E. FACHINI
A. MAGGIOLI-SCHETTINI

A hierarchy of primitive recursive sequence functions

RAIRO. Informatique théorique, tome 13, n° 1 (1979), p. 49-67

<http://www.numdam.org/item?id=ITA_1979__13_1_49_0>
A HIERARCHY OF
PRIMITIVE RECURSIVE SEQUENCE FUNCTIONS (*)

by E. FACHINI and A. MAGGIOLO-SCHETTINI (1)

Communicated by G. AUSSIELLO

Abstract. — In this paper we give a characterization of primitive recursive functions
\( f : \mathbb{N}^r \rightarrow \mathbb{N}^s \) (\( r \geq 0, s > 0 \)) and define a Hierarchy of classes \( J_{a+b}^a \) (\( a, b \geq 0 \)) of these functions by a
syntactic measure of complexity. The behavior of the classes \( J_{a+b}^a \) with respect to different operators is
also analyzed. The classes \( J_{a+b}^a \) coincide with the ones of Cleave’s hierarchy for \( a \geq 2, b \geq 0 \) and give a
refinement of the Meyer-Ritchie hierarchy.

INTRODUCTION

Partial recursive sequence functions, i.e. partial functions of type \( f : \mathbb{N}^r \rightarrow \mathbb{N}^s \),
have been studied by Eilenberg and Elgot and by Germano and Maggiolo-
Schettini (see [6, 7, 8, 9, 10]). In this paper we consider a characterization of
primitive recursive functions \( N^r \rightarrow N^s \) (\( r \geq 0, s > 0 \)), which reduce obviously to
primitive recursive functions when \( s = 1 \). Such functions are obtained by starting
with a finite set of basic functions and taking the closure with respect to
composition, cylindrification and repetition operators.

We consider a hierarchy of length \( \omega^2 \) of primitive recursive sequence functions
in a very simple manner: every class \( J_{a+b}^a \) contains the functions obtained from
the basic one by \( a \) nested repetitions and \( b \) successive compositions. (The idea of
constructing an \( \omega^2 \) hierarchy was suggested by Cleave in [4] where an \( \omega^2 \)
hierarchy of functions \( N^r \rightarrow N \) computed by a register machine is presented and
the equivalence with a characterization of the chain of classes in terms of the
substitutions and recursions occurring in the functions of each class is shown.)

(*) Received June 1978, revised October 1978.

(1) Gruppo Nazionale di Informatica Matematica c/o Istituto di Scienze dell’Informazione
dell’Università Salerno, Italy.

R.A.I.R.O. Informatique théorique/Theoretical Informatics, 0399-0540/1979/49/$ 1.00
© Bordas-Dunod.
We introduce LOOP programs (see Meyer and Ritchie [13]) with \( r \) input variables and \( s \) output variables. We consider the class of functions computed by these programs and prove that it coincides with the class of primitive recursive sequence functions. We define the class \( M_{a+b} \) of LOOP programs as the classes of programs with \( b \) successive subprograms containing at most \( a+1 \) nested loop instructions. It is easily seen that the corresponding classes of functions coincide with the classes \( J_{a+b} \) and then a hierarchy of LOOP programs follows. It can be also shown that these classes are computation time closed. (Note that in [3] Beck introduces a \( \omega^2 \) hierarchy of Meyer and Ritchie LOOP programs inspired by Cleave's idea. The classes of programs are defined here similarly as in Beck's paper but the proof and the point of view are different.)

If we consider the subclasses \( J_{a+b}^L \subseteq J_{a+b} \) of functions \( N^r \rightarrow N \) we can compare our hierarchy with the known hierarchies of primitive recursive functions defined by Axt, Cleave, Grzegorczyk, Meyer-Ritchie (see [2, 4, 11, 13] respectively).

In section 1 primitive recursive sequence functions are defined and the relationship with primitive recursive functions is shown. In section 2 the classes of primitive recursive sequence functions are defined in terms of composition and repetition. In section 3 the proper containment of each class in the following one is shown. In section 4 we introduce LOOP programs and the hierarchy defined in terms of nesting and concatenation of loops and we prove that the corresponding hierarchy of functions coincides with the one of sections 2-3. In section 5 we recall the definition of Axt, Cleave, Grzegorczyk and Meyer-Ritchie hierarchies and compare these hierarchies with our hierarchy (restricted obviously to functions \( N^r \rightarrow N \)). In section 6 we extend some known decidability results to the classes \( J_{a+b}^L \).

A rather complete synthesis of works on complexity classes of functions is in the book by Ausiello (see [1]).

We are grateful to Egon Börger for encouragement at the beginning of our work, to Giorgio Ausiello for many discussions and to Jean-François Perrot for discussions and for bringing the papers by Beck and by Huwig and Clausto our attention.

1. PRIMITIVE RECURSIVE SEQUENCE FUNCTIONS

In this section we introduce a characterization of primitive recursive functions from sequences of natural numbers to sequences of natural numbers, briefly primitive recursive sequence functions.
In the following we will use the letters $x$, $y$, possibly with indices, for natural numbers and the letters $u$, $v$ for tuples of natural numbers without specifying the arity of the tuple when it is clear from the context.

Consider the set of functions

$$
\Sigma = \{ S = \lambda x. (x + 1), \ K = \lambda x, y. (x), \ O = (0) \}.
$$

**Definition 1.1:** The set $\mathcal{F}$ of primitive recursive sequence functions $f : \mathbb{N}^r \to \mathbb{N}^s$, with $r \geq 0$, $s > 0$ is defined as the least set of functions containing $\Sigma$ and closed with respect to the following operators:

1° the composition operator $\lambda f, g. (f \cdot g)$ such that if $f : \mathbb{N}^r \to \mathbb{N}^s$ and $g : \mathbb{N}^s \to \mathbb{N}^t$ then $f \cdot g : \mathbb{N}^r \to \mathbb{N}^t$ and $(f \cdot g)(u) = g(f(u))$;

2° the left cylindrification operator $\lambda f. c$ such that if $f : \mathbb{N}^r \to \mathbb{N}^s$ then $c_f : \mathbb{N}^{r+1} \to \mathbb{N}^{s+1}$ and $c_f(x, u) = x, f(u)$;

3° the right cylindrification operator $\lambda f. c^r$ such that if $f : \mathbb{N}^r \to \mathbb{N}^s$ then $f^r : \mathbb{N}^{r+1} \to \mathbb{N}^{s+1}$ and $f^r(u, x) = f(u)$, $x$;

4° the repetition operator $\lambda f. f^r$ such that if $f : \mathbb{N}^r \to \mathbb{N}^r$ then $f^r : \mathbb{N}^{r+1} \to \mathbb{N}^r$ and $f^r(x, u) = f^x(u) = (f \cdot \ldots \cdot f)(u)$.

