RAIRO. INFORMATIQUE THÉORIQUE

M. DEZANI-CIANCAGLINI
S. RONCHI DELLA ROCCA
L. SAITTA

# Complexity of λ-term reductions

*RAIRO. Informatique théorique*, tome 13, nº 3 (1979), p. 257-287
<http://www.numdam.org/item?id=ITA_1979__13_3_257_0>

R.A.I.R.O. Informatique théorique/Theoretical Informatics
(vol. 13, n° 3, 1979, p. 257 à 287)

# COMPLEXITY OF λ-TERM REDUCTIONS (*)

by M. DEZANI-CIANCAGLINI,
S. RONCHI DELLA ROCCA and L. SAITTA (1)

Communicated by J.-F. PERROT

Résumé. — *Dans cet article on définit une mesure de la complexité d'un λ-terme comme le nombre de β-réductions nécessaires pour arriver à sa forme normale, si elle existe. On étudie d'abord quelques propriétés générales de la complexité d'applications entre formes normales, ensuite on calcule un maximum pour la complexité dans le cas d'applications entre formes normales appartenant à des classes particulières.*

Abstract. — *The complexity of a λ-term is defined as the number of β-reductions needed to reach its normal form (n. f.). In the present paper some general properties of the complexity of applications of n. f. s. are stated and this complexity is maximized, in some interesting cases, as a function of parameters describing the structures of the current n. f. s.*

## 1. INTRODUCTION

As the very considerable quantity of literature on the subject shows ([13] and [1] are merely the first and last studies published, chronologically speaking), the use of λ-calculus in the study of programming languages is now a classic topic.

Formerly it was usual to consider, in the said approach, only that subset (of the set Λ of all λ-terms) which represents programs and data. Since it has been found that both programs [19] and data [16] can be represented by λ-terms in normal form (n. f.) they are naturally related to a subset of the n. f. set $\mathscr{N}$.

Our starting point is somewhat different, in that we consider the whole Λ as a programming language, whose properties we will examine. The definition of semantics for Λ [15, 17, 18, 20] acts as a support to this approach. Considering Λ as a programming language, the execution of a program, applied to some data, is defined by the β-reduction of a λ-term until one attains its n. f., if it exists. In Λ models too, the set $\mathscr{N}$ is pre-eminent, in that:

— two different n. f. s. have different meanings, otherwise the model becomes inconsistent [20];

(*) Received May 1978, revised January 1979.
(1) Istituto di Scienza dell'Informazione, Università di Torino, Torino, Italia.

— only λ-terms that have n. f. represent finite computations, hence such terms are more significant from the programming point of view [20].

In the present work we will define and compute the "complexity" of applications such as:

$$NM_1 \ldots M_k, \tag{1}$$

where $N$ and $M_i$ $(1 \leqq i \leqq k)$ are n. f. s. By *complexity* we mean the number of β-reductions (= steps) needed to reach the n. f. of (1), if it has a n. f., according to a given reduction strategy. This complexity measure satisfies Blum's axioms for step counting functions [6].

It is well known that the property of having n. f. (halt problem) is, in general, semirecursive, but we will use a result given in [5], which states that applications of n. f. s. of a given shape have n. f. More precisely, the set $\mathcal{N}$ is split into $\omega + 1$ disjoint classes $\mathcal{N}_h (h \geqq 0)$. If a n.f. $N \in \mathcal{N}_h$, then all λ-terms obtained by applying $N$ to $k$ arbitrary n. f. s., with $k \leqq h$, have n. f. too, but there exist $(h+1)$ n. f. s. $M_1 \ldots, M_{h+1}$ such that

$$NM_1 \ldots M_{h+1},$$

does not have n. f. If $N \in \mathcal{N}_\omega$, then all λ-terms obtained by applying $N$ to $k$ $(k \geqq 0)$ arbitrary n. f. s. have n. f. too. So the problem treated in this paper is to compute the maximum of the complexity of λ-terms like $NM_1 \ldots M_k$, when $N \in \mathcal{N}_h$, $M_i \in \mathcal{N}$ $(1 \leqq i \leqq k \leqq h)$. This maximum is found to be an elementary function of some integer parameters, which we introduce to describe the structures of $N, M_1, \ldots, M_k$. These parameters are not measures of size as defined in [6], since there exist infinite applications of n. f. s. with given values of such parameters.

The relations between λ-calculus, combinatory logic and *URS* are well known. The notion of "complexity" inside the other two formalisms has been studied in [4, 7, 8 and 2].

In [4], Batini and Pettorossi give the axioms for structural and computational complexity in the case of weak reductions inside combinatory logic. Moreover they introduce some possible measures and connect structural and computational measures in special cases. By using the same approach, Canal and Vignolle [7, 8] have obtained new and more far-reaching results.

Barendregt introduces in [2] a "norm" for *URS* which represents a measure of the computation length according to [14].

The present paper may be viewed as a continuation of and an improvement on [11]. In fact, a correspondence was established in [11] between n. f. s. and trees (which is used here too) and the complexity of applications like (1) was upper-bounded in certain cases. Here, general properties of complexity are established and the complexity of applications like (1) is maximized in certain cases different from those treated in [11]. In [12] we use the same approach as here to compute the maximum of the complexity of applications studied in [11].

In Section 2 we state some general properties of complexity, which are independent of the reduction algorithm; these properties justify further definitions and a tree representation of n. f. s.

In Section 3 we study applications like (1), when $N$ corresponds to a tree which is confined to a single root; in this case we get simple properties of complexity.

Section 4 shows the complexity evaluation of an application $NM$ where $N$ and $M$ are n. f. s. which satisfy suitable conditions.

We believe that the present paper may be a step towards a rigorous complexity theory for λ-calculus. Further studies will be carried out by the authors to compute the complexity of more extended classes of λ-terms.

## 2. KEY NOTIONS AND GENERAL PROPERTIES

In the present section we clarify some general properties of the complexity of applications of n. f. s. Besides their intrinsic interest, these properties are an help in understanding the intuitive meaning of the definitions we will introduce here. In [11] we took already into account some of these properties although we did not state them explicitly.

First we briefly recall the nomenclature referring to λ-terms. It is well known that any n. f. can be represented as follows:

$$N \equiv \lambda x_1 \ldots x_n . \zeta N_1 \ldots N_m$$

where $\equiv$ denotes identity (modulo α-reduction); $\zeta$ is a variable and $N_1 \ldots N_m$ are n.f.s.

In $N$, $\zeta$ is the *head*, $N_i$ $(1 \leqq i \leqq m)$ is the $i$-th *main argument* and $\lambda x_1 \ldots x_n$ is the *prefixed sequence* [9] (p. 88, 162). A n.f. without prefixed sequence is a λ-free n.f. $\#(N)$ will denote the number $n$, i.e. the number of variables in the prefixed sequence of $N$.

We will assume that different variables in a λ-term have different labels. This may always be achieved by α-reductions. In this way we may talk of occurrences of both free and bound variables without danger of confusion.

A λ-term is *λ-free* iff it reduces to a λ-free n. f. According to [9] (p. 162), if $T$, $U$ and $V$ are λ-terms, we will say that a given occurrence of $T$ in $U$ is a *functional* *occurence* just when $T$ occurs in a component $TV$ of $U$. A given occurrence of $T$ is an *argument occurrence* in $U$ just when $T$ occurs in a component $VT$ of $U$.

As usual, a *context* is a λ-term $C[\ ,\ \ldots,\ ]$ in which a given number (say $p$) of components is missing; then $C[M_1, \ldots, M_p]$ denotes the result of filling the missing components with $M_1, \ldots, M_p$.

Let $\mathscr{C}[T]$ denote the *complexity* of the λ-term $T$, i. e. the number of β-reductions necessary to reach the n. f. (if it exists) of $T$ according to a given reduction strategy (and infinite otherwise). In the present paper we will study the complexity of applications of n. f. s.

The properties which we consider in this section do not depend on the reduction algorithm, so we can choose it later.

The first lemma states the complexity independence of the abstractions which can never be reduced because there are too few arguments.

LEMMA 1: *Let* $N \equiv \lambda x_1 \ldots x_n . \zeta N_1 \ldots N_m$ *be a n.f. and* $N^* \equiv \lambda x_1 \ldots x_k . \zeta N_1 \ldots N_m$, *with* $k \leqq n$. *For any* $X_1, \ldots, X_k \in \Lambda$:

$$\mathscr{C}[N X_1 \ldots X_k] = \mathscr{C}[N^* X_1 \ldots X_k]. \tag{2}$$

*Proof:* $N X_1 \ldots X_k$ and $N^* X_1 \ldots X_k$ reduce in $k$ steps respectively to $\lambda x_{k+1} \ldots x_n . \zeta' N'_1 \ldots N'_m$ and $\zeta' N'_1 \ldots N'_m$ where $\zeta' \equiv \zeta[x_i/X_i]$ and $N'_j \equiv N_j[x_i/X_i]$ $(1 \leqq j \leqq m)$, $(1 \leqq i \leqq k)$. As the possible further reductions may take place only in the component $\zeta' N'_1 \ldots N'_m$, equality (2) holds. □

While lemma 1 allows us to eliminate some variables bound in the prefixed sequence of $N$, following lemma 2 allows us to neglect some arguments, when the head of $N$ is a free variable.

LEMMA 2: *If* $N \equiv \lambda x_1 \ldots x_n . a N_1 \ldots N_m$, *a is free in N and* $n \leqq k$, *then, for any* $X_1, \ldots, X_k \in \Lambda$:

$$\mathscr{C}[N X_1 \ldots X_k] = \mathscr{C}[N X_1 \ldots X_n].$$

*Proof:* As the head $a$ of $N$ is free then $N X_1 \ldots X_k$ is reduced, in $k$ steps, to $a N'_1 \ldots N'_m X_{n+1} \ldots X_k$, where $N'_j \equiv N_j[x_i/X_i]$ $(1 \leqq j \leqq m)$ $(1 \leqq i \leqq n)$. As further reductions cannot involve $X_{n+1}, \ldots, X_k$, we get the proof. □

Following lemma 3 proves that the introduction of a free variable as head of a n. f. $N$ does not modify the complexity of the application $N X_1 \ldots X_k$ when $\#(N) = k$.

LEMMA 3: *Let* $N \equiv \lambda x_1 \ldots x_k . \zeta N_1 \ldots N_m$ *be a n. f. and*

$$N^* \equiv \lambda x_1 \ldots x_k . a (\zeta N_1 \ldots N_m);$$

*then, for any* $X_1, \ldots, X_k \in \Lambda$:

$$\mathscr{C}[NX_1 \ldots X_k] = \mathscr{C}[N^* X_1 \ldots X_k]. \tag{3}$$

*Proof:* $NX_1 \ldots X_k$ and $N^* X_1 \ldots X_k$ are reduced, in $k$ steps, respectively to: $\zeta' N'_1 \ldots N'_m$ and $a(\zeta' N'_1 \ldots N'_m)$ where $\zeta' \equiv \zeta[x_i/X_i]$ and $N'_j \equiv N_j[x_i/X_i]$ $(1 \leqq j \leqq m)$ $(1 \leqq i \leqq k)$. All the further reductions take place in the component $\zeta' N'_1 \ldots N'_m$; so equality (3) holds true. □

In lemma 1 the bound variables $x_{k+1}, \ldots, x_n$, never reduced, behave as free variables. This fact may be generalized, obtaining a refinement of the distinction between free and bound variables. This refinement was already given in [5]. We recall here the following:

DEFINITION 1: In a n. f. $N \equiv \lambda x_1 \ldots x_k . \zeta N_1 \ldots N_m$ we recursively define the *h-replaceable* $(1 \leqq h \leqq k)$ variables as follows:

– the variables $x_i$ $(1 \leqq i \leqq h)$ are $h$-replaceable;

– if $Z \equiv \lambda z_1 \ldots z_r . \theta Z_1 \ldots Z_s$ is a component of $N$ and $\theta$ is $h$-replaceable, then all the variables which are bound in the prefixed sequence of $Z_j$ $(1 \leqq j \leqq s)$ are $h$-replaceable.

