RAIRO. INFORMATIQUE THÉORIQUE

E. FACHINI
A. MAGGIOLO-SCHETTINI

A hierarchy of primitive recursive sequence functions

RAIRO. Informatique théorique, tome 13, nº 1 (1979), p. 49-67
<http://www.numdam.org/item?id=ITA_1979__13_1_49_0>

R.A.I.R.O. Informatique théorique/Theoretical Informatics
(vol. 13, n° 1, 1979, p. 49 à 67)

A HIERARCHY OF PRIMITIVE RECURSIVE SEQUENCE FUNCTIONS (*)

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

Communicated by G. AUSIELLO

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

INTRODUCTION

Partial recursive sequence functions, i. e. partial functions of type $f: N^r \to N^s$, 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 $N^r \to N^s (r \geqq 0, s > 0)$, 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 ω^2 of primitive recursive sequence functions in a very simple manner: every class $\mathscr{I}_{\omega a+b}$ contains the functions obtained from the basic one by a nested repetitions and b successive compositions. (The idea of constructing an ω^2 hierarchy was suggested by Cleave in [4] where an ω^2 hierarchy of functions $N^r \to N$ 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.

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 $M_{\omega a+b}$ 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 $\mathscr{I}_{\omega a+b}$ 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 ω^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 $\mathscr{I}'_{\omega a+b} \subsetneqq \mathscr{I}_{\omega a+b}$ 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 $\mathscr{I}_{\omega a+b}$.

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 $\mathscr{S}$ of primitive recursive sequence functions $f : N^r \to N^s$, with $r \geqq 0$, $s>0$ is defined as the least set of functions containing Σ and closed with respect to the following operators :

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

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

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

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

Consider the following functions:

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

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

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

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

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

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

4° the functions $I_i^r : N^r \to N$ $(1 \leqq i \leqq r)$ such that

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

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

$$I^r(u)=u;$$

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

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

7° the functions $C^r_m : N^r \to N\,(m,\ r \geqq 0)$ such that

$$C^r_m(u) = m.$$

For brevity set $\Theta = \Theta^2_2 = \Theta^2$; $\Delta = \Delta^1$, $I = I^1_1$.

Consider the following operators:

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

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

LEMMA 1.1: *The class $\mathscr{S}$ contains Θ^r, Δ^r, Θ^r_i, I^r, I^r_i, $T^r_{i,j}$, C^r_m.*

Proof: It is immediate from the definitions. □

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

Proof: Il $f : N^r \to N^p$ and $g : N^q \to N^s$ then it holds that $f \times g = f^{c^q} . {}^{c^p}g$ If $f : N^r \to N^p$ and $g : N^r \to N^s$ then it holds that $f \hat{\ } g = \Delta^r . f^{c^r} . {}^{c^p}g$. □

NOTATION: Let $\mathscr{S}^{i,j}$ denote the set of primitive recursive sequence functions $f : N^i \to N^j$ for a certain $i \geqq 0$ and $j > 0$, so that $\mathscr{S} = \bigcup_{\substack{i \geqq 0 \\ j > 0}} \mathscr{S}^{i,j}$.

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

THEOREM 1.1: $\mathscr{S} = \{ f = f_1 \hat{\ } \ldots \hat{\ } f_s \mid f_i \in \mathscr{P}_r,\ r \geqq 0,\ s > 0 \}$.

Proof: (*a*) $\mathscr{S} \subseteqq \{ f = f_1 \hat{\ } \ldots \hat{\ } f_s \mid f_i \in \mathscr{P}_r,\ r \geqq 0,\ s > 0 \}$.

It is true for the basic functions because $\Sigma \subset \{ 0, S, I^r_i \}$.