Consider the following functions:

1° the functions $\Theta : \mathbb{N}^r \to \mathbb{N}^r$ $(r > 1)$ such that

$$
\Theta(x_1, \ldots, x_r) = (x_2, \ldots, x_r, x_1);
$$

2° the functions $\Delta : \mathbb{N}^r \to \mathbb{N}^{2r}$ $(r > 0)$ such that

$$
\Delta(u) = u, u;
$$

3° the functions $\Theta_i : \mathbb{N}^r \to \mathbb{N}^r$ $(i \leq r$, $r > 1)$ such that

$$
\Theta_i(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_r) = x_i, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_r;
$$

4° the functions $I_i : \mathbb{N}^r \to \mathbb{N}$ $(1 \leq i \leq r)$ such that

$$
I_i(x_1, \ldots, x_r) = x_i;
$$

5° the functions $I : \mathbb{N}^r \to \mathbb{N}^r$ $(r > 0)$ such that

$$
I(u) = u;
$$

6° the functions $T_{i, j} : \mathbb{N}^r \to \mathbb{N}^r$ $(i \neq j$, $1 \leq i, j \leq r$, $r > 1)$ such that

$$
T_{i, j}(x_1, \ldots, x_i, \ldots, x_{j-1}, x_j, x_{j+1}, \ldots, x_r)
= x_1, \ldots, x_i, \ldots, x_{j-1}, x_i, x_{j+1}, \ldots, x_r;
$$

vol. 13, n° 1, 1979
7° the functions $C'_m : N^r \rightarrow N(m, r \geq 0)$ such that

$$C'_m(u) = m.$$  

For brevity set $\Theta = \Theta^2 = \Theta^2$; $\Delta = \Delta^1$, $I = I^1_1$.

Consider the following operators:

1° the cartesian product $\lambda f, g . (f \times g)$ such that if $f : N^r \rightarrow N^p$ and $g : N^q \rightarrow N^s$ then $f \times g : N^{r+q} \rightarrow N^{p+s}$ and $(f \times g)(u, v) = f(u), g(u)$;

2° the juxtaposition operator $\lambda f, g . (f^g)$ such that if $f : N^r \rightarrow N^p$ and $g : N^r \rightarrow N^s$ then $f^g : N^r \rightarrow N^{p+s}$ and $(f^g)(u) = f(u), g(u)$.

**Lemma 1.1:** The class $\mathcal{S}$ contains $\Theta^r, \Delta^r, \Theta^1, I^l, I^1_1, I^e, C'_m$.

**Proof:** It is immediate from the definitions. □

**Lemma 1.2:** The class $\mathcal{S}$ is closed with respect to cartesian product and juxtaposition.

**Proof:** If $f : N^r \rightarrow N^p$ and $g : N^q \rightarrow N^s$ then it holds that $f \times g = f^e . g$. If $f : N^r \rightarrow N^p$ and $g : N^r \rightarrow N^s$ then it holds that $f^g = \Delta^r . f^e . g$. □

**Notation:** Let $\mathcal{S}^{i, j}$ denote the set of primitive recursive sequence functions $f : N^i \rightarrow N^j$ for a certain $i \geq 0$ and $j > 0$, so that $\mathcal{S} = \bigcup_{i \geq 0, j > 0} \mathcal{S}^{i, j}$.

Let the set $\mathcal{P}$ of primitive recursive functions be defined as the smallest set containing $O$, the zero of zero arguments, $S$, the successor, $I^l_i$ (for $1 \leq i \leq r$), the projectors, and closed with respect to substitution and recursion. We will denote $\mathcal{P}_i$ the set of primitive recursive functions of $i$ arguments, so that $\mathcal{P} = \bigcup_{i \geq 0} \mathcal{P}_i$.

**Theorem 1.1:** $\mathcal{S} = \{ f = f_1 . . . f_s \mid f_i \in \mathcal{P}_r, r \geq 0, s > 0 \}$.

**Proof:** (a) $\mathcal{S} \subseteq \{ f = f_1 . . . f_s \mid f_i \in \mathcal{P}_r, r \geq 0, s > 0 \}$.

It is true for the basic functions because $\Sigma \subset \{ O, S, I^l_i \}$.

Assume it is true for $f : N^r \rightarrow N^s$. It holds that

$$(^c f)(u) = (f . I_1^{s+1})(u) = \begin{cases} I_{i-1}^{s+1}(u) & \text{for } i = 1; \\ I_{i-1}^{s+1}(f(I_2^{r+1}(u), \ldots, I_{s+1}^{r+1}(u))) & \text{for } 1 < i < s+1. \end{cases}$$

As by induction hypothesis $f_i = f . I_i \in \mathcal{P}_r$ and $\mathcal{P}$ is closed with respect to substitution it follows

$$(^c f)(u) = (f . I_1 \cdots I_s)(u)$$

with $(^c f)_i \in \mathcal{P}_{r+1}$. Analogously for $f^c$. 
Assume the thesis true for \( f : N^p \to N^q \) and \( g : N^q \to N^r \). It holds that
\[
(f \circ g)(u) = (f \circ g)(I_1(u)) = (f \circ (g \circ I_q))(u) = g_1(f_1(u), \ldots, f_q(u)).
\]
As by induction hypothesis \( f_i \in \mathcal{P}_p \) and \( g_i \in \mathcal{P}_q \) and \( \mathcal{P} \) is closed with respect to substitution it follows
\[
f \cdot g = (f \cdot g)_1 \ast \cdots \ast (f \cdot g)_r
\]
with \( (f \cdot g)_i \in \mathcal{P}_p \).

Assume the thesis true for \( f : N^r \to N^r \). It holds that
\[
(f^R)_1(0, u) = I_1(u),
\]
\[
(f^R)_1(S(x), u) = (f \circ (f^R)_1)(x, u), \ldots, (f^R)_r(x, u).
\]
As by induction hypothesis \( f \cdot I^*_1 \in \mathcal{P}_r \) and \( \mathcal{P} \) is closed with respect to simultaneous recursion it follows
\[
f^R = (f^R)_1 \ast \cdots \ast (f^R)_r
\]
with \( (f^R)_i \in \mathcal{P}_{r+1} \).

(b) \( \mathcal{S} \subseteq \{ f = f_1 \ast \cdots \ast f_s \mid f_i \in \mathcal{P}_r, r \geq 0, s > 0 \} \).

As \( \mathcal{S} \) is closed with respect to juxtaposition it suffices to show that \( \mathcal{P}_r \in \mathcal{S}_r \) for every \( r \geq 0 \).

The set of the basic functions of \( \mathcal{P} \) is contained in \( \mathcal{S} \) by definition and lemma 1.2.

Given \( s+1 \) functions \( f, g_1, \ldots, g_s \in \mathcal{P} \) with \( f : N^s \to N \) and \( g_i : N^r \to N \). let \( h \) be the function such that \( h(u) = f(g_1(u), \ldots, g_s(u)) \). It holds that \( h \in \mathcal{S} \) because
\[
h = (g_1 \ast \cdots \ast g_s) \circ f.
\]

Given two functions \( g, h \in \mathcal{P} \) with \( g : N^r \to N \) and \( h : N^{r+2} \to N \), let \( f \) be the function such that
\[
f(u, 0) = g(u),
\]
\[
f(u, S(x)) = h(u, x, f(u, x)).
\]
It holds that \( f \in \mathcal{S} \) because
\[
f = \Theta^r_{r+1} \cdot \Delta^r \cdot g \cdot \Delta^r \cdot (\Delta^r)^{r+1} \cdot e \cdot h \cdot e = \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \·
Let $\Sigma = \Sigma \cup \{ \Theta, \Delta \}$ and, for a subset $X$ of $\mathcal{S}$, let $\mathcal{C}(X)$ be the closure of $X$ with respect to composition and left and right cylindrification and $\mathcal{R}(X)$ the set of functions obtained from $X$ by repetition.