A variable which does not satisfy the previous definition is said *non-h-replaceable*.

Moreover we say that a variable is *ω-replaceable* (*non-ω-replaceable*) if it is $h$-replaceable for some $h \geqq 1$ (non-$h$-replaceable for every $h$).

*Example* 1: In the following n. f. the $h$-replaceable variables which are non-$(h-1)$-replaceable are underlined by $h$ lines:

$$\lambda xy . \underline{x}(\lambda tu . a(\underline{t}(\lambda v . b(\underline{u}(c\underline{x})))))(\underline{\underline{y}}(\lambda z . d(\underline{\underline{z}}x)))$$

In [5] it is proved that iff a variable is $h$-replaceable then it can be replaced by an arbitrary n. f. whenever the n. f., in which it occurs, is applied successively to (at least) $h$ suitable n. f. s.

As an application of definition 1, we prove the following lemma, which assures us that the complexity of a λ-term of the shape $NX_1 \ldots X_k$ is independent of the number of non-$k$-replaceable variables which occur in "particular" positions in $N$.

LEMMA 4: *Let $N$ be a n. f. and $Z$ be a component of $N$ such that either $Z$ is λ-free and it has a non-k-replaceable head or the variables bound in the prefixed sequence of $Z$ are non-k-replaceable. If for some context $C[\ ]$, $N \equiv C[Z]$, $N^* \equiv C[aZ]$ and $a$ is non-k-replaceable when it fills the hole of $C[\ ]$ then, for any $X_1, \ldots, X_k \in \Lambda$:*

$$\mathscr{C}[NX_1 \ldots X_k] = \mathscr{C}[N^* X_1 \ldots X_k].$$

*Proof:* By hypothesis $Z$ is either of the shape: $Z \equiv b Z_1 \ldots Z_q$ or of the shape: $Z \equiv \lambda y_1 \ldots y_p . \overline{Z}$ where $y_i$ $(1 \leqq i \leqq p)$ and $b$ are non-$k$-replaceable variables. When $N \equiv \lambda x_1 \ldots x_n . \overline{N}$ is applied to $X_1, \ldots, X_k$, the reductions involving $Z$ involve respectively only $Z_1, \ldots, Z_q$ in the first case and $\overline{Z}$ in the second case. If $Z$ is changed into $a Z$ the previous reductions remain unchanged; moreover, as $a$ is non-$k$-replaceable, then we are sure that the introduction of $a$ does not create new redexes. So the lemma is proved. □

In [5] the notion of $h$-replaceable variables allowed a partition of the n. f. s. into classes $\mathcal{N}_h (h \geqq 0)$, such that applications of a n. f. belonging to $\mathcal{N}_h$ to $k \leqq h$ n. f. s., possess n. f. too. To make self-contained the present paper, we recall here this partition. First we need the notion of nested occurrences of variables. Two occurrences of variables are said nested in a n. f. $N$ iff the first one is the head of a component $Z$ of $N$, being the other the head of some main argument of $Z$. More formally we have:

DEFINITION 2: If $Z \equiv \lambda z_1 \ldots z_r . \theta Z_1 \ldots Z_s$ is a component of a n. f. $N$, then this occurrence of $\theta$ and the occurrence of the head of $Z_j$ $(1 \leqq j \leqq s)$ are *nested* in $N$.

Let us note that this relation is irreflexive, symmetric and non-transitive.

*Example* 2: In the n. f. of example 1:

- the leftmost occurrence of x and the occurrence of a are nested;
- the leftmost occurrence of x and the occurrence of y are nested;
- the occurrence of a and t are nested, etc.

Moreover, let us define *group of nested occurrences* in a n. f. $N$ a sequence $(y_1, \ldots, y_t)$ of variable occurrences, such that:

- $y_1$ is the head of $N$;
- $y_i$ and $y_{i+1}$ are nested $(1 \leqq i \leqq t-1)$;
- $y_t$ is an argument occurrence.

*Example* 3: In the n. f. of example 1 we have two groups of nested occurrences: (x, a, t, b, u, c, x) and (x, y, d, z, x).

We can now define the classes $\mathcal{N}_h (h \geqq 0)$:

DEFINITION 3: A n. f. $N$ with $\# (N) = n$ belongs to $\mathcal{N}_h$ $(0 \leqq h \leqq n-1)$ iff in $N$:

- there exist no two nested occurrences of $h$-replaceable variables;
- there exist at least two nested occurrences of $(h+1)$-replaceable variables.

DEFINITION 4: A n. f. $N$ with $\# (N) = n$ belongs to $\mathcal{N}_n \cup \mathcal{N}_\omega$ iff in $N$ there exist no two nested occurrences of $n$-replaceable variables. More specifically $N \in \mathcal{N}_n$ or $N \in \mathcal{N}_\omega$ according to the head of $N$ is bound or free.

*Example* 4: The n. f. of example 1 belongs to $\mathcal{N}_1$ since:

– there are no two nested occurrences of 1-replaceable variables;

– in the subterm zx the occurrences of z and x are nested and moreover these variables are both 2-replaceable.

REMARK 1: From definition 4 the components which occur in $N \in \mathcal{N}_\omega$ as main arguments of components whose heads are ω-replaceable belong in their turn to $\mathcal{N}_\omega$. Moreover, if $Z$ is one of such components of $N$, and $\lambda x_1 \ldots x_n$ is the prefixed sequence of $N$, then also $\lambda x_1 \ldots x_n . Z \in \mathcal{N}_\omega$.

In [5] it is proved that a n. f. $N$ belongs to the class $\mathcal{N}_h$ iff all the λ-terms obtained by applying $N$ to $h$ arbitrary n. f. s. possess n. f. too, but there exist $h+1$ n. f. s. $M_1, \ldots, M_{h+1}$ such that $NM_1 \ldots M_{h+1}$ possesses no n. f. More precisely the following theorems are stated:

THEOREM 1: *A n. f.* $N \in \mathcal{N}_h$ $(h \geqq 0)$ *iff:*

– $\forall M_1, \ldots, M_h \in \mathcal{N}: NM_1 \ldots M_h$ *possesses n. f.*;

– $\exists M_1, \ldots, M_{h+1} \in \mathcal{N}: NM_1 \ldots M_{h+1}$ *possesses no n. f.*

THEOREM 2: *A n. f.* $N \in \mathcal{N}_\omega$ *iff:*

– $\forall h (h \geqq 0), \forall M_1, \ldots, M_h \in \mathcal{N}: N M_1 \ldots M_h$ *possesses n. f.*

REMARK 2: From theorem 2 it follows that, if $N \in \mathcal{N}_\omega$ and $M \in \mathcal{N}$, then the n. f. of $NM$ will belong to $\mathcal{N}_\omega$ too.

The previous classification assures us that some applications of n. f. s. reduce to n. f. in a finite number of steps. More precisely from theorems 1 and 2, we can assure that the application:

$$NM_1 \ldots M_k, \tag{4}$$

has n.f. when $N \in \mathcal{N}_h$, $h \geqq k$ (ω is considered greater than any integer) and $M_1, \ldots, M_k$ are n.f.s. We wont evaluate finite complexities, then only applications satisfying the latest conditions will be considered in the following. Moreover, utilizing the results of the present section, we can limit ourselves to consider application $NM_1 \ldots M_k$ with $N \in \mathcal{N}_\omega$ and $\#(N)=k$. Namely we can prove the following lemma:

LEMMA 5: *If* $N \in \mathcal{N}_h$ *and* $k \leqq h$, *for any* $X_1, \ldots, X_k \in \Lambda$ *there exists a n.f.* $N^*$ *and an integer* $r \leqq k$ *such that* $N^* \in \mathcal{N}_\omega$, $\#(N^*)=r$ *and moreover:*

$$\mathscr{C}[NX_1 \ldots X_k] = \mathscr{C}[N^* X_1 \ldots X_r].$$

*Proof:* Let $\#(N)=n$. If $N \in \mathcal{N}_\omega$ we consider separately two cases:

– if $n \geqq k$ let $r=k$ and let us choose $N^*$ as defined in lemma 1. We point out that $N^* \in \mathcal{N}_\omega$;

— if $n<k$ let $r=n$ and $N^* \equiv N$. Lemma 2 assures us that the complexity does not change.

If $N \in \mathcal{N}_h$ $(h \neq \omega)$ by definitions 3 and 4 we have $n \geqq h$; let then $N \equiv \lambda x_1 \ldots x_k \ldots x_n . \zeta N_1 \ldots N_m$. In this case we choose $r=k$. By lemma 1, we can replace $N$ by the n.f.:

$$N' \equiv \lambda x_1 \ldots x_k . \zeta N_1 \ldots N_m.$$

Since $N \in \mathcal{N}_h$ and $h \geqq k$, $N' \in \mathcal{N}_k \cup \mathcal{N}_\omega$; then if $N' \in \mathcal{N}_\omega$ we can choose $N^* \equiv N'$. Otherwise, by lemma 3, we can replace $N'$ by: $N^* \equiv \lambda x_1 \ldots x_k . a(\zeta N_1 \ldots N_m)$. Since $N' \in \mathcal{N}_k$, then $N^* \in \mathcal{N}_\omega$ and it is the desired n.f. □

We point out that the existence of $N^*$, in lemma 5, is proved in a constructive way.

To summarize the results of this section, we can say that, in what follows, only applications $NM_1 \ldots M_k$, where $N \in \mathcal{N}_\omega$, $\#(N)=k$ and $M_1, \ldots, M_k$ are n.f.s., will be considered. We call *complete* such applications. Let us note that a $\lambda$-free n.f. $N$ is a complete application according to this definition, since $N \in \mathcal{N}_\omega$ and $\#(N)=k=0$. The word complete is suggested by the fact that the complexity does not rise up by increasing either the number of variables bound in the prefixed sequence of $N$ or the number of arguments to which $N$ is applied.

In a n.f. $N \in \mathcal{N}_h$ a variable is said to be *replaceable* (*non-replaceable*) if it is $h$-replaceable (non-$h$-replaceable). Obviously a variable which is replaceable is also $\omega$-replaceable (but the converse is not always true). Only in the particular case of $N \in \mathcal{N}_\omega$ the replaceable variables of $N$ coincide with the $\omega$-replaceable ones. From definitions 3 and 4 it follows that, in any n.f., there are no two occurrences of replaceable variables which are nested.

*Example* 5: In the n.f. of example 1, which belongs to $\mathcal{N}_1$, the replaceable variables are the 1-replaceable variables. We rewrite this n.f. and encircle its replaceable variables:

$$\lambda xy . \textcircled{x} (\lambda tu . a (\textcircled{t} (\lambda v . b (\textcircled{u} (c \textcircled{x}))))) (y (\lambda z . d (z \textcircled{x}))).$$

Using the definition of replaceable variables, we can rewrite lemma 4 by substituting non-replaceable for non-$k$-replaceable, provided that $N \in \mathcal{N}_h$ and $k \leqq h$. Then lemma 4 states that the complexity is independent of the number of non-replaceable, nested variables. Now we introduce a tree representation of n.f.s. which is independent of this number but makes evident the occurrences of replaceable variables as heads of components whose main arguments are not all $\lambda$-free. This representation will be used to compute the complexity of applications of n.f.s. The same representation was used, in an analogous way, in [11].

