lambda x(xx)}\,\text{Fi-}a$$

Starting from the canceled premise $1x$ we can deduce, in the same way, $\vdash (1)1\lambda x(xx)$. We notice that these two types are incomparable. Clearly other types could be assigned to $\lambda x(xx)$.

We need now prove (Theorem 1) that when X is different from a single variable $\mathscr{B} \vdash \tau X$ implies that some types can be assigned to the (immediate) subterms of X. Lemma 2 is preliminary to Theorem 1.

Lemma 2. *If X is a λ-term different from a single variable and $\mathscr{B} \vdash \tau X$ then there exists a deduction of it in which the last applied rule is different from* Fp *or* Le.

Proof. Since τX cannot belong to $\mathscr{B}$, the last applied rule in $\mathscr{B} \vdash \tau X$ cannot be Fp. Let us consider a deduction of $\mathscr{B} \vdash \tau X$ such that the last applied rule is Le. The last but one step in this deduction cannot be one application of Ev or Le since we do not allow neither two consecutive applications of rule Le nor one application of Ev followed by one application of Le. The last but one step in this deduction of $\mathscr{B} \vdash \tau X$ cannot be one application of Fp since otherwise $\mathscr{B}$ would contain σX with $\tau \sqsubset \sigma$. Therefore we split this proof according to the last but one applied rule is Fe or Fi.

Case 1. If $X \equiv (YZ)$ the last two steps of this deduction look as:

$$\dfrac{\dfrac{(\varrho)\sigma Y \qquad \varrho Z}{\sigma(YZ)}\,\text{Fe}}{\tau(YZ)}\,\text{Le},$$

where ϱ, σ are types and $\tau \sqsubset \sigma$. We can replace these steps by the following two steps:

$$\dfrac{\dfrac{(\varrho)\sigma Y}{(\varrho)\tau Y}\,\text{Le} \qquad \varrho Z}{\tau(YZ)}\,\text{Fe}.$$

In this way we obtain a deduction which satisfies the desired condition.

Case 2. If $X \equiv \lambda x Y$ the last two steps of this deduction look as:

$$\dfrac{\dfrac{\varrho Y}{(\sigma)\varrho\, \lambda x Y}\,\text{Fi-}\{\sigma x\}}{\tau\, \lambda x Y}\,\text{Le},$$

where σ, ϱ are types and $\tau \sqsubset (\sigma)\varrho$. Let be $\tau = (\tau_1)\tau_2$, we must have $\sigma \sqsubseteq \tau_1$ and $\tau_2 \sqsubseteq \varrho$. If $\sigma x \vdash \varrho Y$ then $\tau_1 x \vdash \varrho Y$ since $\sigma \sqsubseteq \tau_1$ and we can deduce σx from $\tau_1 x$ by applying Le or Ev. From ϱY we can deduce $\tau_2 Y$ by Le or Ev since $\tau_2 \sqsubseteq \varrho$. Then $\tau_1 x \vdash \tau_2 Y$. Lastly we can deduce $\tau \lambda x Y$ from $\tau_1 x \vdash \tau_2 Y$ in the following way:

$$\dfrac{\dfrac{\tau_2 Y}{(\tau_1)\tau_2 \lambda x Y}\,\text{Fi-}\{\tau_1 x\}}{\tau \lambda x Y}\,\text{Ev},$$

where the last application of Ev is useless when $\tau \equiv (\tau_1)\tau_2$. In this way we obtain a deduction which satisfies the desired condition. □

Theorem 1. *Let X be a λ-term different from a single variable, and $\mathscr{B} \vdash \tau X$. Then:*
a) *if $X \equiv YZ$ then there exists a type σ such that $\mathscr{B} \vdash (\sigma)\tau Y$ and $\mathscr{B} \vdash \sigma Z$,*
b) *if $X \equiv \lambda x Y$ and $\tau = (\sigma)\varrho$ then $\mathscr{B}, \sigma x \vdash \varrho Y$.*

Proof. By Lemma 1 we can consider a deduction of $\mathscr{B} \vdash \tau X$ in which the last applied rule is Fe, Fi or Ev.