**Definition 2.1:**

$$\mathcal{I}_0 = \mathcal{C}(\Sigma),$$

$$\mathcal{I}_{oa+b+1} = \{ f = f_1 \circ f_2 | f_1, f_2 \in \mathcal{R}(\mathcal{I}_{oa+b}) \}$$ for every $a, b \geq 0$,

$$\mathcal{I}_{oa} = \bigcup_{i < \omega} \mathcal{I}_{oa(a-1)+i} \quad \text{for } a \geq 1.$$

Note that $\Sigma$ is the least set of functions such that $I_1 \in \mathcal{I}_0$.

**Lemma 2.1:** The following properties hold:

1° $\mathcal{I}_i \subseteq \mathcal{I}_{i+1}$ for $i \geq 0$;

2° $\mathcal{C}(\mathcal{R}(\mathcal{I}_{oa})) = \mathcal{I}_{oa(a+1)}$ for $a \geq 0$;

3° $\mathcal{C}(\mathcal{I}_{oa}) = \mathcal{I}_{oa}$ for $a \geq 0$;

4° if $f \in \mathcal{I}_{oa+b}$ then $f = f_0 \cdot f_1 \ldots \cdot f_b$ with $f_0 \in \mathcal{I}_{oa}, f_1, \ldots, f_b \in \mathcal{R}(\mathcal{I}_{oa})$;

5° if $f \in \mathcal{I}_{oa+b}$ then $f^c \in \mathcal{I}_{oa+b+c}$ for $a, b, c < \omega$.

**Proof:** It is immediate from definition 2.1. □

The property 2.1.4 affirms that the functions of the class $\mathcal{I}_{oa+b}$ are obtained by composition of a function in $\mathcal{I}_{oa}$ with $b$ functions in $\mathcal{R}(\mathcal{I}_{oa})$ which, by property 2.1.2, means $b$ functions obtained from $\mathcal{I}_0$ by a nested repetitions.

The following lemmas characterize the behavior of the above classes of functions with respect to the operators and lead eventually to the proof (see lemma 2.6) that the procedure of generating such classes ends with the class

$$\mathcal{I}_\omega^* = \bigcup_{a < \omega} \mathcal{I}_{oa+b} \quad \text{and} \quad \mathcal{I}_\omega^* = \mathcal{I}.$$ 

**Lemma 2.2:** $\mathcal{R}(\mathcal{I}_{oa}) \subseteq \mathcal{I}_{oa+1}$.

**Proof:** It is immediate by definition 2.1.

**Lemma 2.3:** If $f \in \mathcal{I}_{oa+b}$, $g \in \mathcal{I}_{oa+c}$ then $f \cdot g \in \mathcal{I}_{oa+b+c}$ for $a, b, c < \omega$.

**Proof:** It is by induction on $b$ and $c$.

For $b = c = 0$ the thesis is true by property 2.1.3.

Assume the thesis is true for $b = 0$, $c \leq n$. Let $f \in \mathcal{I}_{oa}$ and $g \in \mathcal{I}_{oa+n+1}$, then $f \cdot g \in \mathcal{I}_{oa+n+1}$ by definition of $\mathcal{I}_{oa+n+1}$.

Assume the thesis is true for $b < m$ and for every $c$. Consider the case $c = 0$. Let $f \in \mathcal{I}_{oa+m}$ and $g \in \mathcal{I}_{oa}$. Then $f \cdot g = f_1 \cdot f_2 \cdot g$ where $f_1 \in \mathcal{I}_{oa+m-1}$ and $f_2 \in \mathcal{R}(\mathcal{I}_{oa})$. 

R.A.I.R.O. Informatique théorique/Theoretical Informatics
As by induction hypothesis and lemma 2.1 \( f_2 \cdot g \in \mathcal{I}_{aa+1} \) then, again by induction hypothesis, it follows that \( f_1 \cdot f_2 \cdot g \in \mathcal{I}_{aa+m} \). Assume the thesis is true for \( b = m \) and \( c < n \). Let \( f \in \mathcal{I}_{aa+m} \) and \( g \in \mathcal{I}_{aa+n} \). Then \( f \cdot g = f \cdot g_1 \cdot g_2 \) where \( g_1 \in \mathcal{I}_{aa+n-1} \) and \( g_2 \in \mathcal{B}(\mathcal{I}_{aa}) \). As by induction hypothesis \( f \cdot g_1 \in \mathcal{I}_{aa+n+m-1} \) then, by definition of \( \mathcal{I}_{aa+m+n} \), \( f \cdot g \in \mathcal{I}_{aa+m+n} \.

**Lemma 2.4:** If \( f \in \mathcal{I}_{aa+b} \), \( g \in \mathcal{I}_{aa+c} \) then \( f \cdot g \in \mathcal{I}_{aa+b+c} \), for \( a, b, c < \omega \).

**Proof:** It follows from lemma 2.3 reminding that for \( f : \mathbb{N}^r \to \mathbb{N}^p \) and \( g : \mathbb{N}^r \to \mathbb{N}^s \), \( f \cdot g = \Delta' \cdot f^c \cdot g \) and \( \Delta' \in \mathcal{O} \).

**Lemma 2.5:** If \( f \in \mathcal{I}_{aa+b} \), \( g \in \mathcal{I}_{aa+c} \) then \( f \times g \in \mathcal{I}_{aa+b+c} \), for \( a, b, c < \omega \).

**Proof:** It follows from lemma 2.3 reminding that for \( f : \mathbb{N}^r \to \mathbb{N}^p \) and \( g : \mathbb{N}^q \to \mathbb{N}^s \), \( f \times g = f^d \cdot g \).

Let \( \mathcal{I}_{aa} = \bigcup_{a < \omega} \mathcal{I}_{aa+b} \).

**Lemma 2.6:** (a) \( \mathcal{C}(\mathcal{I}_{aa}) = \mathcal{I}_{aa} \);
(b) \( \mathcal{B}(\mathcal{I}_{aa}) = \mathcal{I}_{aa} \);
(c) \( \mathcal{I}_{aa} = \mathcal{I} \).

**Proof:** (a) Let \( f \in \mathcal{C}(\mathcal{I}_{aa}) \) then \( f = f_1 \cdot f_2 \) where \( f_1, f_2 \in \mathcal{I}_{aa} \), by definition of \( \mathcal{I}_{aa} \), there exist \( a, b, c \), such that \( f_1 \in \mathcal{I}_{aa+b} \) and \( f_2 \in \mathcal{I}_{aa+c} \). But by lemma 2.2 it follows that \( f_1 \cdot f_2 \in \mathcal{I}_{aa+b+c} \subseteq \mathcal{I}_{aa} \) then \( \mathcal{C}(\mathcal{I}_{aa}) \subseteq \mathcal{I}_{aa} \);
(b) analogously;
(c) by (a) and (b) and definition of \( \mathcal{I} \) and \( \mathcal{I}_{aa} \).

3. A HIERARCHY OF PRIMITIVE RECURSIVE SEQUENCE FUNCTIONS

In this section we show the strict containment of the class \( \mathcal{I}_i \) in the class \( \mathcal{I}_{i+1} \) (for \( i < \omega^2 \)).

The proof for \( i < \omega \) is based on the fact that a function \( f \) defined by patching together from several cases must be computed with at least a new repetition not reducible to the ones needed for the definition of the functions expressing the different cases of \( f \).