Given a n.f. $N$, we build up the corresponding tree according to the following (alternative) rules:

RULE 1: If in $N$ every replaceable variable is head of a component whose main arguments are all λ-free n.f.s., then the corresponding tree is a single root labelled by $N$.

RULE 2: Otherwise we scan $N$ from left to right and every time we find a component whose head is replaceable we store its main arguments and then replace each of these in $N$ by a $*$. After this procedure let $M_1, \ldots, M_s$ $(s \geqq 1)$ be the stored n.f.s. and $N_*$ the obtained string. Then the tree which represents $N$ is a root with label $N_*$ connected by $s$ (ordered) branches to the trees which represent respectively $M_1, \ldots, M_s$.

REMARK 3: We notice that (according to rule 1) a n.f. without occurrences of replaceable variables is represented by a single root.

The above construction may be visualized as in figure 1.

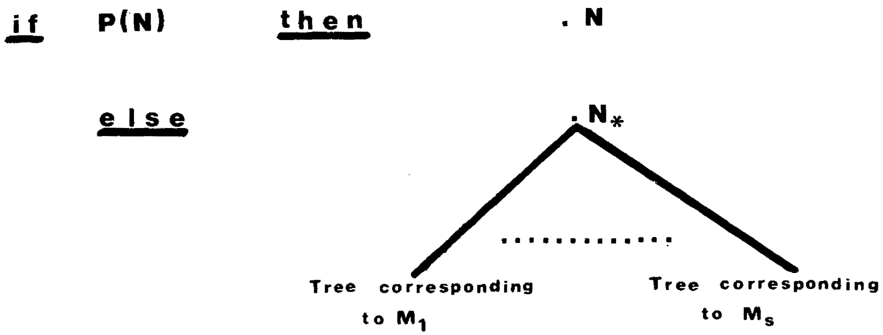


**Figure 1. – Tree corresponding to a n.f. $N$, where: $P$ is the predicate tested in rule 1; $N_*$ and $M_j (1 \leqq j \leqq s)$ have the same meaning as in rule 2.**

*Example* 6: The tree which represents the n.f. of example 1 is shown in figure 2.

About the labels of nodes in trees which represent n.f.s. we notice that:

– the labels of terminal nodes are n.f.s.;

– the labels of non-terminal nodes are n.f.s. with some main arguments replaced by $*$.

We may extend in an obvious way the given definitions of head, main arguments and prefixed sequence including $*$ into the set of n.f.s.

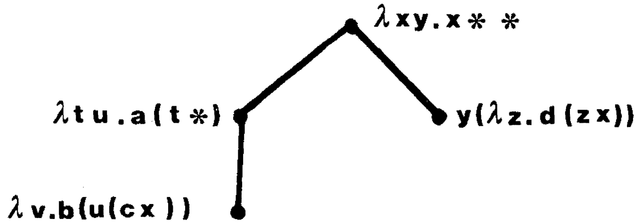


**Figure 2. — Tree corresponding to the n.f. of example 1.**

Since in any n.f. there are no two nested occurrences of replaceable variables, then all labels of nodes different from the root have heads which are non-replaceable.

We say that the *level* of a node in a tree is the maximum number of branches which may belong to a path from this node to a terminal node. More formally, we have:

— the level of terminal nodes is 0;

— the level of a non-terminal node is the maximum between the levels of its sons plus 1.

Now, let us classify the n.f.s. according to the levels of the roots in the corresponding trees.

DEFINITION 5: $N \in \mathscr{F}_l$ iff the root of the tree corresponding to $N$ has level $l$.

*Example* 7: The n.f. of example 1 belongs to $\mathscr{F}_2$ since the root of the tree in figure 2 has level 2.

As the 0-level n.f.s. have quite simple properties in regard to the β-reduction, in next section we will examine applications of such n.f.s.

In the last definition of the present section we associate to some variables the level at which they are bound.

DEFINITION 6: Let τ be the tree corresponding to a n.f. $N$. The *t-bound* variables in $N$ are the variables which are bound in the prefixed sequence of labels of nodes at level $t$ in τ.

*Example* 8: In the n.f. whose tree is shown in figure 2:

— x, y are 2-bound;

— t, u are 1-bound;

— v is 0-bound.

We notice that in a n.f. $N \in \mathscr{F}_l$:

— the $l$-bound variables are ω-replaceable (since they are bound in the prefixed sequence of $N$);

— the $t$-bound variables ($0 \leqq t \leqq l-1$) are replaceable (since they are bound in the prefixed sequence of components of $N$ which are main arguments of other components whose heads are replaceable).

### 3. PROPERTIES OF N.F.S. OF LEVEL 0

In section 2 we associated to each n.f. the level of the corresponding tree; the n.f.s. corresponding to trees consisting of a single root (n.f.s. of level 0) look quite simple. This simplicity in the representation corresponds to an analogous simplicity in evaluating the complexity of applications. In fact let us consider a complete application:

$$N M_1 \ldots M_k \tag{5}$$

where $N \in \mathscr{F}_0$. In order to obtain the n.f. of (5) it is sufficient to reduce:

— the prefixed sequence of $N$, obtaining thus a λ-term $N'$;

— the prefixed sequence of the functional occurrences of $M_i$ ($1 \leqq i \leqq k$) in $N'$.

The conditions on $N$ assure us that all the arguments of $M_i$ in $N'$ are λ-free λ-terms, and hence the reduction ot the prefixed sequences of $M_i$ does not generate new redexes.

To obtain further results in studying such applications we need some parameters, in order to quantify the number of occurrences of effectively replaced variables, i.e. variables bound in the prefixed sequences of $N$, $M_1$, ..., $M_k$. We will distinguish variables bound in different abstractions of $N$ and in different $M_i$ ($1 \leqq i \leqq k$).

DEFINITION 7: If $N M_1 \ldots M_k$ is an application and $N \equiv \lambda x_1 \ldots x_k . \overline{N}$, we define:

— $\sigma_j$ maximum number of occurrences of $x_j$ ($1 \leqq j \leqq k$) within a group in $N$;

— $n$ number of groups of nested occurrences of replaceable variables in $N$;

— $\nu$ maximum number of main arguments of the components of $N$ whose head is replaceable;

— $\rho_j$ number of occurrences in $M_j$ of the variables bound in its prefixed sequence ($1 \leqq j \leqq k$);

— $\delta$ maximum of $\#(M_j)$ ($1 \leqq j \leqq k$).

*Example* 9: Let

$$N \equiv \lambda xyz . a(\lambda t . x(ty)(cx))(z(d(ze))),$$

$$M_1 \equiv \lambda uv . uvv,$$

$$M_2 \equiv \lambda u . u(\lambda w . ww) f,$$

$$M_3 \equiv \lambda uv . v(\lambda w . w) u.$$

We have in this case:

$$\sigma_1=2,\quad n=3,\quad \rho_1=3,\quad \delta=2,$$
$$\sigma_2=1,\quad \nu=2,\quad \rho_2=1,$$
$$\sigma_3=2,\quad \rho_3=2.$$

By means of the parameters introduced in definition 7 we can evaluate how the number of occurrences of a variable in a n.f. $N \in \mathscr{F}_0, \mathscr{N}_\omega$ will increase in the n.f. of the application $NM$. We notice that the proof of the following lemma does not require that $NM$ is a complete application. This maximization is obviously independent of the used reduction strategy, and moreover we are sure to reach the n.f., whatever the reduction strategy is, thanks to the given conditions on $N$. This is proved in [5] for the general case of $N \in \mathscr{N}_h (h \geqq k)$ and $\mathscr{F}_l (l \geqq 0)$. Then in the proof of the following lemma we will use a "suitable" reduction strategy, which permits us to induce on the parameter $\sigma_1$. We define

$$[r]^s = \begin{cases} r^s & \text{if } r>0, \\ 1 & \text{if } r=0. \end{cases}$$

LEMMA 6: *Let $NM$ be an application, where $N \in \mathscr{F}_0, \mathscr{N}_\omega$ and $M \in \mathscr{N}$. If a variable a occurs $\gamma$ times in $N$, and a does not occur in $M$, then in the n.f. of $NM$ a occurs at most $\gamma . [\rho_1]^{\sigma_1}$ times, where $\sigma_1$ and $\rho_1$ are computed for $NM$ according to definition* 7.

*Proof:* We make this proof by induction on $\sigma_1$. Let $N \equiv \lambda x_1 . \overline{N}$.

*First step:* $\sigma_1 = 0$ means that the desired n.f. is $\overline{N}$, and the lemma is obviously true.

*Inductive step:* We assume this lemma true for $\sigma_1 \leqq v$ and we will prove it for $\sigma_1 = v+1$. $\overline{N}$ will contain a given number (say $n$) of groups of nested occurrences. Let $N^*$ be obtained from $\overline{N}$ by replacing, in each group, the outermost occurrence of $x_1$ by a variable (say $b$) which does not occur in $\overline{N}$. $N^*$ contains a given number (say $q$) of free occurrences of $b$; clearly it will be $q \leqq n$, since one occurrence of $b$ may belong to more than one group. Let $P_1, \ldots, P_q$ be the components of $N^*$ whose head is $b$ [we notice that some $P_j$ $(1 \leqq j \leqq q)$ may coincide with one argument occurrence of $b$]. Since these components are disjoint by construction, there exists a context $C[\ ,\ldots,\ ]$ such that $N^* \equiv C[P_1, \ldots, P_q]$. We distinguish the occurrences of $a$ according to they occur in one $P_j (1 \leqq j \leqq q)$ or not. More precisely we define:

– $\gamma_j$ as the number of occurrences of $a$ in $P_j$;

$$-\gamma^* = \gamma - \sum_{j=1}^{q} \gamma_j.$$

For $(\lambda x_1 . P_j) M$ we have $\sigma_1 \leqq v$ and then, by inductive hypothesis, its n.f. $P'_j$ contains at most $\gamma_j [\rho_j]^v$ occurrences of $a$. By construction $C[P'_1, \ldots, P'_q]$ is the n.f. $N'$ of $(\lambda x_1 . N^*) M$. Again by construction $(\lambda x_1 . \overline{N}) M = (\lambda b . N') M$. Let $P'_j \equiv b P_1^{(j)} \ldots P_{m_j}^{(j)}$ for $1 \leqq j \leqq q$. Then in $N'[b/M]$ we must reduce at most $q$ redexes of the shape: $M P_1^{(j)} \ldots P_{m_j}^{(j)}$ for $1 \leqq j \leqq q$. From above, in $P_1^{(j)}$, $P_2^{(j)}, \ldots, P_{m_j}^{(j)} a$ occurs $\gamma_j$ times if $\rho_1 = 0$ and $\gamma_j [\rho_1]^v$ times if $\rho_1 > 0$. In the first case, by reducing the variables of the prefixed sequence of $M$, the number of occurrences of $a$ cannot increase, and in the n.f. of $(\lambda x_1 . \overline{N}) M$ $a$ occurs at most $\gamma$ times. Otherwise, at most $\rho_1$ copies of the same component are created and so $a$ occurs at most $\gamma_j . [\rho_1]^{v+1}$ times in the n.f. of $M P_1^{(j)} \ldots P_{m_j}^{(j)}$. Then in the n.f. of $(\lambda x_1 . \overline{N}) M$ $a$ will occur at most $\sum_{j=1}^{q} \gamma_j [\rho_1]^{v+1} + \gamma^* \leqq \gamma . [\rho_1]^{v+1}$ times. □

Necessary conditions to reach the maximum evaluated in lemma 6 are:

– all occurrences of $x_1$ are functional;

– the occurrences of $a$ are all inner to the innermost functional occurrence of $x_1$;

– $M$ has exactly $\rho_1$ occurrences of a variable bound in its prefixed sequence.