The first two cases coincide with the corresponding ones of the classical theory. In the third case the last step looks as:

$$\frac{\nu X}{\tau X}\,\mathrm{Ev}\,,$$

where $\nu = \tau$. Since we don't allow two consecutive applications of Le and Ev and νX cannot belong to $\mathscr{B}$, the last but one step must be one application of Fe or Fi. Then the theorem holds for ν and, since $=$ is a transitive relation, also for τ. □

Theorem 1 proves that rules Fe and Fi are invertible. We notice that rule Le cannot be reversed, i.e. $\mathscr{B} \vdash \tau X$ doesn't imply that there exists a type σ such that $\tau \sqsubset \sigma$ and $\mathscr{B} \vdash \sigma X$.
Many of the proofs of the following theorems and lemmas are performed by induction on the number of variable occurrences $|X|$ in a λ-term X.
The inductive definition of $|X|$ is:
a) if $X \equiv x$ then $|X| = 1$,
b) if $X \equiv YZ$ then $|X| = |Y| + |Z|$,
c) if $X \equiv \lambda x Y$ then $|X| = 1 + |Y|$.

4. Properties of Type Assignments to n.f.s

In this section we prove some results on the type assignments to n.f.s. More specifically we will prove that:
a) each n.f. has type 0 if we assign type 1 to all its free variables (Theorem 2),
b) given a n.f. N, a basis $\mathscr{B}$ and a type τ, $\mathscr{B} \vdash \tau N$ is decidable (Theorem 3).

Theorem 2. *If N is a* n.f. *and $\mathscr{B}$ is a basis which assigns type 1 to all free variables of N, then $\mathscr{B} \vdash 0N$.*

Proof. By induction on $|N|$.

First Step. $|N| = 1$, i.e. N is a free variable (say x). Then by hypothesis $1x \in \mathscr{B}$. Therefore:

$$\mathscr{B} \vdash 1x \quad (\text{by Fp}) \text{ and } \quad \mathscr{B} \vdash 0x \quad (\text{by Le}).$$

Inductive Step. We split the proof according to two possible cases:
1) $N \equiv x N_1 \ldots N_m \quad (m > 0)$.
Since $|N_i| < |N|$ for $1 \leqq i \leqq m$ then $\mathscr{B} \vdash 0 N_i$ by inductive hypothesis. $1x \in \mathscr{B}$ since x occurs free is N. Taking into account that $1 = \underbrace{(0) \ldots (0)}_{\text{m times}}1$, we obtain $\mathscr{B} \vdash 1N$ by one application of Ev and m applications of Fe. Lastly $\mathscr{B} \vdash 0N$ by Le.
2) $N \equiv \lambda x \bar{N}$.
Since $|\bar{N}| < |N|$ then $\mathscr{B}, 1x \vdash 0\bar{N}$ by inductive hypothesis. Therefore $\mathscr{B} \vdash 0N$ by Fi and Ev [we recall that $(1)0 = 0$]. □

Example 7. We show a deduction of $\vdash 0\lambda x(xx)$.

$$\not{a}.1x$$

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{a}{1x}\,\text{Fp}}{(0)1x}\,\text{Ev} \qquad \cfrac{\cfrac{a}{1x}\,\text{Fp}}{0x}\,\text{Le}}{1(xx)}\,\text{Fe}}{0(xx)}\,\text{Le}}{\cfrac{(1)0\lambda x(xx)}{0\lambda x(xx)}\,\text{Ev}}\,\text{Fi-}a$$

Theorem 3. *If N is a* n.f., *then $\mathscr{B}\vdash\tau N$ is decidable.*

Proof. By induction on $|N|$.

First Step. $|N|=1$, i.e. N is a free variable (say x). In this case $\mathscr{B}\vdash\tau x$ iff $\sigma x\in\mathscr{B}$ and $\tau\sqsubseteq\sigma$.

Inductive Step. We split the proof according to two possible cases:
1) $N\equiv xN_1...N_m$. In this case $\mathscr{B}$ must contain the statement σx. Let $\sigma=(\sigma_1)\ldots(\sigma_m)\sigma_{m+1}$, then by Theorem 1 we have:

$$\mathscr{B}\vdash\tau N \quad \text{iff} \quad \mathscr{B}\vdash\sigma_i N_i \quad \text{for} \quad 1\leqq i\leqq m \quad \text{and} \quad \tau\sqsubseteq\sigma_{m+1}\,.$$

$\mathscr{B}\vdash\sigma_i N_i$ for $1\leqq i\leqq m$ is decidable by inductive hypothesis and $\tau\sqsubseteq\sigma_{m+1}$ is decidable.
2) $N\equiv\lambda x\bar{N}$. Let be $\tau=(\tau_1)\tau_2$. By Theorem 1

$$\mathscr{B}\vdash\tau N \quad \text{iff} \quad \mathscr{B},\tau_1 x\vdash\tau_2\bar{N}\,.$$

$\mathscr{B},\tau_1 x\vdash\tau_2\bar{N}$ is decidable by inductive hypothesis. □