The proof for the classes \( \mathcal{I}_{aa+b} \) with \( a \geq 1, b < \omega \) is inspired by [4] and exploits properties of growth of the functions in \( \mathcal{I}_i \) with respect to a proper ordering of the sequences.

We prove first some lemmas. In lemma 3.1 the strict containment of the class \( \mathcal{I}_i \) in the class \( \mathcal{I}_{i+1} \) (\( i < \omega \)) is shown. In lemma 3.2 the strict containment of the

vol. 13, n° 1, 1979
class $\mathcal{I}_{\omega+i}$ in the class $\mathcal{I}_{\omega+i+1}$ ($i < \omega$) is shown under the normality hypothesis. In lemma 3.4 the normality property is stated for $\mathcal{I}_{\omega}$. From lemmas 3.1-3.4 theorem 3.1 follows which affirms the strict containment of the class $\mathcal{I}_{i}$ in the class $\mathcal{I}_{i+1}$ for every $i < \omega^2$.

Let $<$ denote the usual lexicographic order on sequences of natural numbers.

**Definition 3.1:** A subset $X$ of $\mathcal{I}$ containing $\Sigma \cup \{ + \}$ and closed with respect to composition and left and right cylindrification is said to be normal if it contains a strictly increasing function $h : N \to N$ with the property that for every $g : N^r \to N^s \in X$ there exists $m \in N$ such that, for every $u = x_1, \ldots, x_r$,

$$g(u) < F_0 (m, \max x_i) \quad \text{if} \quad r > 0, \quad g(u) < F_0 (m, 0) \quad \text{if} \quad r = 0.$$ 

with $F_0 (x, y) = (h^* \cdot (h \cdot S) \cdot h^R) (x, y)$.

**Lemma 3.1:** For every $i < \omega$ there exists a function $f \in \mathcal{I}_{i+1} - \mathcal{I}_i$.

**Proof:** Let $r_m (x)$ the residue of the division of $x$ by a constant $m$. Consider the following functions:

$$f_1 (x) = mx,$$

$$f_2 (x) = \begin{cases} x + m_1 & \text{if} \quad r_m (x) = 0, \\ C^1_m (x) & \text{otherwise}, \end{cases}$$

$$f_i+1 (x) = \begin{cases} f_i (x) & \text{if} \quad r_m (x) = 0 \\ C^1_m (x) & \text{otherwise} \end{cases} \quad \text{for} \quad i \geq 2,$$

with $m > 1$ and $r_m (m_j) \neq 0$ for every $i \neq j$.

It is easy to see that if we want to define a function by patching together from several cases we cannot dispense with introducing repetitions (see also how in the recursive function theory a function defined by cases is reduced to the composition of functions obtained by recursion).

In our case, assuming that $f_i \in \mathcal{I}_i$ but not $f_i \in \mathcal{I}_{i-1}$, we obtain $f_{i+1}$ from $f_i$ by composition with a repetition on $C^1_m$ and therefore $f_{i+1} \notin \mathcal{I}_i$.

Now

$$f_1 = ^0 O \cdot (S^m)^R.$$ 

As it is easy to check that the functions in $\mathcal{I}_0$ are of the type

$$f (x_1, \ldots, x_r) = (x_i + m_1, \ldots, x_i + m_s)$$

with $x_i \in \{ x_1, \ldots, x_r \} \cup \{ 0 \}, m_i \geq 0, s > 0$, then $f_1 \notin \mathcal{I}_0$ cannot be expressed in $\mathcal{I}_0$. Furthermore

$$f_2 = \Delta \cdot r_m \cdot ^c S^m \cdot (C^1_m)^R$$

where $r_m = C^0_1 \cdot ^2 C^0_2 \cdot \ldots \cdot e^{r-1} C^0_{m-1} \cdot e^t C^0_0 \cdot (\Theta^m)^R \cdot I^m_1$. As $r_m \in \mathcal{I}_1 - \mathcal{I}_0$ and

R.A.I.R.O. Informatique théorique/Theoretical Informatics
because we cannot spare the repetition, it follows that \( f_2 \in \mathcal{F}_2 - \mathcal{F}_1 \). Finally
\[
f_{i+1} = \Delta \cdot \text{re}_{m_1, \ldots, m_i} \cdot c^{m_i} \cdot (I_{i}^1 \cdot C_{m_i}^1 \cdot C_0^0 \cdot \ldots \cdot C_0^{0^c_{i+1}})^R.
\]
\[
(I_{i}^{-1}^1 \cdot C_{m_i-1}^1 \cdot C_0^0 \cdot \ldots \cdot C_0^{0^c_{i+1}})^R \cdot \ldots \cdot (C_m^1)^R
\]
where
\[
\text{re}_{m_1, \ldots, m_i} = (C_0 \cdot c^0 \cdot c^0 \cdot \ldots \cdot c^{m_i-2} \cdot C_{m_i-1} \cdot C_0) \cdot \ldots \cdot (C_0 \cdot c^0 \cdot c^0 \cdot \ldots \cdot c^{m_i-2} \cdot C_{m_i-1} \cdot C_0)
\]
is the function such that \( \text{re}_{m_1, \ldots, m_i}(x) = (\text{re}_{m_i}(x), \ldots, \text{re}_{m_1}(x)) \). As \( f_{i+1} \) can be obtained by compositions of functions in \( \mathcal{F}_0 \) with \( \text{re}_{m_1, \ldots, m_i} \) (which can be always computed by a single repetition) and \( i \) functions obtained by repetition of functions in \( \mathcal{F}_0 \), it follows \( f_{i+1} \in \mathcal{F}_{i+1} \) and \( f_{i+1} \notin \mathcal{F}_i \).

**Lemma 3.2:** Suppose that \( \mathcal{F}_\omega \) is normal. Consider the functions \( F_i : N^2 \to N \), \( G_i : N \to N \), \( H_i : N \to N \) defined as follows:
\[
F = h^c \cdot (h \cdot S) \cdot h^R; \quad F_{i+1} = F_i \cdot \Delta \cdot h^R; \quad G_i = (C_m^0)^c \cdot F_i; \quad H_i = \Delta \cdot F_i
\]

Then it holds that:
1° \( F_i \in \mathcal{F}_{\omega + i + 1} \), \( G_i \in \mathcal{F}_{\omega + i} \), \( H_i \in \mathcal{F}_{\omega + i + 1} \);
2° for every \( i < \omega \), for every \( g : N^r \to N^s \in \mathcal{F}_{\omega + i} \) there exists \( m \in N \) such that \( g(u) < F_i(m, \text{max} \ x_i) \);
3° for every \( i < \omega \), for every \( g : N \to N \in \mathcal{F}_{\omega + i} \) there exists \( m \in N \) such that \( g(x) < H_i(x) \) for \( x > m \).

**Proof:** 1° it follows immediately from the definition;
2° the thesis is true for \( i = 0 \) by hypothesis.

Suppose the thesis is true for \( i < j \). Consider a function \( g : N^r \to N^s \in \mathcal{F}_{\omega + j} \). By definition \( g = g_1 \cdot g_2^k \), \( g_1 \in \mathcal{F}_{\omega + j - 1} \) and \( g_2 \in \mathcal{F}_\omega \). By induction hypothesis there exist \( m_2 \), \( m_i' \) such that
\[
(g_1 \cdot I_{i+1}^j) \cdot (x_1, \ldots, x_r) < F_{j-1} \cdot (m_i', \text{max} \ x_i)
\]
and \( g_2 \cdot (y_1, \ldots, y_s) < F_0 \cdot (m_2, \text{max} \ y_i) \).