We avoid to analyze all the possible structures of $N$ and $M$, and we confine ourselves to prove that a particular choice of $N$ and $M$ reaches this maximum.

LEMMA 7: *The maximum of lemma* 6 *is reached by choosing:*

$$N \equiv \underbrace{\lambda x_1 . c(\underbrace{x_1(c(x_1 \ldots (c(x_1}_{\sigma_1}\underbrace{(a \ldots a)}_{\gamma_1})) \ldots ))) \ldots (\underbrace{x_1(c(x_1 \ldots (c(x_1}_{\sigma_n}\underbrace{(a \ldots a)}_{\gamma_n})) \ldots )))}_{n}$$

$$M \equiv \begin{cases} \lambda z . \underbrace{z \ldots z}_{\rho_1} & \text{if } \rho_1 > 0, \\ d & \text{if } \rho_1 = 0, \end{cases}$$

where $\sum_1^n {}_i \gamma_i = \gamma$.

The proof of this lemma is performed by executing the reductions of $NM$ and hence it is omitted. □

*Example* 10: If the values of the parameters are the following:

$$\sigma_1 = 2, \quad n = 3, \quad \rho_1 = 2, \quad \gamma = 3,$$

we build

$$N \equiv \lambda x_1 . c(x_1(c(x_1 a)))(x_1(c(x_1 a)))(x_1(c(x_1 a))),$$

$$M \equiv \lambda z . zz.$$

The n.f. of NM is

$$N' \equiv cRRR \qquad \text{where} \quad R \equiv c(aa)(c(aa))$$

and a occurs really $\gamma . \rho_1^{\sigma_1} = 3 . 2^2 = 12$ times in N′

Now we must choose a reduction strategy, because we will consider also the behaviour of intermediate redexes. As in [5] it has been proved that any complete application reaches its normal form whatever strategy is applied, we choose the leftmost-innermost one. This strategy is efficient in the λ-I-calculus, i.e. for λ-terms in which all bound variables occur at least once (it is well known [3] that no recursive strategy minimizes the complexity of all λ-terms).

As usual $\geqq$ will denote reducibility in a finite (possibly zero) number of steps and $\geqq_i$ will denote reducibility in exactly $i$ steps.

If $N \equiv \lambda x_1 \ldots x_k . L_0$, the reduction of (5), according to the leftmost-innermost strategy, yields consecutively:

$$\begin{aligned}(\lambda x_1 \ldots x_k . L_0) M_1 \ldots M_k &\geqq_1 (\lambda x_2 \ldots x_k . L_0 [x_1/M_1]) M_2 \ldots M_k \\ &\geqq (\lambda x_2 \ldots x_k . L_1) M_2 \ldots M_k \geqq_1 (\lambda x_3 \ldots x_k . L_1 [x_2/M_2] M_3 \ldots M_k \geqq \ldots \\ &\geqq (\lambda x_k . L_{k-1}) M_k \geqq_1 L_{k-1} [x_k/M_k] \geqq L_k\end{aligned}$$

where $L_j (1 \leqq j \leqq k)$ denotes the n.f. of $L_0 [x_1/M_1, \ldots, x_j/M_j]$. At present we disregard the number of reductions needed to obtain $L_j$ from $L_{j-1} [x_j/M_j]$ $(1 \leqq j \leqq k)$, even if it is easy to evaluate, as we will see later.

We notice that all redexes of $NM_1 \ldots M_k$ which look as: $(\lambda x_{j+1} \ldots x_k . L_j) M_{j+1} \ldots M_k$ are applications of n.f.s. again. We call these redexes *principal redexes* of $NM_1 \ldots M_k$ and we use them to examine the properties of applications like (5). The n.f. $L_k$ of $NM_1 \ldots M_k$ is also a principal redex.

First we prove that all functional occurrences of n.f.s. in principal redexes belong to $\mathscr{F}_0$, and such principal redexes are still complete applications.

LEMMA 8: *All principal redexes of a complete application*

$$NM_1 \ldots M_k,$$

*where* $N \in \mathscr{F}_0$ *are complete applications too, where the n.f.s. occurring in functional position belong to* $\mathscr{F}_0$.

The proof of this lemma is given in the appendix.

Now we prove that the number of nested occurrences of replaceable variables of $N$ does not increase during the reduction, even if the total number of variable occurrences do (since $N \in \mathscr{F}_0$, $\mathscr{N}_\omega$, the replaceable variables of $N$ are the variables bound in its prefixed sequence).

LEMMA 9: *Let* $NM_1 \ldots M_k$ *be a complete application, where* $N \in \mathscr{F}_0$, *and let* $(\lambda x_{j+1} \ldots x_k . L_j) M_{j+1} \ldots M_k$ *be one of its principal redexes* $(0 \leqq j \leqq k-1)$. *In* $L_j$ *there are at most* $\sigma_l$ *occurrences of the variables* $x_l$ *inside a group, for* $(j+1 \leqq l \leqq k)$, *where* $\sigma_1, \ldots, \sigma_k$ *are computed for* $NM_1 \ldots M_k$ *according to definition* 7.

The proof of this lemma is given in the appendix, too.

To evaluate the complexity it is more convenient first maximize, by means of lemmas 6, 8 and 9, the number of occurrences of the variables bound in the prefixed sequence of $N$ inside the principal redexes of $NM_1 \ldots M_k$.

LEMMA 10: *Let* $NM_1 \ldots M_k$ *be a complete application where* $N \in \mathscr{F}_0$ *and let* $(\lambda x_{j+1} \ldots x_k . L_j) M_{j+1} \ldots M_k$ *be one of its principal redexes* $(0 \leqq j \leqq k-1)$. *In* $L_j$ *the variables* $x_l (j+1 \leqq l \leqq k)$ *will occur at most* $n . \sigma_1 . \prod_{i=1}^{j} [\rho_i]^{\sigma_i}$ *times, where all parameters are computed for* $NM_1 \ldots M_k$ *according to definition* 7.

*Proof:* We will prove this lemma by induction on the index $j$.

*First step:* For $j=0$ we have $\prod_{i=1}^{0} [\rho_i]^{\sigma_i} = 1$, and the lemma obviously holds, because the principal redex is just $NM_1 \ldots M_k$, where $x_l$ occurs at most $n . \sigma_l$ times.

*Inductive step:* Suppose that the lemma holds for $j=r$, and let us prove that it holds for $j=r+1$. In $(\lambda x_{r+1} \ldots x_k . L_r) M_{r+1} \ldots M_k$ there are at most $n . \sigma_1 . \prod_{i=1}^{r} [\rho_i]^{\sigma_i}$ occurrences of $x_l$ (by inductive hypothesis) and at most $\sigma_l$ occurrences of $x_l$ inside a group (by lemma 9) for $r+1 \leqq l \leqq k$. Now we must reduce $(\lambda x_{r+1} . L_r) M_{r+1}$ where $\lambda x_{r+1} . L_r \in \mathscr{F}_0, \mathscr{N}_\omega$ (by lemma 8). To compute the number of occurrences of $x_l$ in $L_r [x_{r+1} / M_{r+1}]$ we can then apply lemma 6, with $\gamma$ equal to the number of occurrences of $x_l$ in $L_r$, and $\rho_1^*, \sigma_1^*$ computed for $(\lambda x_{r+1} . L_r) M_{r+1}$ (the $*$ has been adjoint to avoid confusion with the values of the same parameters computed for $NM_1 \ldots M_k$). Then we have:

$$\gamma = n . \sigma_l . \prod_{i=1}^{r} [\rho_i]^{\sigma_i}, \qquad \rho_1^* = \rho_{r+1}, \qquad \sigma_1^* = \sigma_{r+1}.$$

So, $x_l$ occurs at most $\gamma . [\rho_1^*]^{\sigma_1^*}$ times, i.e. $n . \sigma_l . \prod_{i=1}^{r+1} [\rho_i]^{\sigma_i}$ times in the n.f. of $(\lambda x_{r+1} . L_r) M_{r+1}$. □

The maximum computed in lemma 10 is actually reached if:

– in $N$ each group of nested occurrences of variables contains exactly $\sigma_i$ functional occurrences of $x_i (1 \leqq i \leqq k)$ and moreover they occur from left to right in non-decreasing order of the index $i$;

– $M_i$ has $\rho_i$ occurrences of the same variable.

We prove that there exists a choice of $N, M_1, \ldots, M_k$ which reaches these maxima.

LEMMA 11: *The maxima of lemma* 10 *are obtained for the choice:*

$$N \equiv \lambda x_1 \ldots x_k . c \underbrace{\overline{N}_1 \ldots \overline{N}_1}_{n}$$

*where:*

$$\overline{N}_i \equiv \begin{cases} \underbrace{(x_i(c(x_i(c \ldots (x_i}_{\sigma_i}(c\overline{N}_{i+1})\underbrace{c \ldots c}_{\nu-1}) \ldots)\underbrace{c \ldots c}_{\nu-1}))\underbrace{c \ldots c}_{\nu-1}) & \text{if } \rho_i > 0 \\ \underbrace{(c(x_i\underbrace{c \ldots c}_{\nu}) \ldots (x_i\underbrace{c \ldots c}_{\nu})}_{\sigma_i}\overline{N}_{i+1}) & \text{if } \rho_i = 0 \end{cases}$$

$$(1 \leqq i \leqq k-1)$$

$$\overline{N}_k \equiv \underbrace{(x_k(c(x_k(c \ldots (x_k}_{\sigma_k} \underbrace{c \ldots c}_{\nu}) \ldots)\underbrace{c \ldots c}_{\nu-1})) \underbrace{c \ldots c}_{\nu-1})$$

*and:*

$$M_1 \equiv \begin{cases} \lambda z t_1 \ldots t_{\delta-1} . \underbrace{z \ldots z}_{\rho_i} & \text{if } \rho_i > 0 \\ \lambda t_1 \ldots t_\delta . d & \text{if } \rho_i = 0 \end{cases} \qquad (1 \leqq i \leqq k).$$

We omit the proof of this lemma, because it consists in executing the reductions. □

We can now evaluate the complexity (according to the leftmost-innermost reduction strategy) of the complete application $NM_1 \ldots M_k$, when $N \in \mathcal{F}_0$. For a function $f(x_1, \ldots, x_n)$ $(n>0)$, we will write $f(x_1, \ldots, x_n) \leqq s$ to indicate that $\forall x_1, \ldots, x_n, f(x_1, \ldots, x_n) \leqq s$ and in the current domain of $f(x_1, \ldots, x_n)$ there exist $x'_1, \ldots, x'_n$ such that $f(x'_1, \ldots, x'_n) = s$.