Example 8. We prove that $\vdash(1)1\,\lambda x(xx)$ but $\nvdash(0)0\,\lambda x(xx)$.
$\vdash(1)1\lambda x(xx)$ iff $1x\vdash 1(xx)$ (by Fi). Since $1=(0)1$, $1x\vdash 1(xx)$ iff $1x\vdash 0x$ (by Fe) then $\vdash(1)1\lambda x(xx)$.
$\vdash(0)0\lambda x(xx)$ iff $0x\vdash 0(xx)$ (by Fi). Since $0=(1)0$, $0x\vdash 0(xx)$ iff $0x\vdash 1x$ (by Fe) which is not true since Le cannot be reversed.

5. Properties of n.f.s Possessing Type 1

In this section, we will study the set of n.f.s which possesses type 1. By Le these n.f.s possess any type. To this aim, we introduce here the notion of replaceable variables as given in [3, 9, 4].

Lemma 3 relates the replaceable variables of a n.f. N with their types in one deduction of $\mathscr{B} \vdash 1N$. Then we define the set $\mathscr{N}_\omega$ of n.f.s as in [3, 9, 4] and we prove that $\mathscr{N}_\omega$ contains the set of n.f.s which possess type 1[6] (Lemma 4).
This permits us to use Theorem 2 of [3] and Lemma 1 of [4] to assure that if $\mathscr{B} \vdash 1\mathscr{N}$ and N, M are n.f.s, then the applications NM and MN possess n.f. too (Theorem 4).

Definition 3. In a n.f. $N \equiv \lambda x_1 \ldots \lambda x_n (\mathfrak{z} N_1 \ldots N_m)$:
a) $x_1, \ldots, x_n$ are replaceable,
b) the free variables are non-replaceable,
c) if $M \equiv z M_1 \ldots M_p$ is a subterm of N and z is (non-)replaceable then all the variables bound in the initial abstractions of M_p $(p \geqq 1)$ are (non-)replaceable.

According to Definition 3 each variable which occurs in a n.f. is either replaceable or non-replaceable in this n.f. (refinements of this classification have been done in [3, 9, 4].)

Example 9. In the n.f. $N \equiv \lambda x_1 \lambda x_2 (u \lambda x_3 (x_2 \lambda x_4 (x_3 x_4)))$:
a) x_1, x_2, and x_4 are replaceable,
b) u and x_3 are non-replaceable.

Lemma 3. *If N is a* n.f. *and* (2) $\mathscr{B} \vdash 1N$, *then there exists a deduction of* (2) *in which to all replaceable variables of N is assigned type* 0.

Proof. We prove this Lemma by induction on the number r of applications of rule c) of Definition 3.

First Step. $r = 0$. i.e. we consider a variable x bound in the initial abstractions of N.
Let $N \equiv \lambda y_1 \ldots \lambda y_s \lambda x \bar{N}$.
By $s+1$ applications of Theorem 1 to $\mathscr{B} \vdash 1N$ we obtain:

$$\mathscr{B}, 0y_1, \ldots, 0y_s, 0x \vdash 1\bar{N}.$$

Inductive Step. Let this Lemma be true for $r \leqq v$; we prove it for $r = v+1$. Let $M \equiv z M_1 \ldots M_p$ be a subterm of N, z be a replaceable variable and x be bound in the initial abstractions of M_p $(p \geqq 1)$.
By inductive hypothesis we have the statement $0z$ in one deduction of (2). Since $0 = \underbrace{(1)\ldots(1)}_{p\,\text{times}}0$, by rules Ev and Fe we must have $\mathscr{B}' \vdash 1M_p$ for some basis $\mathscr{B}'$. If $M_p \equiv \lambda y_1 \ldots \lambda y_s \lambda x \bar{M}_p$ then by $s+1$ applications of Theorem 1 to $\mathscr{B}' \vdash 1M_p$ we have $\mathscr{B}', 0y_1, \ldots, 0y_s, 0x \vdash 1\bar{M}_p$. □

[6] Really these two sets coincide, but we omit here the proof that each n.f. which belongs to $\mathscr{N}_\omega$ possesses type 1.