For every \( y > 0 \) it results
\[
g_2^k \cdot (y, y_1, \ldots, y_s) < ((C_m^0)^c \cdot F_0)^R(y, \text{max} \ y_i)
\]

vol. 13, n° 1, 1979
and then for every \((x_1, \ldots, x_r) \in \mathbb{N}^r:\]

\[g(x_1, \ldots, x_r) < ((C_{m_1}^0\cdot F_0)^r (F_{j-1}(m_1, \max x_i), F_{j-1}(m_1, \max x_i)) = ((C_{m_1}^0\cdot F_{j-1} \cdot \Delta \cdot (C_{m_2}^0\cdot F_0)^r) (\max x_i)
\]

Let \(f_{m_1, m_2} = (C_{m_1}^0\cdot F_{j-1} \cdot \Delta \cdot (C_{m_2}^0\cdot F_0)^r\). It holds that

\[f_{m_1, m_2}(x) = h^{(m_2)}(S(\ldots(h^{(m_2)}(S(F_{j-1}(m_1, x))\ldots)))
\]

\[\leq (h^{(m_2)+1})^{F_{j-1}(m_1, x)} (F_{j-1}(m_1, x))
\]

\[= h^R((C_{m_1}^0\cdot F_{j-1} \cdot \Delta \cdot (S^{(m_2)+1})^R)(x), F_{j-1}(m_1, x)).
\]

Now \((C_{m_1}^0\cdot F_{j-1} \cdot \Delta \cdot (S^{(m_2)+1})^R) \in \mathcal{I}_{\omega + j - 1}\) and then by induction hypothesis there exists \(m > m_1, m_2\) such that \((C_{m_1}^0\cdot F_{j-1} \cdot \Delta \cdot (S^{(m_2)+1})^R)(x) < F_{j-1}(m, x)\) for every \(x\). So finally we obtain \(f_{m_1, m_2}(x) < h^R(F_{j-1}(m, x), F_{j-1}(m, x))\) and then

\[g(x_1, \ldots, x_r) < F_j(m, \max x_i);
\]

3° by 2 for every \(f : \mathbb{N} \to \mathbb{N} \in \mathcal{I}_{\omega + i}\) there exists \(m \in \mathbb{N}\) such that for every \(x, f(x) < F_i(m, x)\). As \(F_i(m, x) < H_i(x)\) for every \(x > m\) then \(f(x) < H_i(x)\) for \(x > m\).

**Lemma 3.3:** If \(\mathcal{I}_\omega\) is normal with the function \(h\) then \(\mathcal{I}_{\omega + 1}\) is normal with the function \(h^* = H_0\).

**Proof:** Consider \(F_0^* : \mathbb{N}^2 \to \mathbb{N}, G_0^* : \mathbb{N} \to \mathbb{N}\) defined as follows

\[F_0^* = h^* \cdot c(h^* \cdot S) \cdot h^* R\quad \text{and} \quad G_0^* = (C_{m_1}^0\cdot F_0^*).
\]

By definition \(h^* \in \mathcal{I}_{\omega + 1} \subseteq \mathcal{I}_{\omega + 2}, F_0^* \in \mathcal{I}_{\omega + 2}, G_0^* \in \mathcal{I}_{\omega + 2}\). Suppose \(g : \mathbb{N}^r \to \mathbb{N}\) and \(g \in \mathcal{I}_{\omega + 2}\). In order to prove that there exists \(m \in \mathbb{N}\) such that, for every \(u = x_1, \ldots, x_r, g(u) < F_0^*(m, \max x_i)\) we must show, by lemma 3.2.2, that, for every \(n, i \in \mathbb{N}\), there exists \(m \in \mathbb{N}\) such that, for every \(x, F_i(n, x) < F_0^*(m, x)\). By the definitions of \(F_i\) and \(F_0^*\) it results

\[F_i(n, x) = ((\Delta \cdot h^R)^i)(i, F_0(n, x))
\]

and

\[F_0^*(m, x) = ((\Delta \cdot h^R \cdot c(h \cdot c(S \cdot h^R))^R)(F_0(m, m), F_0(x, x) + 1).
\]

Let \(g = \Delta \cdot h^R \cdot c(h \cdot c(S \cdot h^R)). It is easy to prove by induction on \(n\) that, for every \(x, F_0(n, x) < g^R(n, F_0(x, x) + 1) \leq F_0^*(m, x)\) where \(m\) is the least integer such that \(n \leq F_0(m, m)\). If we suppose that \(F_{i-1}(n, x) < g^R(n + i - 1, F_0(x, x) + 1)\), for every \(n \) and \(x\), we obtain that \(F_i(n, x) < g^R(n + i, F_0(x, x) + 1)\). Then for every \(i, n \in \mathbb{N}\) it results \(F_i(n, x) < F_0^*(m, x)\), where \(m\) is the least integer such that \(F_0(m, m) \geq n + i\).

**Lemma 3.4:** \(\mathcal{I}_\omega\) is normal with \(h = \Delta \cdot S^R\).
Proof: By induction on $J_m$. We have shown that if $f \in \mathcal{S}_0$ then

$$f(x_1, \ldots, x_r) = (x_i + m_1, \ldots, x_i + m_s)$$

with $x_i \in \{x_1, \ldots, x_r\} \cup \{0\}$, $m_j \in \mathbb{N}$. Let $m = \max x_i$. It is immediate to see that

$$f(x_1, \ldots, x_r) \leq S^R(m, \max x_i) < 2^{2m}(2 \max x_i + 1) = F_0(m, \max x_i).$$

Suppose now $r = s$ and consider $f^R$. It holds that

$$f^R(x, x_1, \ldots, x_r) \leq ((\Delta^c \cdot S^R)^R \cdot I_2^2)(x, m, \max x_i) < F_0(m, \max (x, x_i)).$$

Suppose the thesis is true for $j < i$. Consider $f \in \mathcal{S}_i$. If $f \in \mathcal{S}_i$ then $f = f_1 \cdot f_2$ with $f_1 \in \mathcal{S}_{i-1}, f_2 \in \mathcal{S}_0$. Then there exist $m'_1, m'_2$ such that

$$(f_1 \cdot I^1_{j+1})(x_1, \ldots, x_r) < F_0(m'_1, \max x_i) \leq F_0(m_1, \max x_i)$$

where $m_1 = \max m'_1$ and $f'_2(y, y_1, \ldots, y_r) < F_0(m_2, \max (y, y_i))$ and finally

$$f(x_1, \ldots, x_r) < F_0(m_2, F_0(m_1, \max x_i)) < F_0(2m_2 + m_1 + 1, \max x_i).$$

The strict containment of the class $\mathcal{S}_{2\omega + i - 1}$ in $\mathcal{S}_{2\omega + i}$ for every $i$ can be proved by defining a new sequence of functions $F^*_i$ starting with $F^*_0$ as it has been done in lemma 3.2 starting with $F_0$. Now $\mathcal{S}_{3\omega}$ can be proved to be normal with $h^{**} = H^* = \Delta \cdot F^*_0$. By repeating the same reasoning up to $\mathcal{S}_\omega$, the following theorem can be stated.