THEOREM 3: *Let $NM_1 \ldots M_k$ be a complete application and $N \in \mathcal{F}_0$. According to the leftmost-innermost reduction strategy:*

$$\mathcal{C}[NM_1 \ldots M_k] \leqq k + n . \min[\nu, \delta] . \sum_{r=0}^{k-1} \sigma_{r+1} . \prod_{i=1}^{r} [\rho_i]^{\sigma_i}.$$

*Proof:* First we prove that the given expression is an upperbound for the complexity of $NM_1 \ldots M_k$. To prove this it is convenient to prove somewhat more, i.e., that we need at most $1 + n . \sigma_{r+1} \prod_{i=1}^{r} [\rho_i]^{\sigma_i} . \min[\nu, \delta]$ steps to obtain from one principal redex the following one ($r$ is the order number of the principal redex which is reduced). In fact, to obtain from a redex:

$$(\lambda x_{r+1} \ldots x_k . L_r) M_{r+1} \ldots M_k$$

the following one, we must execute one reduction to get

$$(\lambda x_{r+2} \ldots x_k . L_r [x_{r+1}/M_{r+1}]) M_{r+2} \ldots M_k$$

and then we must reduce $L_r[x_{r+1}/M_{r+1}]$ to n.f. By lemma 10 there will be at most $n.\sigma_{r+1} \prod_{i=1}^{r} [\rho_i]^{\sigma_i}$ occurrences of $x_{r+1}$ in $L_r$. So we will have at most the same number of components to reduce in $L_r[x_{r+1}/M_{r+1}]$. Each one of these components needs at most min $[\nu, \delta]$ reductions. So the first point of the theorem is proved; in fact

$$\sum_{r=0}^{k-1} (1+n.\sigma_{r+1} . \prod_{i=1}^{r} [\rho_i]^{\sigma_i} . \min[\nu, \delta]) = k + n . \min[\nu, \delta] . \sum_{r=0}^{k-1} \sigma_{r+1} . \prod_{i=1}^{r} [\rho_i]^{\sigma_i}.$$

To prove this expression is really a maximum for the complexity it will be sufficient to choose $N, M_1, \ldots, M_k$ as in the proof of lemma 11. □

*Example* 11: If the values of the parameters are the following:

$k=2, \quad \sigma_1=1, \quad \sigma_2=2, \quad n=2, \quad \rho_1=2, \quad \rho_2=0, \quad \nu=2, \quad \delta=1$

we build:

$$\overline{N}_2 \equiv (x_2 (c (x_2 cc)) c),$$

$$\overline{N}_1 \equiv (x_1 (c \overline{N}_2) c),$$

$$N \equiv \lambda x_1 x_2 . c \overline{N}_1 \overline{N}_1,$$

$$M_1 \equiv \lambda z . zz,$$

$$M_2 \equiv \lambda t . d,$$

$$NM_1 M_2 \geqq_1 (\lambda x_2 . c \overline{N}_1 [x_1/M_1] \overline{N}_1 [x_1/M_1]) M_2$$

where:

$$\overline{N}_1 [x_1/M_1] \equiv (M_1 (c \overline{N}_2) c) \geqq_1 (c \overline{N}_2 (c \overline{N}_2) c) \equiv N'_1.$$

In the principal redex $(\lambda x_2 . c N'_1 N'_1) M_2$, $x_2$ occurs $n . \sigma_2 . \rho_1^{\sigma_1} = 8$ times, as stated in lemma 10.

$$(\lambda x_2 . c N'_1 N'_1) M_2 \geqq_1 c N'_1 [x_2/M_2] N'_1 [x_2/M_2]$$

where:

$$N'_1 [x_2/M_2] \equiv (c(M_2(c(M_2 cc))c)(c(M_2(c(M_2 cc))c))c \geqq_4 (c(dc)(c(dc))c).$$

Then the total number of β-reductions is exactly

$$\left\{k+n.\min[v,\delta].\sum_{r=0}^{k-1}\sigma_{r+1}.\prod_{i=1}^{r}[\rho_i]^{\sigma_i}\right\}=2+2.[1+2.2]=12.$$

## 4. PROPERTIES OF LEVELLED N.F.S.

Now we will consider n. f. s. represented by trees of level greater than 0. Since we are unable to give general properties of applications of these n. f. s. we will study a subset of $\mathscr{N}$. The interest of this subset is that it is, in some sense, complementary to the subset considered in [11]. In fact, as said explicitly in [12], each n.f. can be split into nested contexts such that each one of these belongs either to the subset considered here or to the subset considered in [11] (with an obvious generalization of the definitions from n. f. s. to contexts). In this section we will study n. f. s. such that, in the corresponding trees, the replaceable variables which occur in labels of non-terminal nodes are bound in the prefixed sequence of the same label. More precisely, we will study the complexity of applications of levelled n. f. s., defined in the following way:

DEFINITION 8: Let $\tau$ be the tree corresponding to a n. f. $N$. $N$ is *levelled* iff in $\tau$ all replaceable variables which occur in labels of nodes of level $i$ are $i$-bound for $i>0$.

*Example* 12: The n. f. of example 1 is non-levelled, since the variable y, which is 2-bound, occurs in the label of a node of level 1. On the contrary, the n. f. :

$$M \equiv \lambda z.z(\lambda y.d(y(\lambda t.e(yf)))(y(\lambda u.gu))) \in \mathscr{F}_2$$

is levelled. This may be easily verified on the tree representation of M (see fig. 3).

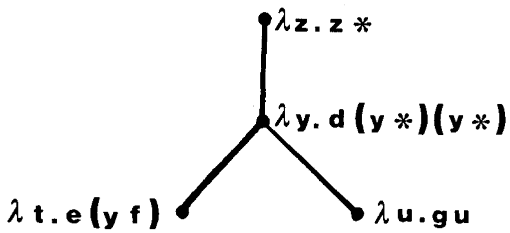


Fig. 3. — Tree which represents the n. f. of example 12.

*Example* 13: From definition 9, it follows that:

– if a n.f.N belongs to $\mathscr{F}_0$, then N is levelled;

– if $Y_1, \ldots, Y_r$ are r levelled n.f.s., which do not contain the variables $x_1, \ldots, x_s$, then $N \equiv \lambda x_1 \ldots x_s . x_i Y_1 \ldots Y_r$ is a levelled n.f. ($s \geqq 1$, $r \geqq 1$, $1 \leqq i \leqq s$).

REMARK 4: From definition 8 it clearly follows that, if $N$ is a levelled n. f., in its representing tree all the proper subtrees represent levelled n. f. s. too.

In what follows we will consider n. f. s. such that we must reduce at most one abstraction for each prefixed sequence. Hence we will give the following definition:

DEFINITION 9: A n. f. $N$ is *monadic* iff all components of $N$, whose head is replaceable, have at most one main argument with at most one variable bound in its prefixed sequence.

*Example* 14: The n. f. M of example 12 is monadic.

REMARK 5: From the given definition it follows that, if $N$ is a monadic n. f., in its representing tree all the subtrees represent monadic n. f. s. too.

Now we try to represent the first reductions necessary to reach the n. f. of the complete application $NM$, when $M$ is levelled and monadic, by extending the tree representation to λ-terms not in n. f. In such a way we describe a reduction strategy which is a slight modification of the leftmost-innermost one.

If $N \equiv \lambda x . \overline{N}$, then $NM \geqq_1 \overline{N}[x/M]$. We can represent $\overline{N}[x/M]$ simply by replacing, in the tree $\tau$ that represents $\overline{N}$, the occurrences of $x$ by $M$. So we will obtain a tree that, in general, does not represent a n. f., but it is an useful tool for studying next reductions. Since $N$ is levelled, $x$ can occur only in the terminal nodes and in the root of $\tau$. If, in $\tau$, the labels of terminal nodes were $T_1, \ldots, T_q$, then there exists a context $C[\ ,\ \ldots,\ ]$ such that $\overline{N} \equiv C[T_1, \ldots, T_q]$. In the tree of $\overline{N}[x/M]$ first we reduce to n. f. these labels. This corresponds to fill the missing components of $C[\ ,\ \ldots,\ ][x/M]$ with the n.f.s. of $(\lambda x . T_i) M$ for $1 \leqq i \leqq q$. Then we obtain a tree such that all labels of its nodes, except the root, are in n. f. In the label of the root there is a certain number, say $p$, of functional occurrences of $M$; each of these ones will be applied to an $*$ which represents a n. f. Let $Z_1, \ldots, Z_p$ be these n. f. s., which we will call *primary n. f. s.* of the application $NM$. Then the problem of reaching the n. f. of $NM$ is reduced to the reaching the n. f. of:

$$MZ_1, \ldots, MZ_p .$$

We can so compute the complexity of $NM$ as a function of the complexities of $MZ_i$ ($1 \leqq i \leqq p$).

In the following lemma we will prove that $Z_1, \ldots, Z_p$ are levelled and monadic n. f. s., which belong to $\mathcal{N}_\omega$ and whose level is less than the level of $N$.

LEMMA 12: *For a complete application $NM$, where $N$ is a levelled and monadic n. f., the primary n. f. s. are levelled, monadic n. f. s. belonging to $\mathcal{N}_\omega$ and their levels are less than the level of $N$.*

The proof of this lemma is given in the appendix.

Now we will introduce some parameters for a levelled n. f. which will be useful for computing the complexity of $NM$, when $N$ and $M$ are n. f. s. satisfying suitable conditions.

DEFINITION 10: Let $N$ be a levelled n. f., and let $\tau$ be the tree which represents it, of level $l$. Let us define, in $\tau$:

– $\mu_i^{(N)}$ number of occurrences of $i$-bound variables at level $i$ $(1 \leqq i \leqq l)$;

– $n^{(N)}$ maximum number of groups of nested occurrences of replaceable variables in the labels of a node at level 0;

– $\varphi_i^{(N)}$ maximum number of occurrences of $i$-bound variables within a group in labels of nodes at level 0 $(0 \leqq i \leqq l)$.

$\vec{\mu}^{(N)}$ and $\vec{\varphi}^{(N)}$ will denote respectively the vectors $\langle \mu_1^{(N)}, \ldots, \mu_l^{(N)} \rangle$ and $\langle \varphi_0^{(N)}, \varphi_1^{(N)}, \ldots, \varphi_l^{(N)} \rangle$.

REMARK 6: About the parameters of definition 10, we notice that:

– all $\mu_i^{(N)}$ for $1 \leqq i \leqq l$ are positive;

– if $n=0$, then $\varphi_i^{(N)}=0$ for $0 \leqq i \leqq l$; if $n \neq 0$, there exists at least one index $j (0 \leqq j \leqq l)$ such that $\varphi_j^{(N)} > 0$;

– if $N$ is a monadic n. f., then the number of terminal nodes in the tree which represents it is less than or equal to $\mu_1^{(N)}$.

*Example* 15: In the n. f. M of example 12, the parameters just defined have the following values:

$$\mu_2^{(M)}=1, \qquad \varphi_2^{(M)}=0,$$
$$\mu_1^{(M)}=2, \qquad \varphi_1^{(M)}=1,$$
$$n^{(M)}=1, \qquad \varphi_0^{(M)}=1.$$

The following lemma computes a maximum for the values of the parameters, given in definition 11, in the primary n. f. s. of the complete application $NM$, where $N$ is a levelled and monadic n. f.

LEMMA 13: *If $Z$ is a primary n. f. of the complete application $NM$, where $N \in \mathcal{F}_l$ and it is a levelled and monadic n. f., the following disequalities hold:*

$$\mu_i^{(Z)} \leqq \mu_i^{(N)} \qquad (1 \leqq i \leqq l-1),$$
$$n^{(Z)} \leqq n^{(N)} . [\rho^{(M)}]^{\varphi_l^{(N)}},$$
$$\varphi_i^{(Z)} \leqq \varphi_i^{(N)} \qquad (0 \leqq i \leqq l-1),$$

*where* $\rho^{(M)}$ *is the number of occurrences in M of variables bound in its prefixed sequence.*

The proof of this lemma is given in the appendix.

Now we are able to compute the complexity of $NM$, where $N, M \in \mathcal{N}_\omega$ and they are levelled and monadic n. f. s.

*Example* 16: If we choose $N \equiv \lambda x . a(x(\lambda u . b(uc)))(x(\lambda v . d(v(ex)))) \in \mathcal{F}_1$ and M as in example 12, we obtain an application satisfying the given requests. The tree representation of N is given in figure 4.