Example 10. We consider the deduction: $1u \vdash 1N$ where N is the n.f. of Example 9.

$$\not{a}.0x_1 \quad \not{b}.0x_2 \quad c.1u \quad \not{d}.1x_3 \quad \not{e}.0x_4$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{c}{1u}\,\mathrm{Fp}}{(0)1u}\,\mathrm{Ev} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{b}{0x_2}\,\mathrm{Fp}}{(1)0x_2}\,\mathrm{Ev} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{d}{1x_3}\,\mathrm{Fp}}{(0)1x_3}\,\mathrm{Ev} \qquad \dfrac{e}{0x_4}\,\mathrm{Fp}}{1(x_3x_4)}\,\mathrm{Fe}}{(0)1\lambda x_4(x_3x_4)}\,\mathrm{Fi}\text{-}e}{1\lambda x_4(x_3x_4)}\,\mathrm{Ev}}{0(x_2\lambda x_4(x_3x_4))}\,\mathrm{Fe}}{(1)0\lambda x_3(x_2\lambda x_4(x_3x_4))}\,\mathrm{Fi}\text{-}d}{0\lambda x_3(x_2\lambda x_4(x_3x_4))}\,\mathrm{Ev}}{1(u\lambda x_3(x_2\lambda x_4(x_3x_4)))}\,\mathrm{Fe}}{(0)1\lambda x_2(u\lambda x_3(x_2\lambda x_4(x_3x_4)))}\,\mathrm{Fi}\text{-}b}{(0)\,(0)1\lambda x_1\lambda x_2(u\lambda x_3(x_2\lambda x_4(x_3x_4)))}\,\mathrm{Fi}\text{-}a\,.}{1\lambda x_1\lambda x_2(u\lambda x_3(x_2\lambda x_4(x_3x_4)))}\,\mathrm{Ev}$$

Let $N \equiv \lambda x_1 \ldots \lambda x_n\, (\mathfrak{z} N_1 \ldots N_m)$ be a n.f., we call $\mathfrak{z}$ the *head variable* of N and N_l $(1 \leqq l \leqq m)$ the l-th *component* of N.

Definition 4. A n.f. $N \in \mathcal{N}_\omega$ iff:
1) the head variable of N is free,
2) if the head variable of a proper subterm M of N is replaceable, then all the head variables of the components of M are non-replaceable.

Example 11. If N is as in Example 9, then $N \in \mathcal{N}_\omega$.

Lemma 4. *If N is a* n.f. *such that, for some basis $\mathscr{B}$, $\mathscr{B} \vdash 1N$, then $N \in \mathcal{N}_\omega$.*

Proof. If N is a free variable this Lemma is obviously true. Otherwise we prove that N satisfies Conditions 1 and 2 of Definition 4.

Condition 1. Let be $N \equiv \lambda x_1 \ldots \lambda x_n\, (\mathfrak{z} N_1 \ldots N_m)$. By n applications of Theorem 1 to $\mathscr{B} \vdash 1N$ we obtain:

$$\mathscr{B}, 0x_1, \ldots, 0x_n \vdash 1(\mathfrak{z} N_1 \ldots N_m)\,.$$

Then $\mathfrak{z}$ must have type $(v_1)\ldots(v_m)1$ for suitable values of v_l $(1 \leqq l \leqq m)$. Since for all v_l $(1 \leqq l \leqq m)$ $0 \sqsubset (v_1)\ldots(v_m)1$, $\mathfrak{z}$ cannot coincide with x_i for some i $(1 \leqq i \leqq n)$, i.e. $\mathfrak{z}$ must be a free variable.

Condition 2. Let $M \equiv z M_1 \ldots M_p$ be a proper subterm of N and z be a replaceable variable. We suppose ad absurdum that also the head variable y of M_p be a replaceable variable. By Lemma 3 there exists a deduction in which $0z$ and $0y$. $0z$ implies $1M_p$ by rules Ev and Fe. We cannot have $0y$ and $1M_p$ by the same argument given in the proof of Condition 1. □

Theorem 4. *If* N, M *are* n.f.s *and, for some basis* $\mathscr{B}$, $\mathscr{B} \vdash 1N$ *then* NM *and* MN *possess* n.f.

Proof. By Lemma 4 $N \in \mathscr{N}_\omega$. The proof that NM possesses n.f. is given in Theorem 2 of [3].
The proof that MN possesses n.f. is given in Lemma 1 of [4]. □

6. Properties of λ-Terms Possessing Types

In this section we show that the property of having a type for a λ-term is a sufficient (but not necessary) condition for it to have n.f. To prove this we shall first prove, in Lemmas 5 and 6, that the property of having a type is invariant for β-reductions, i.e., if $\mathscr{B} \vdash \sigma X$ and $X \geqq X'$ then $\mathscr{B} \vdash \sigma X'$.
The most general results are stated in Theorems 5 and 6. Lastly, Corollary 1 proves that all the subterms of a λ-term which possesses a type reduce to n.f.