**Theorem 3.1:** $\mathcal{S}_i \subseteq \mathcal{S}_{i+1}$ for $i < \omega^2$.

### 4. Complexity Classes of Loop Programs

In [10] partial recursive sequence functions have been proposed to give a semantics of a simple recursive language (i.e. the language SL introduced by the authors). Analogously primitive recursive sequence functions can be used to give a meaning to Meyer-Ritchie LOOP programs (see [13]) in a version which allows more than one output variable. In this manner we obtain a relationship between the structural complexity of LOOP programs and the computational complexity of primitive recursive sequence functions.

**Definition 4.1:** A LOOP program has the following form:

$$\text{IN } s; I_1; \ldots; I_k; \text{ OUT } t \quad (k \geq 0)$$

where $s$ is a list (possibly empty) of names for variables (without repetitions) and $I_i$ is an instruction of one of the following types:

(a) $X_i \gets 0$ where $X_i$ is an input variable or a variable introduced before or a new variable;

vol. 13, n° 1, 1979
(b) \( X_t \leftarrow X_j \) where \( X_i \) and \( X_j (i \neq j) \) are input variables or variables introduced before;

c) \( X_t \leftarrow X_t + 1 \) where \( X_i \) is an input variable or a variable introduced before;

d) LOOP \( X_i ; I_1 ; \ldots ; I_j ; \) END where \( X_i \) is a input variable or a variable introduced before and \( I_i \) are instructions of types \( a, b, c, d \);

and where \( t \) is a (non empty) list of names either contained in the input list or introduced in \( I_i (1 \leq i \leq k) \).

**Definition 4.2:** A function \( f: N^r \rightarrow N^s \in \mathcal{S} \) is computed by a LOOP program \( P \) with input list \( s \) and output list \( t \) if before the execution of \( P \) the input list contains \( x_1, \ldots, x_r \in N^r \) (and the other registers are empty) and after the execution of \( P \) the output list contains the sequence \( f(x_1, \ldots, x_r) \in N^s \). The meaning of a LOOP program \( P \) is the function computed by \( P \).

Let \( \mathcal{F} \) be the set of functions computed by the LOOP programs.

**Theorem 4.1:** \( \mathcal{F} = \mathcal{S} \).

**Proof:** (a) \( \mathcal{F} \subseteq \mathcal{S} \).

Let \( s = (X_1, \ldots, X_r) \) and \( t = (X_{r+1}, \ldots, X_r) \), where \( i_j \in \{1, \ldots, r\} \cup \{r+1, \ldots, r+p\} \), and \( X_{r+1}, \ldots, X_{r+p} \) are the new variables introduced by the instructions of the programs.

Case 1: The program \( P : \text{IN} \ s; \text{OUT} \ t \) computes the function \( I_{i_1} \cdots I_{i_q} \) (in this case is always \( q \leq r, p = 0 \)).

Case 2: The program \( P : \text{IN} \ s; X_t \leftarrow 0; \text{OUT} \ t \) computes the function

\[
(c ^{-1} (C_0) c ^{-1} )(I_{i_1} \cdots I_{i_q} ), \quad \text{if} \quad 1 \leq i \leq r,
\]

and the function

\[
c \quad O. (I_{i_1} ^{+1} \cdots I_{i_q} ^{+1} ),
\]

otherwise.

The program \( P : \text{IN} \ s; X_t \leftarrow X_j; \text{OUT} \ t \) computes the function

\[
T_{i,j} ^{1} (I_{i_1} \cdots I_{i_q} ).
\]

The program \( P : \text{IN} \ s; X_t \leftarrow X_t + 1; \text{OUT} \ t \) computes the function

\[
c ^{-1} S ^{-1} (I_{i_1} ^{+1} \cdots I_{i_q} ^{+1} ).
\]

Case 3: Consider the program \( P : \text{IN} \ s; I_1 ; \ldots ; I_k; \text{OUT} \ t \) where \( I_i \) are instructions of the types \( a, b, c, d \). Take the programs \( P_i : \text{IN} \ s_i ; I_i; \text{OUT} \ t_i \) (1 \( \leq i \leq k \)) with \( s_i = s, s_i = (s, s_{i-1} ) \), for \( 1 < i \leq k \), where \( s_{i-1} \) is the list of new variables introduced by \( I_1, \ldots, I_{i-1} \), and \( t_1 = (s, s_1') \), \( t_i = s_{i+1}, \) for \( 1 < i < k \), and \( t_k = t \). Assume that the functions \( f_i \) computed by the programs \( P_i \) belong to \( \mathcal{S} \). Then the function \( f \) computed by the program \( P \) can be written as composition of functions in \( \mathcal{S} \) and therefore \( f \in \mathcal{S} \).

R.A.I.R.O. Informatique théorique/Theoretical Informatics
Case 4: Consider the program $P : \text{IN} \ s; \text{LOOP} \ X_t; I_x; \ldots; I_k; \text{END}; \text{OUT} \ t$ where the instructions $I_i$ are of the types $a, b, c, d$. The function $f$ computed by the program $P$ is

$$f' \circ \ldots \circ f' \circ \Delta^{e^{r-1}} \cdot \Theta^{i_p+1} \cdot f^{R} \cdot (I_i^{p} \circ \ldots \circ I_k^{p}),$$

where $f'$ is the function computed by $P' : \text{IN} (s, s'); \text{loops} \ I_1; \ldots; I_k; \text{OUT} (s, s')$ being the list of the variables introduced by the statements $I_i$. Assume that $f' \in \mathcal{F}$, then $f \in \mathcal{S}$.

(b) $\mathcal{F} \supseteq \mathcal{S}$.

Programs computing the functions of $\Sigma$ are constructed easily. If $f_1, f_2 \in \mathcal{S}$ and $P_1, P_2$ are the programs computing $f_1, f_2$ respectively, then the program computing the function $f_1 \cdot f_2$ is obtained from $P_1$ and $P_2$ by the insertion of instructions which take the contents of the output registers of $P_1$ into the input registers of $P_2$. If $f \in \mathcal{S}$ and $P$ is the program computing $f$, then the program computing $f^R$ is obtained by including the instructions of $P$ in a couple LOOP-END.

**Theorem 4.2:**

$$\mathcal{M}_{\omega a + b} = \mathcal{I}_{\omega a + b} \text{ for every } a, b < \omega.$$  

**Proof:** It can be given easily by induction on $a$ and $b$.

In fact if $f : N^r \to N^s \in \mathcal{I}_{\omega c + n}$ then $f = f_1 \cdot f_2$ with $f_1 \in \mathcal{I}_{\omega c + n - 1}$ and $f_2 \in \mathcal{I}_{\omega c + 1}$ and by induction hypothesis there exist $P_1 \in \mathcal{M}_{\omega c + n - 1}$ and $P_2 \in \mathcal{M}_{\omega c + 1}$ computing $f_1$ and $f_2$ respectively. Then the program $P$ computing $f$ is a program in $\mathcal{M}_{\omega c + n}$. Therefore $\mathcal{I}_{\omega c + n} \subseteq \mathcal{M}_{\omega c + n}$.