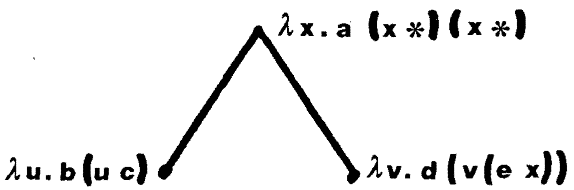


**Fig. 4. — Tree which represents the n. f. of example 16.**

We define now a function $\chi$ of the given parameters, which we will prove to maximize the complexity of the application $NM$. Let $\vec{p}$ denote a vector, $\vec{p}/$denote the vector obtained from $\vec{p}$ by erasing the last component, and $v.\vec{p}$ denote the vector obtained by multiplying each component of $\vec{p}$ by the constant $v$.

DEFINITION 11: Let $l_1, l_2, n_1, n_2$ be integers, $\vec{p}, \vec{q}$ be respectively the vectors $\langle p_1, \ldots, p_{l_1} \rangle$ and $\langle q_0, \ldots, q_{l_1} \rangle$; $\vec{s}, \vec{t}$ be respectively the vectors $\langle s_1, \ldots, s_{l_2} \rangle$ and $\langle t_0, \ldots, t_{l_2} \rangle$.

We define:

$$\chi\{l_1; l_2; n_1; n_2; \vec{p}; \vec{q}; \vec{s}; \vec{t}\} = \textit{if } l_1 = 0 \textit{ then } 1 + n_1 . q_0 \textit{ else}$$
$$1 + n_1 . p_1 . q_{l_1} + p_{l_1} . \chi\{l_2; l_1 - 1; n_2; n_1 . [s_{l_2} + s_1 . n_2 . t_{l_2}]^{q_{l_1}}; \vec{s}; \vec{t}; \vec{p}/; \vec{q}/\}.$$

We notice that the function $\chi$ of definition 11 is a monotonic function in respect to all arguments, since it is built up by operations of sum, product and [ ] (as defined at page 268) applied to non-negative integers. Moreover we notice that, if $l_2=0$, $s_{l_2}$ and $s_1$ are undefined, but in this case the value of $\chi$ is independent of that of its fourth argument.

THEOREM 4: *Let $N, M \in \mathcal{N}_\omega$ be levelled and monadic, and let the levels of the corresponding trees be respectively $l$ and $j$. Then if one follows the described reduction strategy:*

$$\mathscr{C}[NM] \leqq \chi\{l; j; n^{(N)}; n^{(M)}; \vec{\mu}^{(N)}; \vec{\varphi}^{(N)}; \vec{\mu}^{(M)}; \vec{\varphi}^{(M)}\}.$$

*Proof:* First we prove that the given expression is an upperbound for the complexity of $NM$. We will make this proof by induction on the sum of the levels of $N$ and $M$.

*First step:* If $l=0$, we can apply theorem 3, taking into account that:

- $0 \leqq k, \nu, \delta \leqq 1$ since $N$ and $M$ are monadic n. f. s.;
- $n = n^{(M)}$ by definition;
- $\sum_{r=0}^{0} \sigma_{r+1} = \varphi_0^{(N)}$ by definition;
- $\prod_{i=1}^{0} [\rho_i]^{\sigma_i} = 1$.

the complexity of $NM$ is then upperbounded by:

$$1 + n^{(N)} . \varphi_0^{(N)}.$$

*Inductive step:* Let us suppose that the theorem is true for $l+j \leqq u$. We must prove it for $l+j=u+1$.

If $l=0$, we are in the case studied in the first step. Otherwise, let $N \equiv \lambda x . \overline{N}$; obviously $NM \geqq_1 \overline{N}[x/M]$. To obtain the n. f. of $NM$, we can execute, in the tree which represents $\overline{N}[x/M]$, the necessary reductions in the following order:

1) the functional occurrences of $M$ in the terminal nodes;
2) the applications $MZ$, for every primary n. f. $Z$ of $NM$.

Let us compute how many reductions are effected in each one of the previous steps:

1) one reduction for every functional occurrence of $M$ in the terminal nodes of the tree which represents $\overline{N}[x/M]$. The terminal nodes are at most $\mu_1^{(N)}$, and each one contains at most $n^{(N)} . \varphi_1^{(N)}$ functional occurrences of $M$. Then the number of effected reductions is less than or equal to:

$$\mu_1^{(N)} . n^{(N)} . \varphi_l^{(N)};$$

2) there are at most $\mu_l^{(N)}$ applications of the shape $MZ$. Each of these applications satisfies, by lemma 12, the conditions of the present theorem. The level of $Z$ is less than or equal to $l-1$, then the sum of levels of $M$ and $Z$ is less than or equal to $u$, and by inductive hypothesis the complexity of their application is maximized by:

$$\mathscr{C}[MZ] \leqq \chi\{j;\ l-1;\ n^{(M)};\ n^{(Z)};\ \vec{\mu}^{(M)};\ \vec{\varphi}^{(M)};\ \vec{\mu}^{(Z)};\ \vec{\varphi}^{(Z)}\}.$$

The maximum values of the parameters of $Z$ as given in lemma 13, taking into account that $\rho^{(M)} \leqq \mu_j^{(M)} + \mu_1^{(M)}.n^{(M)}.\varphi_j^{(M)}$, are:

$$\mu_i^{(Z)} \leqq \mu_i^{(N)} \qquad (1 \leqq i \leqq l-1),$$
$$n^{(Z)} \leqq n_l^{(N)}.[\mu_j^{(M)} + \mu_1^{(M)}.n^{(M)}.\varphi_j^{(M)}]^{\varphi_l^{(N)}};$$
$$\varphi_i^{(Z)} \leqq \varphi_i^{(N)} \qquad (0 \leqq i \leqq l-1).$$

Since $\chi$ is a monotonic function, we obtain:

$$\mathscr{C}[MZ] \leqq \chi\{j;\ l-1;\ n^{(M)};\ n^{(N)}.[\mu_j^{(M)} + \mu_1^{(M)}.n^{(M)}.\varphi_j^{(M)}]^{\varphi_l^{(N)}};\ \vec{\mu}^{(M)};\ \vec{\varphi}^{(M)};\ \vec{\mu}^{(N)}/;\ \vec{\varphi}^{(N)}/\}.$$

Taking into account points 1 and 2 we have:

$$\mathscr{C}[NM] \leqq 1 + \mu_1^{(N)}.n^{(N)}.\varphi_l^{(N)} + \mu_l^{(N)}.\chi\{j;\ l-1;\ n^{(N)};\ n^{(N)} \times [\mu_j^{(M)} + \mu_1^{(M)} n^{(M)}.\varphi_j^{(M)}]\,\varphi_l^{(N)};\ \vec{\mu}^{(M)};\ \vec{\varphi}^{(M)};\ \vec{\mu}^{(N)}/;\ \vec{\varphi}^{(N)}/\}.$$

To prove that this expression is really a maximum for the complexity it is sufficient to choose:

$$V\{l, \vec{q}\} \equiv$$
$$\equiv (\underbrace{x_l(c \ldots (x_l}_{q_l}(c(\underbrace{x_{l-1}(c \ldots (x_{l-1}}_{q_{l-1}}(c \ldots (\underbrace{x_0(c \ldots (x_0\, c)}_{q_0} \ldots)) \ldots)) \ldots)))) \ldots)),$$

$$U\{l;\ 0;\ n;\ \vec{\varepsilon};\ \vec{q}\} \equiv \lambda x_0 . a \underbrace{V\{l;\ \vec{q}\} \ldots V\{l, \vec{q}\}}_{n}$$

($\vec{\varepsilon}$ denotes the empty vector)

$$U\{l;\ i;\ n;\ \vec{p}^{(i)};\ \vec{q}\} \equiv$$
$$\equiv \lambda x_i . a \underbrace{(x_i U\{l;\ i-1;\ n;\ \vec{p}^{(i)}/;\ \vec{q}\}) \ldots (x_i U\{l;\ i-1;\ n;\ \vec{p}^{(i)}/;\ \vec{q}\})}_{p_i} \qquad (1 \leqq i \leqq l)$$

($\vec{p}^{(i)}$ denotes the vector $\langle p_1, \ldots, p_i \rangle$) and

$$N \equiv U\{l; l; n_1; \vec{p}; \vec{q}\},$$
$$M \equiv U\{j; j; n_2; \vec{s}; \vec{t}\}.$$

Then, following the described reduction strategy:

$$\mathscr{C}[NM] = \chi\{l; j; n_1; n_2; \vec{p}; \vec{q}; \vec{s}; \vec{t}\}. \quad \square$$

*Example* 17: If the values of the parameters are the following:

$$l=1, \quad n_1=1, \quad \vec{p}=\langle 1 \rangle, \quad \vec{q}=\langle 2, 1 \rangle,$$
$$j=0, \quad n_2=1, \quad \vec{s}=\vec{\varepsilon}, \quad \vec{t}=\langle 2 \rangle$$

we have:

$$V\{1; \langle 2, 1 \rangle\} \equiv (x_1(c(x_0(c(x_0 c))))),$$
$$U\{1; 0; 1; \vec{\varepsilon}; \langle 2, 1 \rangle\} \equiv \lambda x_0 . a(x_1(c(x_0(c(x_0 c))))),$$
$$N \equiv U\{1; 1; 1; \langle 1 \rangle; \langle 2, 1 \rangle\} \equiv \lambda x_1 . a(x_1 \lambda x_0 . a(x_1(c(x_0(c(x_0 c))))))$$

and:

$$V\{0; \langle 2 \rangle\} \equiv (y_0(d(y_0 d))),$$
$$M \equiv U\{0; 0; 1; \vec{\varepsilon}; \langle 2 \rangle\} \equiv \lambda y_0 . e(y_0(d(y_0 d))).$$

Then:

$$NM \geqq_1 a(MU\{1; 0; 1; \vec{\varepsilon}; \langle 2, 1 \rangle\}[x_1/M])$$

where:

$$U\{1; 0; 1; \vec{\varepsilon} \langle 2, 1 \rangle\}[x_1/M] \equiv \lambda x_0 . a(M(c(x_0(c(x_0 c))))) \geqq_1,$$
$$\lambda x_0 . a(e(c(x_0(c(x_0 c)))(d(c(x_0(c(x_0 c)))d)))) \equiv M^*,$$
$$MM^* \geqq_1 e(M^*(d(M^* d))),$$
$$M^* d \geqq_1 a(e(c(d(c(dc)))(d(c(d(c(dc)))d)))) \equiv \overline{M},$$
$$M^*(d\overline{M}) \geqq_1 a(e(c(d\overline{M}(c(d\overline{M}c)))(d(c(d\overline{M}(c(d\overline{M}c)))d)))) \equiv M'.$$

The n.f. of NM is (a M').

It is easy to verify that $\chi\{1; 0; 1; 1; \langle 1 \rangle; \langle 2, 1 \rangle; \vec{\varepsilon}; \langle 2 \rangle\} = 5$ which is the total number of effected reductions.

We use now previous results in order to study the complete application $NM$ where $N$ is a levelled and monadic n. f., and $M$ is a simple n. f. The simple n. f. s. are defined as follows:

DEFINITION 13: A n. f. $M$ is said *simple* when in $M$ every ω-replaceable variable is head of a component whose main arguments are all λ-free n. f. s., and moreover, if two occurrences of ω-replaceable variables are nested in $M$, then the innermost occurrence is in argument position.

Clearly any simple n. f. belongs to $\mathcal{F}_0$.

*Example* 18: $M \equiv \lambda z.z(a(zz))(bz)$ is a simple n. f.

It is interesting to study complete applications whose arguments are simple n. f. s., since there are some numerical systems in which the numbers are represented by simple n. f. s.