Lemma 5. *If* R *is a* β*-redex,* R' *is its contractum, then* $\mathscr{B} \vdash \tau R$ *implies* $\mathscr{B} \vdash \tau R'$.

Proof. Let $R \equiv \lambda x Y Z$. By Theorem 1 there exists a type σ such that in one deduction of $\mathscr{B} \vdash \tau R$ we have $\mathscr{B} \vdash (\sigma)\tau\lambda x Y$, $\mathscr{B} \vdash \sigma Z$ and x doesn't occur in $\mathscr{B}$. Moreover, again by Theorem 1, from $\mathscr{B} \vdash (\sigma)\tau\lambda x Y$ we have $\mathscr{B}, \sigma x \vdash \tau Y$. By definition $R' \equiv Y[x/Z]$. To show that $\mathscr{B} \vdash \tau Y[x/Z]$ it is sufficient to replace, in the proof of $\mathscr{B}, \sigma x \vdash \tau Y$, the statement σx (which is not in $\mathscr{B}$) with the deduction of $\mathscr{B} \vdash \sigma Z$. □

Lemma 6. If $X \geqq X'$ and (3) $\mathscr{B} \vdash \sigma X$ then $\mathscr{B} \vdash \sigma X'$.

Proof. It is enough to prove the property in the case that X' differs from X for the contraction of only one β-redex. The general case follows immediately by iterated applications of this property.
Let R be any β-redex of X. In one deduction of (3) there must be a deduction of $\mathscr{B}' \vdash \tau R$ (where $\mathscr{B}'$ differs from $\mathscr{B}$ because it contains also the type assignments to the variables which are free in R and bound in X). If R' is the contractum of R, by Lemma 5 there is also a deduction of $\mathscr{B}' \vdash \tau R'$. Then we can prove $\mathscr{B} \vdash \sigma X'$ by replacing in one deduction of (3) the deduction of $\mathscr{B}' \vdash \tau R$ by one deduction of $\mathscr{B}' \vdash \tau R'$. □

We are now able to prove the property that applications of n.f.s preserving types reduce to n.f.

Theorem 5. *If* N *and* M *are* n.f.s *and, for some basis* $\mathscr{B}$ *and type* τ, $\mathscr{B} \vdash \tau(NM)$ *then* $NM \geqq N'$ *in* n.f. *and* $\mathscr{B} \vdash \tau N'$.

Proof. By Theorem 1 if $\mathscr{B}\vdash\tau(NM)$ there exists a type σ such that $\mathscr{B}\vdash(\sigma)\tau N$ and $\mathscr{B}\vdash\sigma M$.
The theorem is proved by a double induction on $\|(\sigma)\tau\|$ and $|N|$.

First Step. If $\|(\sigma)\tau\|=1$ then $(\sigma)\tau$ is an atomic type, i.e. we have either $(\sigma)\tau=0$ or $(\sigma)\tau=1$. In the first case $\mathscr{B}\vdash 0N$ and $\mathscr{B}\vdash 1M$, in the second $\mathscr{B}\vdash 1N$ and $\mathscr{B}\vdash 0M$. In both cases the proof is given in Theorem 4.

Inductive Step. Let the property be true for all types ϱ such that $\|\varrho\|<\|(\sigma)\tau\|$. We prove it for $(\sigma)\tau$. We observe that if $\|(\sigma)\tau\|>1$ then $\|\sigma\|<\|(\sigma)\tau\|$.
Here it is sufficient to prove that $NM\geqq N'$ in n.f. since Lemma 6 assures us that $\mathscr{B}\vdash\tau N'$.
If N is not of the form $\lambda x\bar{N}$, then $N'\equiv NM$.
Otherwise we prove the property by induction on $|\bar{N}|$.

First Step. If $|\bar{N}|=1$ we have $\bar{N}\equiv\mathfrak{z}$ where either $\mathfrak{z}\equiv x$ or $\mathfrak{z}$ is a free variable. Then $NM\geqq N'$ where $N'\equiv M$ if $\mathfrak{z}\equiv x$ and $N'\equiv\mathfrak{z}$ otherwise.