Conversely if $f \in \mathcal{M}_{\omega c + n}$ and $P$ is the program computing $f$ one can always consider $P$ as obtained from the composition of two programs $P_1 \in \mathcal{M}_{\omega c + n - 1}$ and $P_2 \in \mathcal{M}_{\omega c + 1}$. The functions $f_1, f_2$ computed by these programs are, by induction hypothesis, in $\mathcal{I}_{\omega c + n - 1}$ and in $\mathcal{I}_{\omega c + 1}$ respectively and then $f = f_1 \cdot f_2 \in \mathcal{I}_{\omega c + n}$. Therefore $\mathcal{M}_{\omega c + n} \subseteq \mathcal{I}_{\omega c + n}$. 

**Definition 4.3:** Let $M_0$ be the class of LOOP programs obtained by using only instructions of the types $a, b, c$. Let $M_{\omega a + b}$ be the class of LOOP programs $P$ such that in $P$ there are $b$ successive instructions of the form LOOP $X_t; I; \text{END}$ where the instruction $I$ contains at most $a$ nested instructions of the typed. Let $M_{\omega a} = \bigcup_{i < \omega} M_{\omega (a - 1) + 1}$ for $a \geq 1$.

We consider the following classes of primitive recursive sequence functions:

$$M_i = \{ f | f \in \mathcal{S} \text{ and there exists } P \in M_i \text{ such that } f \text{ computable by } P \} \text{ for } i < \omega^2.$$
**Definition 4.4:** Let $P$ a program $\text{IN} \ s; I; \text{OUT} \ s''$ with $I = I_1; \ldots; I_n (n \geq 0)$. Let $s'$ be the list of new variables introduced by $I$ and $r, q$ be the lengths of the variables lists $s$ and $(s, s')$ respectively. Then $t_P$, the computing time function of $P$, is

$$
t = t^{O \cdot \theta_{q+1}} \cdot \ldots \cdot t^{O \cdot \theta_{q+1}} \cdot t \cdot I_{q+1}^q$$

where $t$ is the stepcounter function of $P$ defined as follows:

(a) $t = I_{q+1}^q$ if $n = 0$;

(b) $t = \frac{c}{e} \cdot s_i$ if $I : X_i \rightarrow X_i + 1$;

t = \frac{c}{e} \cdot (C_0 \cdot s_i)$ if $I : X_i \rightarrow 0$;

t = T_{i, j}^q \cdot \frac{c}{e} S$ if $I : X_i \rightarrow X_j$;

t = t_1 \cdot t_2$ if $I : I_1; I_2$;

t = \frac{c}{e} \cdot \frac{\Delta}{\theta_{q+1}} \cdot (t_{1, s} \cdot (S, S))^R \cdot \frac{c}{e} (S, S)$ if $I : \text{LOOP} X_i; I'; \text{END}$.

**Lemma 4.1:** If $P \in M_{aa+b}$ then $t_P \in J_{aa+b}$ for $a, b \geq 0$.

**Proof:** The program $P'$ computing $t_P$ is obtained by $P$ as follows: the input list is the same of $P$; the first instruction is of the type $T \leftarrow 0$, where $T$ is a register not used in $P$; the successive instructions are the ones of $P$ followed by an instruction of the type $T \leftarrow T + 1$, for each instruction of the type $a, b, c$ of $P$ and followed by two instructions of the type $T \leftarrow T + 1$ for each loop instruction. Moreover for each loop instruction there are two instructions of the type $T \leftarrow T + 1$ between the couple $\text{LOOP-END}$; finally $T$ is the only output register.

Now for the classes $M_{aa+b}$ we can prove a result analogous to that proved by Meyer and Ritchie for the classes $L_i$ (see [18]).

Let $F_{n}^{**m}$ be the functions such that for every $g \in J_{aa+n}$ there exists $p \in N$ such that, for every $u = x_1, \ldots, x_r$, $g(u) < F_{n}^{**m}(p, \max x_i)$ (see lemmas 3.2-3.4).

**Lemma 4.2:** Let $P \in M_{aa+b}$. Suppose that $t_P(u) < F_{n}^{**m}(p, \max x_i)$ for $p \in N$ and $1 \leq m < a, b > 0$. Then there exists a program $P' \in M_{aa+n}$ such that $P$ and $P'$ compute the same function, with $n' = n$ if $m > 1, n' = n + 1$ otherwise.

**Proof:** It is analogous to the proof given by Meyer and Ritchie. The program $P'$ can be obtained from the program that compute $F_{n}^{**m}(p, \max x_i)$ and from a program in $M_{aa+1}$ that simulates the instruction sequence of $P$.

By lemmas 4.1 and 4.2 the following theorem can be stated.

**Theorem 4.3:** Given a function $f \in J$ and a program $P$ which computes $f$, $f \in J_{aa+b}$ iff $t_P(u) < F_{b}^{**m}(p, \max x_i)$ for a proper $p$, with $b' = b + 1$ for $a = 1, b > 0$ and with $b' = b$ for $a > 1, b \geq 0$.
5. COMPARISON WITH OTHER HIERARCHIES OF PRIMITIVE RECURSIVE FUNCTIONS

As primitive recursive functions coincide with primitive recursive sequence functions when the output sequence is of length one (see theorem 1.1), it is interesting to compare the above hierarchy in this particular case with known hierarchies of primitive recursive functions.

We recall briefly the definitions of Axt, Cleave, Grzegorczyk, Meyer-Ritchie hierarchies.

Let \( B = \{ S = \lambda x. (x + 1), O = (0), I'_0 = \lambda x_1, \ldots, x_r.(x_i) \} \).

The initial class \( R_0 \) of the Axt hierarchy is defined as the closure of \( B \) with respect to substitution; the class \( R_i \), for \( i < \omega \), is defined as the smallest set of functions containing \( R_{i-1} \) and the functions obtained from those in \( R_{i-1} \) by primitive recursion (see [2]).

The initial class \( E_0 \) of Cleave hierarchy can be defined as the closure of the set \( \Gamma = \{ f_1 = \lambda x, y. (x + y), f_2 = \lambda x, y. (xy), \delta = \lambda x, y, (\text{if } x \neq y \text{ then } 0 \text{ else } 1) \} \) with respect to substitution; the class \( E_{\omega_1 + b} \), for \( a, b < \omega \), can be defined as the set of functions \( f : N^r \to N \) such that \( f (u) = R_0 (u, R_1 (u, \ldots, R_b (u, 1) \ldots )) \), where \( R_i \) is obtained by \( a \) nested simultaneous recursions (see [4]).

The class \( \mathcal{E}_i \), for \( i < \omega \), is defined as the smallest set of functions containing \( B \) and the function \( f_i \), where \( f_0 (x, y) = x + 1; f_1 (x, y) = x + y; f_2 (x, y) = xy; f_n (x, 0) = 1 \) and \( f_n (x, y + 1) = f_{n-1} (x, f_n (x, y)) \), for \( n \geq 3 \), and closed with respect to substitution and limited recursion (see [11, 15]).

The class \( \mathcal{L}_i \) of the Meyer-Ritchie hierarchy is defined as the class of functions computed by LOOP programs (with only one output register) in \( L_i \), i.e. by programs with at most \( i \) nested LOOP-END instructions (see [13]).