Let us remind the definition of numerical system, according to [10] (p. 212). A numerical system is a class of λ-terms $\mathcal{M} \equiv \{[(0)], [(1)], \ldots\}$ to be represented by a quadruple of λ-terms $[(0)]$, $[(\sigma)]$, $[(\pi)]$, $[(\delta)]$ called respectively zero, successor, predecessor, discriminator of the zero. More precisely, if the following relations hold for every integer $n \geqq 0$:

(*a*) $[(n+1)] = [(\sigma)]\,[(n)]$;

(*b*) $[(n)] = [(\pi)]\,[(n+1)]$;

(*c*) $[(\delta)]\,[(0)] = \mathbf{O} \equiv \lambda xy.y$,

(*d*) $[(\delta)]\,[(n+1)] = \mathbf{K} \equiv \lambda xy.x$ or $\mathbf{I} \equiv \lambda x.x$,

the set $\mathcal{M}$, whose elements may be built up from (*a*) by iteration

$$[(n)] = \underbrace{[(\sigma)](\ldots([(\sigma)]}_{n}[(0)])\ldots),$$

is isomorphic with the set of integers.

*Example* 19: We may give two examples of numerical systems such that all $[(n)]$ are simple n. f. s. $(n \geqq 0)$.

If we choose:

$$[(0)] \equiv \mathbf{C} \equiv \lambda xyz.xzy, \; [(\sigma)] \equiv \mathbf{B} \equiv \lambda xyz.x(yz),$$

$$[(\pi)] \equiv \lambda x.x\mathbf{I}, \qquad [(\delta)] \equiv \lambda x.x\mathbf{I}(x\mathbf{II})\mathbf{KKK},$$

then:

$$[(\ )] \equiv \lambda x_1 \ldots x_n uvw.x_1 \ldots x_n uwv.$$

If we choose:

$$[(0)] \equiv \mathbf{O}, \qquad [(\sigma)] \equiv \lambda uvx.uvxv,$$

$$[(\pi)] \equiv \lambda uvx.uv(\lambda y.x), \qquad [(\delta)] \equiv \lambda x.x\mathbf{IO},$$

then:

$$[(n)] \equiv \lambda xy.y\underbrace{x\ldots x}_{n}.$$

Now we will compute how many reductions we need to reach the n.f. of $NM$, when $N \equiv \lambda x.\overline{N}$ and $M$ satisfy previous conditions. We will reduce in the same order as before, i. e., after the reduction of the prefixed sequence of $N$ and of the occurrences of $M$ in the terminal nodes of the tree which represents $\overline{N}[x/M]$, we must reduce a certain number of applications of the shape $MZ$, where $M$ is a simple n.f. and $Z$ is a primary n.f. of $NM$. By lemmas 12 and 13, $Z$ is a n.f. $\in \mathcal{N}_\omega$, levelled, monadic and its parameters can be maximized in function of the parameters of $N$ and $M$. To study the complexity of $MZ$, we split the parameter $\rho^{(M)}$ (defined in lemma 13) into three addenda, so defined:

DEFINITION 14: Let us define, in a simple n.f. $M \equiv \lambda y . \overline{M}$:

— $\eta_1^{(M)}$ number of functional occurrences of $y$ in $\overline{M}$ such that their first main argument is $y$;

— $\eta_2^{(M)}$ number of functional occurrences of $y$ in $\overline{M}$ whose first main argument is different from $y$;

— $\eta_3^{(M)}$ number of argument occurrences of $y$ in $\overline{M}$.
Obviously $\rho^{(M)} = \eta_1^{(M)} + \eta_2^{(M)} + \eta_3^{(M)}$ and $\eta_1^{(M)} \leqq \eta_3^{(M)}$.

*Example* 20: In $M$ as defined in Example 18 we have:

$$\rho^{(M)} = 4, \qquad \eta_1^{(M)} = 1, \qquad \eta_2^{(M)} = 1, \qquad \eta_3^{(M)} = 2.$$

If $M \equiv \lambda y . \overline{M}$, $MZ \geqq_1 \overline{M}[y/Z]$. In $\overline{M}[y/Z]$ there will be exactly $\eta_1^{(M)}$ redexes of the shape $ZZ$ and $\eta_2^{(M)}$ redexes of the shape $ZR$ where $R$ is a $\lambda$-free $\lambda$-term. The complexity of $ZZ$ can be maximized by the function $\chi$. The redexes of the shape $ZR$ are reduced to n.f. in one step, if $R$ is in n.f. We define now the function $\Gamma$, that we will prove to maximize the complexity of $NM$.

DEFINITION 15: Let $l, n, m_1, m_2, m_3$ be integers, $\vec{p}, \vec{q}$ be respectively the vectors $\langle p_1, \ldots, p_l \rangle$ and $\langle q_0, q_1, \ldots, q_l \rangle$. Then:

$$\Gamma\{l;\, n;\, \vec{p};\, \vec{q};\, m_1;\, m_2;\, m_3\} = \textit{if } l=0 \quad \textit{then } 1 + n . q_0$$
$$\textit{else } 1 + p_1 . n . q_l + p_l . [1 + m_1 . \chi\{l-1;\, l-1;\, n . [m_1 + m_2 + m_3]^{q_l};$$
$$n . [m_1 + m_2 + m_3]^{q_l};\, \vec{p}/;\, \vec{q}/;\, \vec{p}/;\, \vec{q}/\} + m_2].$$

THEOREM 5: *Let $N \in \mathcal{N}_\omega$, $\mathcal{F}_l$ be a levelled and monadic n.f., and let $M$ be a simple n.f. Then if one follows the described reduction strategy:*

$$\mathscr{C}[NM] \leqq \Gamma\{l;\, n^{(N)};\, \vec{\mu}^{(N)};\, \vec{\varphi}^{(N)};\, \eta_1^{(M)};\, \eta_2^{(M)};\, \eta_3^{(M)}\}.$$

*Proof:* First we prove that the given expression is an upperbound for the complexity of $NM$. In the case $l=0$, we obtain a value which upperbounds the complexity (by th. 3). In the case $l>0$ (by the argument given in the proof of theorem 4) after at most $1 + \mu_1^{(N)} . n^{(N)} . \varphi_l^{(N)}$ reductions, we have to reduce at most $\mu_l^{(N)}$ applications of the shape $MZ$, where $Z$ is a primary n.f. of $NM$. In each $MZ$, after the reduction of the prefixed sequence of $M$, we obtain $\eta_1^{(M)}$ redexes of the shape $ZZ$ and $\eta_2^{(M)}$ redexes of the shape $ZR$ where $R$ is a λ-free λ-term.

By theorem 4, each redex $ZZ$ reaches its n.f. after at most

$$\chi\{l-1;\ l-1;\ n^{(N)}.[\eta_1^{(M)}+\eta_2^{(M)}+\eta_3^{(M)}]^{\varphi(N)};\ n^{(N)}.[\eta_1^{(M)}+\eta_2^{(M)}+\eta_3^{(M)}]^{\varphi(N)};\ \vec{\mu}^{(M)}/;\ \vec{\varphi}^{(N)}/;\ \vec{\mu}^{(N)}/;\ \vec{\varphi}^{(N)}/\}$$

reductions, where the parameters of $Z$ are computed according to lemma 13. Therefore, if we use the innermost reduction algorithm, each λ-free λ-term $R$, in the application $ZR$, is now in n.f. Then for each application $ZR$ we must effect only one reduction.

It may be verified that, through the choice:

$$M\equiv\lambda z.\underbrace{z(d\ldots z}_{m_2}(d\underbrace{(zz)\ldots(zz)}_{m_1}\ \underbrace{z\ldots z}_{m_3-m_1})\ldots)$$

and

$$N\equiv U\{l;\ l;\ n;\ \vec{p};\ \vec{q}\}$$

as defined in the proof of theorem 4, we have:

$$\mathscr{C}[NM]=\Gamma\{l;\ n;\ \vec{p};\ \vec{q};\ m_1;\ m_2;\ m_3\}.\quad\square$$

*Example* 21: If we choose N as in example 17, and $M\equiv\lambda z.z(d(zz)z)$ then:

$$NM\geqq_1 a(M\lambda x_0.a(M(c(x_0(c(x_0 c)))))),$$
$$\lambda x_0.aM(c(x_0(c(x_0 c))))\geqq_1\overline{N}$$

where

$$\overline{N}\equiv\lambda x_0.a(c(x_0(c(x_0 c)))(d(c(x_0(c(x_0 c)))\ (c(x_0(c(x_0 c)))))(c(x_0(c(x_0 c)))))).$$

$$M\overline{N}\geqq_1\overline{N}(d(\overline{N}\,\overline{N})\overline{N}),$$

$$\overline{N}\,\overline{N}\geqq_1 a(c(\overline{N}(c(\overline{N}c)))(d(c(\overline{N}(c(\overline{N}c)))(c(\overline{N}(c(\overline{N}c)))))(c(\overline{N}(c(\overline{N}c))))))\equiv\overline{\overline{N}}.$$

In $\overline{\overline{N}}$ there are 4 applications $\overline{N}$ c; each one reduces in 1 step to its n.f. N′. Then we must reduce 4 applications $\overline{N}$(c N′); each one reduces in 1 step to its n.f. N″.

Let N* be the n.f. of $\overline{N}\overline{N}$, i.e. $N^*\equiv a(cN''(d(cN'')(cN''))(cN''))$. $\overline{N}(dN^*\overline{N})$ reduces in 1 step to its n.f. $\overline{N}'$.

Clearly the n.f. of NM is $(a\overline{N}')$, and the number of effected reductions is 13.

It is easy to verify that: $\Gamma\{1;\ 1;\ \langle 1\rangle;\ \langle 2,\ 1\rangle;\ 1;\ 1;\ 2\}=13$.

## REFERENCES


1. S. K. Abdali, *A Lambda Calculus Model of Programming Languages. I. Simple Constructs and II. Jumps and Procedures,* J. Comp. Languages, Vol. 1, 1976, pp. 287-301 and 303-320.
2. H. Barendregt, *Normed Uniformity Reflexive Structures,* in C. Böhm, Ed., *λ-Calculus and Computer Science Theory,* Lecture Notes in Computer Science, No. 37, Springer-Verlag, 1975, pp. 272-286.
3. H. Barendregt, J. Bergstra, J. W. Klop and M. Volken, *Degrees, Reductions and Representability in Lambda Calculus,* Preprint, University of Utrecht, Department of Mathematics, 1976.
4. C. Batini and A. Pettorossi, *On Recursiveness in Weak Combinatory Logic,* in C. Böhm, Ed., *λ-Calculus and Computer Science Theory,* Lecture Notes in Computer Science, No. 37, Springer-Verlag, 1975, pp. 297-311.
5. C. Böhm and M. Dezani-Ciancaglini, *λ-terms as Total or Partial Functions on Normal Forms* in C. Böhm, Ed., *λ-Calculus and Computer Science Theory,* Lecture Notes in Computer Science, Vol. 37, Springer-Verlag, 1975, pp. 96-121.
6. M. Blum, *On the Size of Machines,* Information and Control, Vol. 11, 1967, pp. 257-265.
7. R. Canal, *Complexité de la réduction en logique combinatoire,* R.A.I.R.O., Informatique théorique, Vol. 12, 1978, pp. 339-367.
8. R. Canal and J. Vignolle, *Calculs finis et infinis dans les termes combinatoires* in B. Robinet, Ed., *λ-Calcul et Sémantique Formelle des langages de programmation,* L.I.T.P.-E.N.S.T.A., 1979, pp. 109-130.
9. H. B. Curry and R. Feys, *Combinatory Logic,* Vol. I, North-Holland, Amsterdam, 1958.
10. H. B. Curry, J. R. Hindley and J. P. Seldin, *Combinatory Logic,* Vol. II, North-Holland, Amsterdam, 1972.
11. M. Dezani-Ciancaglini and S. Ronchi Della Rocca, *Computational Complexity and Structure of λ-terms* in B. Robinet, Ed., *Programmation,* Dunod, Paris, 1976, pp. 160-181.
12. M. Dezani-Ciancaglini, S. Ronchi Della Rocca and L. Saitta, *Complexité élémentaire dans le λ-calcul* in B. Robinet, Ed., *λ-calcul et Sémantique formelle des langages de programmation,* L.I.T.P.-E.N.S.T.A., 1979, pp. 183-212.
13. P. J. Landin, *A Correspondence Between Algol* 60 *and Church's Lambda Notation,* Comm. A.C.M., Vol. 8, 1965, pp. 89-158.
14. Y. Moscovakis, *Axioms for Computation Theories-first draft* in R. Gandy and M. Yates, Eds., *Logic Colloquium*'69, North Holland, Amsterdam, 1971, pp. 199-255.
15. G. Plotkin, *A set Theoretical Definition of Application,* School of A.I., Memo, M.I.P.-R-95, Edinburgh, 1975.
16. B. Robinet, *Contribution à l'étude de réalités informatiques,* Thèse, Université Paris-VI, 1974.
17. D. Scott, *Continuous Lattices,* Lecture Notes in Mathematics, No. 274, Springer-Verlag, 1972, pp. 97-136.
18. D. Scott, *Data Types as Lattices,* S.I.A.M. J., Comp., Vol. 5, 1976, pp. 522-587.
19. S. Stenlund, *Combinators, λ-terms and Proof Theory,* D. Reidel Publ. Company, Dordrecht-Holland, 1972.
20. C. P. Wadsworth, *Relation between Computational and Denotational Properties for Scott's $D_\infty$-Models of the Lambda Calculus,* S.I.A.M. J. Comp., Vol. 5, 1976, pp. 488-521.