Inductive Step. Let $\bar{N}\equiv\lambda y_1\ldots\lambda y_n(\mathfrak{z}N_1\ldots N_m)$ and $\nu_l\ (1\leqq l\leqq m)$ be the types assigned to N_l in one deduction of $\mathscr{B}\vdash(\sigma)\tau N$.
By Theorem 1 and rule Fi there is a deduction of $\mathscr{B}'\vdash(\sigma)\nu_l\lambda xN_l\ (1\leqq l\leqq m)$ where $\mathscr{B}'\equiv\mathscr{B},\varphi_1y_1,\ldots,\varphi_ny_n$ and $\varphi_1,\ldots,\varphi_n$ are such that $\tau=(\varphi_1)\ldots(\varphi_n)\psi$ for some type ψ.
Since $(\sigma)0\sqsubseteq(\sigma)\nu_l\ (1\leqq l\leqq m)$ then by rule Le there is also a deduction of $\mathscr{B}'\vdash(\sigma)0\lambda xN_l$.
Then $\lambda xN_lM\geqq N_l'$ in n.f. by inductive hypothesis since $|N_l|<|\bar{N}|$ and $\|(\sigma)0\|\leqq\|(\sigma)\tau\|$.
Moreover, by Fe, we have also $\mathscr{B}'\vdash\nu_l(\lambda xN_lM)$ and, by Lemma 6, $\mathscr{B}'\vdash\nu_lN_l'$.
Now if $\mathfrak{z}\not\equiv x$, $NM\geqq\lambda y_1\ldots\lambda y_n(\mathfrak{z}N_1'\ldots N_m')$ which is in n.f. since all $N_l'\ (1\leqq l\leqq m)$ are in n.f.
Otherwise $NM\geqq\lambda y_1\ldots\lambda y_1\ (MN_1'\ldots N_m')$ and we must prove that $MN_1'\ldots N_m'$ has a n.f. We notice that if $\sigma=(\sigma_1)\ldots(\sigma_m)\sigma_{m+1}$ we must have $\sigma_l\sqsubseteq\nu_l\ (1\leqq l\leqq m)$ and $\psi\sqsubseteq\sigma_{m+1}$. In this case, since $\mathscr{B}'\vdash\nu_lN_l'$ there is also, by Le or Ev, a deduction of $\mathscr{B}'\vdash\sigma_lN_l'\ (1\leqq l\leqq m)$. Since $\mathscr{B}\vdash\sigma M$ (by hypothesis), $\mathscr{B}'\vdash\sigma_1N_1'$ (from above) and $\mathscr{B}$ is a subset of $\mathscr{B}'$, then by rule Fe: $\mathscr{B}'\vdash(\sigma_2)\ldots(\sigma_m)\sigma_{m+1}(MN_1')$. Therefore, since $\|(\sigma_1)\ldots(\sigma_m)\sigma_{m+1}\|=\|\sigma\|<\|(\sigma)\tau\|$, by inductive hypothesis MN_1' reduces to a n.f. M_1' and, by Lemma 6, $\mathscr{B}'\vdash(\sigma_2)\ldots(\sigma_m)\sigma_{m+1}M_1'$. By repeating m-2 times this argument we can conclude that NM reduces to N' in n.f. □

We can now prove that any λ-term which possesses a type possesses also a n.f.

Theorem 6. *If X is an arbitrary λ-term and, for some basis $\mathscr{B}$ and type τ, $\mathscr{B}\vdash\tau X$ then X reduces to a* n.f. *X' and $\mathscr{B}\vdash\tau X'$.*

Proof. It is enough to prove that X has n.f., since Lemma 6 assures that $\mathscr{B}\vdash\tau X'$.
We perform the proof by induction an $|X|$.

First Step. If $|X|=1$ then X is a free variable. The Theorem is obviously true.

Inductive Step. We distinguish two possible cases.
a) $X \equiv \lambda x Y$. Since $|Y| < |X|$, $Y \geqq Y'$ in n.f. by inductive hypothesis, and therefore $X' \equiv \lambda x Y'$.
b) $X \equiv YZ$. By Theorem 1 $\mathscr{B} \vdash \tau X$ implies that there exists a type σ such that $\mathscr{B} \vdash (\sigma)\tau Y$ and $\mathscr{B} \vdash \sigma Z$. Since $|Y|$, $|Z| < |YZ|$, $Y \geqq Y'$ in n.f. and $Z \geqq Z'$ in n.f. by inductive hypothesis. Moreover, by Lemma 6, $\mathscr{B} \vdash (\sigma)\tau Y'$ and $\mathscr{B} \vdash \sigma Z'$. Then, by Theorem 5, $Y'Z' \geqq X'$ in n.f. □

Corollary 1. If X is a λ-term and $\mathscr{B} \vdash \tau X$ then all subterms of X possess n.f.