Let \( \mathcal{J}'_i \), for \( i < \omega^2 \), be the subclass \( \mathcal{J}_i \) containing only functions with output sequence of length one, i.e., \( \mathcal{J}_i = \{ f. I' : f : N^r \to N^s \in \mathcal{J}_i, r \geq 0, s > 0 \} \).

By the theorem 4.2 the following lemma holds.

**Lemma 5.1:** \( \mathcal{L}_i = \mathcal{J}'_i \) for \( i < \omega \).

The following lemmas express the relationship between the classes \( \mathcal{J}'_i \) and the classes \( \mathcal{E}_i \) and \( E_i \).

**Lemma 5.2:** \( \mathcal{J}'_i = \mathcal{E}_{i+1} \) for \( i > 1 \).

**Proof:** From [13] we have \( \mathcal{L}_i = \mathcal{E}_{i+1} \) for \( i > 1 \). From lemma 5.1 the thesis follows immediately. \( \square \)

**Lemma 5.3:** \( \mathcal{J}'_i + j = E_{\omega (i-1) + j} \) for \( i > 1, j \geq 0 \).

vol. 13, n° 1, 1979
Proof: From [4] we have $E_{\omega i} = S_{i+2}$ for $i > 0$. From lemma 5.2 the thesis follows immediately. □

Using lemmas 5.1, 5.2, 5.3 and known results (see [2, 13, 14, 16, 17]) we can draw the following diagram to show the relationships among the hierarchies we have considered. In the diagram $A \rightarrow B$ means that the class $A$ is strictly enclosed in the class $B$; $A \dashv \vdash B$ means that the two classes are not comparable.

Note that the classes $E_i(0 \leq i < \omega)$ are not comparable with $E_i(i < 3)$. The class $\mathcal{J}_0 = \mathcal{L}_0 = R_0 \leq E_0$, but $\mathcal{L}_1$, $R_1$ are not comparable with $E_i(i < \omega)$, because the predecessor function is in $\mathcal{L}_1 \cap R_1$ and not in $E_0$, and the product function is in $E$ but not in $\mathcal{L}_1$ or in $R_1$. The classes $\mathcal{J}_i$ are contained in the classes $E_i$, for $0 \leq i < \omega$.

The class of functions computed by Beck class $\mathcal{L}_j$ of LOOP programs (see [3]), defined as the class of programs with $j$ successive different subprograms containing at most $i+1$ nested LOOP-END instructions, is seen easily to coincide with the class of functions $\mathcal{J}_{\omega i+j}$, for $i, j \geq 0$.

If we enlarge the base of the hierarchy with the functions sum, product and “if $x \neq y$ then 0 else 1”, i.e. we take Cleave’s base, we obtain Cleave’s hierarchy again.

Let $\Sigma' = \Sigma \cup \Gamma$. Let us take $\mathcal{F}_0 = \mathcal{C}(\Sigma')$ and define new classes $\mathcal{F}_i$ with a construction analogous to the one used for $\mathcal{F}_i$. The following lemma can be proved easily by induction.
LEMMA 5.4: $E_i = \overline{\mathcal{J}_i}$ for $i < \omega^2$.

In [4] Cleave considers primitive recursive functions $f : N^r \to N^s$ defined as tuple of functions $f_i : N^r \to N$. The juxtaposition operator is used implicitly and the following assertion is proved: "if $f_j : N^r \to N \in E_{\omega a+b}$, then $f = (f_1, \ldots, f_s) : N^r \to N^s \in E_{\omega a+b}^s$ for $a, b < \omega$". Now in lemma 2.4 we proved that if $f \in \mathcal{J}_{\omega a+b}$ then $(f_1 \ldots f_s) \in \mathcal{J}_{\omega a+s}$ but we can prove also the following lemma.

LEMMA 5.5: If $f_i \in \mathcal{J}_{\omega a+b}$, for $a \geq 2$, $b \geq 0$, $1 \leq i \leq s$, then $f_1 \ldots f_s \in \mathcal{J}_{\omega a+b}$.

Proof: Consider the LOOP programs $P_i \in M_{\omega a+b}$ computing the functions $f_i \in \mathcal{J}_{\omega a+b}$. It is easy to construct a program $P$ computing $f = f_1 \ldots f_s$ and consisting of a part in charge of making $s$ copies of the input followed by the $s$ subprograms $P_i$ (possibly with some names of variables changed). As each $P_i$ consists of the $b$ successive loops each containing $a$ nested loops, the program $P$ will consist of $s b$ successive loops each containing $a$ nested loops, i.e. $P \in M_{\omega a+sb}$. For $a \geq 2$ and $b \geq 0$ one can construct a program $P'$ equivalent to $P$ in which $s$ loops are substituted by one only loop which runs on the maximum of the values on which the single loops were running. □

Then from lemmas 5.3 and 5.5 we obtain the following lemma.

LEMMA 5.6:

1° \[ \mathcal{J}_{\omega (a+1)+b} = \bigcup_{s, r} E_{\omega a+b}^s \quad \text{for} \quad a \geq 1, b \geq 0. \]

2° \[ \overline{\mathcal{J}_{\omega a+b}} = \bigcup_{s, r} E_{\omega a+b}^s \quad \text{for} \quad a, b \geq 0. \]

The above results show that the classification of primitive recursive sequence functions is not a trivial extension of classification of primitive recursive functions.

Cleave's method for the uniform generation of classes by simplified $s$ multaneous recursion and substitution needs a hierarchy base containing sum and product. The sequence functions formalism allows the uniform generation of the hierarchy by repetition and composition starting from a very simple base. It results that the sum is in $\mathcal{J}_1$ and the product is in $\mathcal{J}_{\omega+1}$, which seems to be more reasonable than having both operations in the same class, see e.g. [3].

6. REMARKS ON SOME DECIDABILITY RESULTS

Some decidability results proved for the classes $L_i$ and $L^j_i$ (see [3, 13, 17]) are generalized immediately to the classes $\mathcal{J}_i$. 

vol. 13, n° 1, 1979
The following assertions hold:

- the equivalence problem is recursively unsolvable in $\mathcal{F}_{a+b}$, for $a \geq 1$, $b \geq 2$;
- the equivalence problem for functions of is recursively solvable $\mathcal{F}_{a}$ in $\mathcal{F}_{2a}$;
- the problem of determining the least $a, b$ such that $f$ belongs to $\mathcal{F}_{a+b}$ is recursively unsolvable for $a \geq 2$, $b \geq 1$;
- there is an algorithm which for a given function $f \in \mathcal{F}_{i}(i < \omega)$ determines whether the expression defining $f$ contains a repetition on a constant, so that in such a case $f \in \mathcal{F}_{i-1}$.

The method for generating a hierarchy of primitive recursive sequence functions expounded above can be used to generate other hierarchies of primitive recursive sequence functions starting from different bases. In particular we can define classes $\mathcal{F}_{i}^{P}$, $\mathcal{F}_{i}^{t}$ having as base $\Sigma \cup \{ P \}$ and $\Sigma \cup \{ t \}$, with $P = \lambda x . (x - 1)$ and $t = \lambda x, y, z . (\text{if } z = 0 \text{ then } x \text{ else } y)$, respectively.

The results on decidability of the equivalence problem proved by Beck (see [3]) and by Huwig and Claus (see [12]) can be extended easily to the classes $\mathcal{F}_{i}^{P}$ and $\mathcal{F}_{i}^{t}$.

REFERENCES