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

$$({}^c f)_i(u) = ({}^c f . I^{s+1}_i)(u) = \begin{cases} I^{s+1}_i(u) & \text{for } i = 1. \\ I^s_{i-1}(f(I^{r+1}_2(u), \ldots, I^{r+1}_{r+1}(u)) & \text{for } 1 < i \leqq s+1. \end{cases}$$

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

$${}^c f = ({}^c f)_1 \hat{\ } \ldots \hat{\ } ({}^c f)_s$$

with $({}^c f)_i \in \mathscr{P}_{r+1}$. 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

$$(f.g)_i(u) = ((f.g).I_i^r)(u) = (f.(g.I_i^r))(u) = g_i(f_1(u), \ldots, f_q(u)).$$

As by induction hypothesis $f_i \in \mathscr{P}_p$ and $g_i \in \mathscr{P}_q$ and $\mathscr{P}$ is closed with respect to substitution it follows

$$f.g = (f.g)_1 \hat{} \ldots \hat{} (f.g)_r$$

with $(f.g)_i \in \mathscr{P}_p$.

Assume the thesis true for $f: N^r \to N^r$. It holds that

$$(f^R)_i(0, u) = I_i^r(u),$$

$$(f^R)_i(S(x), u) = (f.I_i^r)((f^R)_1(x, u), \ldots, (f^R)_r(x, u)).$$

As by induction hypothesis $f.I_i^r \in \mathscr{P}_r$ and $\mathscr{P}$ is closed with respect to simultaneous recursion it follows

$$f^R = (f^R)_1 \hat{} \ldots \hat{} (f^R)_r$$

with $(f^R)_i \in \mathscr{P}_{r+1}$.

(*b*) $\mathscr{S} \supseteq \{ f = f_1 \hat{} \ldots \hat{} f_s \mid f_i \in \mathscr{P}_r, r \geqq 0, s > 0 \}$.

As $\mathscr{S}$ is closed with respect to juxtaposition it suffices to show that $\mathscr{P}_r \in \mathscr{S}^{r,1}$ for every $r \geqq 0$.

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

Given $s+1$ functions $f, g_1, \ldots, g_s \in \mathscr{P}$ 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 $h \in \mathscr{S}$ because $h = (g_1 \hat{} \ldots \hat{} g_s).f$.

Given two functions $g, h \in \mathscr{P}$ 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 $f \in \mathscr{S}$ because

$$f = \Theta_{r+1}^{r+1} . {}^c\Delta^r . {}^{c^{r+1}}g . {}^{c^{r+1}}O^c . (({\Delta^{r+1}})^c . {}^{c^{r+1}}h . {}^{c^r}S^c)^R . I_{r+2}^{r+2}.$$

Note that, if $r=0$, Θ_{r+1}^{r+1} and ${}^c\Delta^r$ must be replaced by I and Δ respectively. □

2. CLASSES OF PRIMITIVE RECURSIVE SEQUENCE FUNCTIONS

In this section we define classes $\mathscr{I}_i$ of primitive recursive sequence functions and we study the behaviour of these classes with respect to the operators introduced in the previous section.

Let $\Sigma = \Sigma \cup \{\Theta, \Delta\}$ and, for a subset X of $\mathscr{S}$, let $\mathscr{C}(X)$ be the closure of X with respect to composition and left and right cylindrification and $\mathscr{R}(X)$ the set of functions obtained from X by repetition.

DEFINITION 2.1:

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

$$\mathscr{I}_{\omega a + b + 1} = \{ f = f_1 . f_2 \mid f_1 \in \mathscr{I}_{\omega a + b}, f_2 \in \mathscr{R}(\mathscr{I}_{\omega a}) \} \qquad \text{for every} \quad a, b \geqq 0,$$

$$\mathscr{I}_{\omega a} = \bigcup_{i < \omega} \mathscr{I}_{\omega (a-1)+i} \qquad \text{for} \quad a \geqq 1.$$

Note that Σ is the least set of functions such that $I_i^r \in \mathscr{I}_0$.

LEMMA 2.1: *The following properties hold:*

1° $\mathscr{I}_i \subseteq \mathscr{I}_{i+1}$ *for* $i \geqq 0$;

2° $\mathscr{C}(\mathscr{R}(\mathscr{I}_{\omega a})) = \mathscr{I}_{\omega (a+1)}$ *for* $a \geqq 0$;

3° $\mathscr{C}(\mathscr{I}_{\omega a}) = \mathscr{I}_{\omega a}$ *for* $a \geqq 0$;

4° *if* $f \in \mathscr{I}_{\omega a + b}$ *then* $f = f_0 . f_1 . \ldots . f_b$ *with* $f_0 \in \mathscr{I}_{\omega a}, f_1, \ldots, f_b \in \mathscr{R}(\mathscr{I}_{\omega a})$;

5° *if* $f \in \mathscr{I}_{\omega a + b}$ *then* $f^c \in \mathscr{I}_{\omega a + b}$ *and* ${}^c f \in \mathscr{I}_{\omega a + b}$.