# APPENDIX

The proofs of lemmas 8, 9, 12 and 13 may be easily obtained from the proof of the following:

PROPERTY 1: Let $N \equiv \lambda x . \overline{N} \in \mathcal{N}_\omega$ and let $\tau$ be the tree which represents $\overline{N}$. Let $x$ occur only in labels of terminals nodes of $\tau$. We claim that, for an arbitrary n.f. $M$, the n.f. $N'$ of $NM$ is such that:

1) if $\overline{N} \in \mathcal{F}_l$ then $N' \in \mathcal{F}_l$;

2) if $y$ is any ω-replaceable variable in $N$, then the maximum number of occurrences of $y$ within a group in $N'$ is equal to or less than that one in $N$;

3) if $\overline{N}$ is monadic, then $N'$ is monadic too;

4) if $\overline{N}$ is levelled, then $N'$ is levelled too.

*Proof:* Instead of proving conditions 1, 2, 3 and 4, we will prove conditions 2, 3 and the following condition 5, which obviously implies conditions 1 and 4:

5) the tree $\tau'$ of $N'$ is obtained from the tree $\tau$ simply by replacing (without α-reductions) each label $T$ of a terminal node with the n.f. of $T[x/M]$.

We make this proof by induction on the number $t$ of occurrences of $x$ in $\overline{N}$.

*First step:* $t = 0$ means $N' \equiv \overline{N}$ and the conditions obviously hold.

*Inductive step:* Let these conditions be true for $t \leq v$ and let us prove them for $t = v + 1$. Let $P$ be the n.f. obtained from $\overline{N}$ by replacing one of the outermost occurrences of $x$ by a variable (say $a$) which does not occur in $\overline{N}$. Then by construction: $NM \equiv (\lambda a . (\lambda x . P) M) M$.

Let us observe, about $N$, that:

(i) each occurrence of $x$ in $N$ is nested with occurrences of variables which are all non-ω-replaceable in $N$, since $N \in \mathcal{N}_\omega$;

(ii) the main arguments of components whose head is $x$ are λ-free n.f.s., since $x$ occurs only in labels of terminal nodes in $\tau$.

About the n.f. $Q$ of $(\lambda x . P) M$, we observe that:

(iii) $Q$ contains exactly one occurrence of $a$, since $a$ replaces one of the outermost occurrences of $x$;

(iv) $a$ occurs in the label of a terminal node in the tree representation of $Q$, since $x$ occurs only in labels of terminal nodes of $\tau$ and condition 5 must be satisfied by inductive hypothesis;

(v) observations (i) and (ii) are true again for the occurrence of $a$ in $Q$, i.e. a is nested only with variables which are non-ω-replaceable in $\lambda a.Q$ and it has λ-free main arguments, since $a$ replaces one of the outermost occurrence of $x$.

From observations (iii) (iv) and (v) it follows that the label of exactly one terminal node in the tree which represents $Q$ has a component of the shape: $VZ$ where $V$ is λ-free, has a head which is non-ω-replaceable in $\lambda a.Q$, $Z\equiv\lambda y_1 \dots y_p.aZ_1 \dots Z_q$ and $Z_1, \dots, Z_q$ are λ-free n.f.s. whose heads are non-ω-replaceable in $Q$; i.e. there exists a context $\overline{C}[\ ]$ such that $Q\equiv\overline{C}[aZ_l \dots Z_q]$. To obtain $N'$ from $(\lambda a.Q)M$ we have one component to reduce, i.e.:

$$MZ_1 \dots Z_q \tag{6}$$

provided that $q\geqq 1$ and $M$ is not λ-free. The n.f. $M'$ of (6) is reached simply by reducing $\min[q, \neq(M)]$ variables bound in the prefixed sequence of $M$. Then $N'$ may be obtained simply by filling with $M'$ the missing component of $\overline{C}[\ ]$, i.e. $N'\equiv\overline{C}[M']$. By inductive hypothesis $Q\equiv\overline{C}[aZ_1 \dots Z_q]$ satisfies conditions 2, 3 and 5. Then it is sufficient to prove that the replacement of a $Z_1 \dots Z_q$ by $M'$ does not affect these conditions. To this aim we observe that:

(vi) the variables of $M'$ are all non-ω-replaceable in $N'$. In fact, the (possible) rightmost $\neq(M)-q$ variables bound in the prefixed sequence of $M'$ are non-ω-replaceable in $N'$ since the head of $V$ is non-ω-replaceable in $\lambda a.Q$.

(vii) $M'$ may have components of the shape: $Z_l M_1 \dots M_r$ $(1\leqq l\leqq q)$ where some $M_i (1\leqq i\leqq r)$ may have as components some $Z_j (1\leqq j\leqq q)$. The variables which occur in $M_1, \dots, M_r$ are then nested only with the head of $Z_l$, which is non-ω-replaceable in $N'$.

From observation (iv) and from the sketch of reduction of $Q$ to n.f., it follows that condition 5 is satisfied. Observation (vii) implies condition 2, and observations (vi) and (vii) together imply condition 3. □

*Proof of lemma* 8: Let $L_j$ $(0\leqq j\leqq k)$ be defined as in the next, i.e. $N\equiv\lambda x_1 \dots x_k.L_0$ and $L_{j+1}$ be the n.f. of $(\lambda x_{j+1}.L_j)M_{j+1} (0\leqq j\leqq k-1)$. We prove the lemma by induction on the index $j$.

*First step:* when $j=0$ the principal redex is $NM_1 \dots M_k$ and the lemma obviously holds.

*Inductive step:* Suppose that the lemma holds for $j=r$, i.e. $\lambda x_{r+1} \dots x_k.L_r \in \mathscr{N}_\omega$, $\mathscr{F}_0$ and $\#(\lambda x_{r+1} \dots x_k.L_r)=k-r$. Then we have $\lambda x_{r+2} \dots x_k.L_{r+1} \in \mathscr{N}_\omega$ (by remark 2), $\in \mathscr{F}_0$ (by condition 1 of property 1) and $\#(\lambda x_{r+2} \dots x_k.L_{r+1})=k-r-1$ since $\lambda x_{r+1} \dots x_k.L_{r+1} \in\ \ _\omega$ implies that its head is free. □

*Proof of lemma* 9: We prove this lemma by induction on the index $j$.

*First step:* When $j=0$ the principal redex is $NM_1 \dots M_k$ and the lemma obviously holds.

*Inductive step:* Suppose that the lemma holds for $j=r$, i.e. in $L_r$ there are at most $\sigma_l$ functional occurrences of $x_l$ inside a group for $r+1 \leqq l \leqq k$. Condition 2 of property 1 assures us that $L_{r+1}$ satisfies this lemma, since $x_l$ for $r+2 \leqq l \leqq k$ are ω-replaceable variables in $\lambda x_{r+1} \dots x_k . L_r$. □

*Proof of lemma* 12: Let $N \equiv \lambda x . \overline{N}$. If we erase the root of the tree representing $\overline{N}$, we obtain a forest of a given number of trees, say $p$. Let $\overline{Z}_1, \dots, \overline{Z}_p$ be the n.f.s. represented by these trees. Clearly $\overline{Z}_1, \dots, \overline{Z}_p$ are monadic, levelled and their levels are less than the level of $N$. The primary n.f.s. of the application $NM$ are the n.f.s. $Z_i$ of $(\lambda x . \overline{Z}_i) M$ $(1 \leqq i \leqq p)$. Then let us consider the application $(\lambda x . \overline{Z}_i) M$ for an arbitrary value of $i$. First of all, we notice that $\lambda x . \overline{Z}_i$ belongs to $\mathcal{N}_\omega$ (*see* remark 1). Then also $Z_i$ belongs to $\mathcal{N}_\omega$ (*see* remark 2). Moreover $N$ levelled implies that $x$ occurs in $\overline{Z}_i$ only at level 0. Then $\lambda x . \overline{Z}_i$ satisfies all hypotheses done in the proof of property 1, and so $Z_i$ has the same level as $\overline{Z}_i$ (condition 1) and it is monadic (condition 3) and levelled (condition 4). □

*Proof of lemma* 13: First we prove the disequalities where $\leqq$ holds.

Let $N \equiv \lambda x . \overline{N}$. As observed in the proof of lemma 12, each redex $(\lambda x . \overline{Z}) M$, whose n.f. $Z$ is a primary n.f. of $NM$, satisfies the hypotheses of property 1. By condition 5 the labels of non-terminal nodes of $\overline{Z}$ and $Z$ coincide, and so $\mu_i^{(Z)} = \mu_i^{(\overline{Z})} \leqq \mu_i^{(N)}$ for $1 \leqq i \leqq l$. Again by condition 5, $Z$ is obtained from $\overline{Z}$ by replacing each label $T$ of a terminal node by the n.f. of $(\lambda x . T) M$. Since $\lambda x . \overline{Z} \in \mathcal{N}_\omega$ then, by remark 1, $\lambda x . T \in \mathcal{N}_\omega$. The application $(\lambda x . T) M$ satisfies all hypotheses of lemma 6 and therefore this lemma maximizes the increaing of variable number. In fact, since in $\lambda x . \overline{Z}$ there are most $n^{(N)} . \varphi_i^{(N)}$ occurrences of $i$-bound variables $(0 \leqq i \leqq l-1)$ at level 0, in $Z$ there will be a most $n^{(N)} . \varphi_i^{(N)} . [\rho^{(M)}]^{\varphi_i^{(N)}}$ occurrences of these variables at level 0. Moreover by condition 2 of property 1 the occurrence number of ω-replaceable variables inside a group cannot increase, and so we obtain $n^{(Z)} \leqq n^{(N)} . [\rho^{(M)}]^{\varphi_i^{(N)}}$ and $\varphi_i^{(Z)} \leqq \varphi_i^{(N)}$.

To prove the disequalities where $=$ holds, it is sufficient to choose $N \equiv U \{ l; l; n; \vec{\mu}^{(N)}; \vec{\varphi}^{(N)} \}$ as defined in the proof of theorem 4. □