Proof. Immediate from the definition of deduction and Theorem 6. In fact if $\mathscr{B} \vdash \tau X$ and Y is an arbitrary subterm of X, there exists a type σ and a basis $\mathscr{B}'$ such that $\mathscr{B}' \vdash \sigma Y$. Then, by Theorem 6, Y possesses n.f. □

Let us notice, here, that Theorem 6 can be viewed as a proof of the consistency of the whole theory.

7. Semigroups of n.f.s

In [5], Church proves that the set of λ-terms builds a semigroup with respect to the composition combinator $\boldsymbol{B} \equiv \lambda x \lambda y \lambda z(x(yz))$. This is not true for the set of n.f.s since the composition of two arbitrary n.f.s can have no n.f. In the present section we give (through Theorem 7) a sufficient condition to characterize decidable sets of n.f.s which build semigroups with respect to $\boldsymbol{B}$.

Theorem 7. *Let σ, τ be two types such that $\sigma \sqsubseteq \tau$ and $\mathscr{S}[(\sigma)\tau]$ be a set of* n.f.s *defined by:*

$$\mathscr{S}[(\sigma)\tau] = \{N | (\sigma)\tau N\} .$$

Then $\mathscr{S}[(\sigma)\tau]$ builds a semigroup with respect to $\boldsymbol{B}$.

Proof. Let N, M be two n.f.s and

$$(\sigma)\tau N$$
$$(\sigma)\tau M .$$

It is sufficient to prove that $\boldsymbol{B}MN$ possesses the type $(\sigma)\tau$. Since (also in the basic functionality theory) $((\beta)\gamma)((\alpha)\beta)(\alpha)\gamma \boldsymbol{B}$[7], where α, β, and γ are type variables, we have the desired deduction with the choice $\alpha = \beta = \sigma$ and $\gamma = \tau$, observing that $(\sigma)\sigma \sqsubseteq (\sigma)\tau$. □

We notice that in the particular case $\sigma = \tau$, $\mathscr{S}[(\sigma)\sigma]$ is a monoid, i.e. it contains the identity combinator $\boldsymbol{I} \equiv \lambda x x$. Moreover since $\sigma \sqsubseteq \tau$ implies $(\sigma)\sigma \sqsubseteq (\sigma)\tau$, we

[7] Here we don't work inside Combinatory Logic and $\boldsymbol{B}$ stands only as abbreviation for the corresponding λ-term.

have that, thanks to rule Le, each λ-term which has type $(\sigma)\tau$ has also type $(\sigma)\sigma$ (while the conversely is in general not true). So we have that $\mathscr{S}[(\sigma)\tau]$ is properly included in $\mathscr{S}[(\sigma)\sigma]$.

Example 12. The monoid $\mathscr{S}[(1)1]$ contains all Church n-tuples of n.f.s. In fact from $0X_i$ for $1 \leqq i \leqq n$ we obtain easily $(1)1\lambda z(zX_1 \ldots X_n)$ if z doesn't occur free in $X_1, \ldots, X_n$. This is interesting since, if M and N are two n-tuples, then $\boldsymbol{B}MN$ reduces to their reverse concatenation[8] [2]. Let us notice that, since the components of the n-tuples are supposed to be arbitrary n.f.s, this result could not be achieved in the basic functionality theory.

8. An Equivalent Formulation for Combinatory Logic

As suggested by Prof. J. P. Seldin an equivalent system can be obtained from the basic system $\mathscr{F}_1^b(\lambda)$ or $\mathscr{F}_1(\lambda)^T$ of [7] by adjoining some axioms for the λ-term λxx and by postulating a subject reduction rule for $\lambda\eta$-reduction. The interest of this approach is that it is suitable for giving an equivalent formulation of our system inside Combinatory Logic. We will brevily sketch how this formalization can be obtained from the system $\mathscr{F}_1^b(\boldsymbol{S}, \boldsymbol{K})$ of [7, 8].
An appropriate basis $\mathscr{B}_0$ will be formed by the following axiom schemes and axioms[9]:

(FK) $(\alpha)(\beta)\alpha\boldsymbol{K}$
(FS) $((\alpha)(\beta)\gamma)((\alpha)\beta)(\alpha)\gamma\boldsymbol{S}$
$(1)0\boldsymbol{I}$
$(0)(1)0\boldsymbol{I}$
$((1)0)0\boldsymbol{I}$
$(1)(0)1\boldsymbol{I}$
$((0)1)1\boldsymbol{I}$.

$\mathscr{B}_0$ is clearly non-monoschematic in the sense of [8, p. 300]. Besides the usual rule F [7, p. 279] we introduce a rule R which assigns any type of a combinator U to all combinators V such that U is strongly reducible to V.