Proof: It is immediate from definition 2.1. □

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

$$\mathscr{I}_{\omega^2} = \bigcup_{\substack{a < \omega \\ b < \omega}} \mathscr{I}_{\omega a + b} \qquad \text{and} \qquad \mathscr{I}_{\omega^2} = \mathscr{S}.$$

LEMMA 2.2: $\mathscr{R}(\mathscr{I}_{\omega a}) \subseteq \mathscr{I}_{\omega a + 1}$.

Proof: It is immediate by definition 2.1.

LEMMA 2.3: *If* $f \in \mathscr{I}_{\omega a + b}$, $g \in \mathscr{I}_{\omega a + c}$ *then* $f . g \in \mathscr{I}_{\omega a + b + c}$ *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 $b = 0$, $c \leqq n$. Let $f \in \mathscr{I}_{\omega a}$ and $g \in \mathscr{I}_{\omega a + n + 1}$, then $f . g \in \mathscr{I}_{\omega a + n + 1}$ by definition of $\mathscr{I}_{\omega a + n + 1}$.

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

As by induction hypothesis and lemma 2.1 $f_2 . g \in \mathscr{I}_{\omega a+1}$ then, again by induction hypothesis, it follows that $f_1 . f_2 . g \in \mathscr{I}_{\omega a+m}$. Assume the thesis is true for $b=m$ and $c < n$. Let $f \in \mathscr{I}_{\omega a+m}$ and $g \in \mathscr{I}_{\omega a+n}$. Then $f . g = f . g_1 . g_2$ where $g_1 \in \mathscr{I}_{\omega a+n-1}$ and $g_2 \in \mathscr{R}(\mathscr{I}_{\omega a})$. As by induction hypothesis $f . g_1 \in \mathscr{I}_{\omega a+n+m-1}$ then, by definition of $\mathscr{I}_{\omega a+m+n}$, $f . g \in \mathscr{I}_{\omega a+m+n}$.

LEMMA 2.4: *If* $f \in \mathscr{I}_{\omega a+b}$, $g \in \mathscr{I}_{\omega a+c}$ *then* $f \hat{\ } g \in \mathscr{I}_{\omega a+b+c}$, *for* $a, b, c < \omega$.

Proof: It follows from lemma 2.3 reminding that for $f : N^r \to N^p$ and $g : N^r \to N^s$, $f \hat{\ } g = \Delta^r . f^{c^r} . {}^{c^p}g$ and $\Delta^r \in \mathscr{I}_0$.

LEMMA 2.5: *If* $f \in \mathscr{I}_{\omega a+b}$, $g \in \mathscr{I}_{\omega a+c}$ *then* $f \times g \in \mathscr{I}_{\omega a+b+c}$, *for* $a, b, c < \omega$.

Proof: It follows from lemma 2.3 reminding that for f : $N^r \to N^p$ and $g : N^q \to N^s$, $f \times g = f^{c^q} . {}^{c^p}g$.

Let $\mathscr{I}_{\omega^2} = \bigcup\limits_{\substack{a<\omega \\ b<\omega}} \mathscr{I}_{\omega a+b}$.

LEMMA 2.6: (*a*) $\mathscr{C}(\mathscr{I}_{\omega^2}) = \mathscr{I}_{\omega^2}$;

(*b*) $\mathscr{R}(\mathscr{I}_{\omega^2}) = \mathscr{I}_{\omega^2}$;

(*c*) $\mathscr{I}_{\omega^2} = \mathscr{S}$.

Proof: (*a*) Let $f \in \mathscr{C}(\mathscr{I}_{\omega^2})$ then $f = f_1 . f_2$ where $f_1, f_2 \in \mathscr{I}_{\omega^2}$, by definition of $\mathscr{I}_{\omega^2}$ there exist a, b, c, such that $f_1 \in \mathscr{I}_{\omega a+b}$ and $f_2 \in \mathscr{I}_{\omega a+c}$. But by lemma 2.2 it follows that $f_1 . f_2 \in \mathscr{I}_{\omega a+b+c} \subseteqq \mathscr{I}_{\omega^2}$ then $\mathscr{C}(\mathscr{I}_{\omega^2}) \subseteqq \mathscr{I}_{\omega^2}$;

(*b*) analogously;

(*c*) by (*a*) and (*b*) and definition of $\mathscr{S}$ and $\mathscr{I}_{\omega^2}$. □

3. A HIERARCHY OF PRIMITIVE RECURSIVE SEQUENCE FUNCTIONS

In this section we show the strict containment of the class $\mathscr{I}_i$ in the class $\mathscr{I}_{i+1}$ (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 $\mathscr{I}_{\omega a+b}$ with $a \geqq 1, b < \omega$ is inspired by [4] and exploits properties of growth of the functions in $\mathscr{I}_i$ with respect to a proper ordering of the sequences.

We prove first some lemmas. In lemma 3.1 the strict containment of the class $\mathscr{I}_i$ in the class $\mathscr{I}_{i+1}$ ($i < \omega$) is shown. In lemma 3.2 the strict containment of the class $\mathscr{I}_{\omega+i}$ in the class $\mathscr{I}_{\omega+i+1}$ $(i < \omega)$ is shown under the normality hypothesis. In lemma 3.4 the normality property is stated for $\mathscr{I}_{\omega}$. From lemmas 3.1-3.4 theorem 3.1 follows which affirms the strict containment of the class $\mathscr{I}_i$ in the class $\mathscr{I}_{i+1}$ for every $i<\omega^2$.

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

DEFINITION 3.1: A subset X of $\mathscr{S}$ 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 $u = x_1, \ldots, x_r$,

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

with $F_0(x, y) = (h^c . {}^c(h . S) . h^R)(x, y)$.

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

Proof: Let $\mathrm{re}_m(x)$ the residue of the division of x by a constant m. Consider the following functions:

$$f_1(x) = m x,$$

$$f_2(x) = \begin{cases} x + m_1 & \text{if } \mathrm{re}_{m_1}(x) = 0, \\ C^1_{m_1}(x) & \text{otherwise,} \end{cases}$$

$$f_{i+1}(x) = \left\{ \begin{array}{ll} f_i(x) & \text{if } \mathrm{re}_{m_i}(x) = 0 \\ C^1_{m_i}(x) & \text{otherwise} \end{array} \right\} \quad \text{for } i \geqq 2,$$

with $m > 1$ and $\mathrm{re}_{m_i}(m_j) \neq 0$ 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 $f_i \in \mathscr{I}_i$ but not $f_i \in \mathscr{I}_{i-1}$, we obtain f_{i+1} from f_i by composition with a repetition on $C^1_{m_{i+1}}$ and therefore $f_{i+1} \notin \mathscr{I}_i$.

Now

$$f_1 = {}^cO . (S^m)^R.$$

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

$$f(x_1, \ldots, x_r) = (x_{i_1} + m_1, \ldots, x_{i_s} + m_s)$$

with $x_{i_j} \in \{x_1, \ldots, x_r\} \cup \{0\}$, $m_i \geqq 0$, $s > 0$, then $f_1 \in \mathscr{I}_1$ cannot be expressed in $\mathscr{I}_0$. Furthermore

$$f_2 = \Delta . \mathrm{re}^c_{m_1} . {}^cS^{m_1} . (C^1_{m_1})^R$$

where $\mathrm{re}_{m_1} = {}^cC^0_1 . {}^{c^2}C^0_2 . \ldots . {}^{c^{m_1-1}}C^0_{m_1-1} . {}^{c^{m_1}}C^0_0 . (\Theta^{m_1})^R . I^{m_1}_{m_1}$. As $\mathrm{re}_{m_1} \in \mathscr{I}_1 - \mathscr{I}_0$ and because we cannot spare the repetition, it follows that $f_2 \in \mathscr{I}_2 - \mathscr{I}_1$. Finally