Rule F. $(\sigma)\tau U, \sigma V \vdash \tau(UV)$.

Rule R. $\tau U, U \succ V \vdash \tau V$.

The equivalence between the present formulation and that one given in previous sections seems intuitively clear. This equivalence may be formally proved using a technique similar to that one of [7, Theorems 1–3, pp. 318 and 319].
We leave out this proof since it doesn't augment our insight in this type assignment.

[8] I.e. if $M = \langle M_1, \ldots, M_p \rangle$ and $N = \langle N_1, \ldots, N_q \rangle$ then $\boldsymbol{B}MN \geqq \langle N_1, \ldots, N_q, M_1, \ldots, M_p \rangle$, where $\langle X_1, \ldots, X_n \rangle$ denotes the n-tuple $\lambda z\,(zX_1, \ldots, X_n)$.

[9] (FK) and (FS) are axiom schemes and therefore α, β, and γ are type variables.

We only notice here that the partial order relation of Definition 2 can be proved in this system showing that from $(\tau_1)\sigma_1 \boldsymbol{I}$ and $(\sigma_2)\tau_2 \boldsymbol{I}$ we can deduce $((\sigma_1)\sigma_2)(\tau_1)\tau_2 \boldsymbol{I}$. As a consequence of this we obtain that $\boldsymbol{I}$ possesses all types $(\sigma)\tau$ with $\sigma \sqsupseteq \tau$. This property is then consistent with that one proved in Theorem 3 of [7, p. 370]. In fact the object Ξ' introduced there becomes in our case $\lambda x \lambda y((x)y\boldsymbol{I})$.

Conclusion

It has been shown that, given a type τ and a n.f. N, it is decidable if τ can be assigned to N. The question if this property can be proved for an arbitrary λ-term not in n.f. is still open. This would not lead to contradiction with the well known semi-decidability of the halt problem. In fact, in the present type assignment, a subject expansion theorem is valid only for expansions which satisfy conditions similar to those of Theorem 3 [7, p. 298].
In a recent work the authors have proved that a suitable generalization of the present theory (as well as of the basic theory of [7, 8]) will lead to a system in which any two convertible terms possess the same set of types. In this case, obviously, the problem of deciding if a given term possesses a type becomes semi-recursive.

Acknowledgement. We are very grateful to Prof. J. P. Seldin for his criticism and suggestions.

REFERENCES

[1] Birkhoff, G.: Lattice theory. American Mathematical Society Colloquium Publications **25** (1940).
[2] Böhm, C., Groß, W.: Introduction to the CUCH. In: Caianiello, E. R. (ed.): Automata Theory, pp. 35—65. Amsterdam: North-Holland 1966.
[3] Böhm, C., Dezani-Ciancaglini, M.: λ-Terms as total or partial function on normal forms. In: Böhm, C. (ed.): λ-Calculus and computer science theory. Lecture Notes in Computer Science, Vol. 37, pp. 96—121. Berlin, Heidelberg, New York: Springer 1975.
[4] Böhm, C., Coppo, M., Dezani-Ciancaglini, M.: Termination test inside λ-calculus. In: Salomaa, A. and Steinby, M. (eds.): Lecture Notes in Computer Science, Vol. 52, pp. 95—110. Berlin, Heidelberg, New York: Springer 1977.
[5] Church, A.: Combinatory logic as a semigroup. Bull. Amer. Math. Soc. (Abstract) **43**, 333 (1937).
[6] Church, A.: A formalization of the simple theory of types. J. Symbol. Logic **5**, 56—58 (1950).
[7] Curry, H. B., Feys, R.: Combinatory logic, Vol. 1. Amsterdam: North-Holland 1958.
[8] Curry, H. B., Hindley, J. R., Seldin, J. P.: Combinatory logic, Vol. 2. Amsterdam: North-Holland 1972.
[9] Dezani-Ciancaglini, M.: A type theory for λ-β-normal forms. Proc. of the Symposium Informatica **75**, BLED (1975).
[10] Gentzen, G.: Investigations into logical reductions. In: The Collected Papers of Gerard Gentzen. Amsterdam: North-Holland 1969.

[11] Hindley, R., Lercher, B., Seldin, J. P.: Introduction to combinatory logic. London North. Soc. Lecture Note Series **7** (1972).
[12] Seldin, J. P.: The theory of generalized functionality. I. Unpublished manuscript (1976).

Prof. M. Coppo und
Prof. M. Dezani-Ciancaglini
Università di Torino,
Istituto di Scienze dell'informazione,
Corso M. d'Azeglio 42,
10125 Torino
Italien