$$f_{i+1} = \Delta \,.\, \mathrm{re}^{c}_{m_i, \ldots, m_1} \,.\, {}^{c^i}S^{m_1} \,.\, (I^i_1 \,.\, C^1_{m_i} \,.\, C^{0c}_0 \,.\, \ldots \,.\, C^{0c^{i-1}}_0)^R \,.\, (I^{i-1}_1 \,.\, C^1_{m_{i-1}} \,.\, C^{0c}_0 \,.\, \ldots \,.\, C^{0c^{i-2}}_0)^R \,.\, \ldots \,.\, (C^1_{m_1})^R$$

where

$$\mathrm{re}_{m_i, \ldots, m_1} = {}^{c}(C^0_1 \,.\, {}^{c}C^0_2 \,.\, \ldots \,.\, {}^{c^{m_i-2}}C^0_{m_i-1} \,.\, {}^{c^{m_i-1}}C^0_0) \,.\, \ldots \,.\, {}^{c^{m_i+\ldots+m_2+1}}(C^0_1 \,.\, {}^{c}C^0_2 \,.\, \ldots \,.\, {}^{c^{m_1-2}}C^0_{m_1-1} \,.\, {}^{c^{m_1-1}}C^0_0) \,.\, ((\Theta^{m_i})^{c^{m_{i-1}+\ldots+m_1}} \,.\, \ldots \,.\, {}^{c^{m_i+\ldots+m_2}}(\Theta^{m_1}))^R \,.\, I^{m_i}_{m_i} {}^\wedge \ldots {}^\wedge I^{m_1}_{m_1}$$

is the function such that $\mathrm{re}_{m_i, \ldots, m_1}(x) = (\mathrm{re}_{m_i}(x), \ldots, \mathrm{re}_{m_1}(x))$. As f_{i+1} can be obtained by compositions of functions in $\mathscr{I}_0$ with $\mathrm{re}_{m_i, \ldots, m_1}$ (which can be always computed by a single repetition) and i functions obtained by repetition of functions in $\mathscr{I}_0$, it follows $f_{i+1} \in \mathscr{I}_{i+1}$ and $f_{i+1} \notin \mathscr{I}_i$.

LEMMA 3.2: *Suppose that $\mathscr{I}_\omega$ is normal. Consider the functions $F_i : N^2 \to N$, $G_i : N \to N$, $H_i : N \to N$ defined as follows:*

$$F = h^c \,.\, {}^{c}(h \,.\, S) \,.\, h^R; \qquad F_{i+1} = F_i \,.\, \Delta \,.\, h^R;$$
$$G_i = (C^0_m)^c \,.\, F_i; \qquad H_i = \Delta \,.\, F_i$$

Then it holds that:

1° $F_i \in \mathscr{I}_{\omega+i+1}$, $G_i \in \mathscr{I}_{\omega+i}$, $H_i \in \mathscr{I}_{\omega+i+1}$;

2° for every $i < \omega$, for every $g : N^r \to N^s \in \mathscr{I}_{\omega+i}$ there exists $m \in N$ such that $g(u) < F_i(m, \max x_i)$;

3° for every $i < \omega$, for every $g : N \to N \in \mathscr{I}_{\omega+i}$ 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 $g : N^r \to N^s \in \mathscr{I}_{\omega+j}$. By definition $g = g_1 \,.\, g_2^R$, $g_1 \in \mathscr{I}_{\omega+j-1}$ and $g_2 \in \mathscr{I}_\omega$. By induction hypothesis there exist m_2, m'_i such that

$$(g_1 \,.\, I^{s+1}_i)(x_1, \ldots, x_r) < F_{j-1}(m'_i, \max x_i)$$
$$\leqq F_{j-1}(m_1, \max x_i) \qquad \text{for } m_1 = \max m'_i$$

and $g_2(y_1, \ldots, y_s) < F_0(m_2, \max y_i)$.

For every $y > 0$ it results

$$g_2^R(y, y_1, \ldots, y_s) < ((C^0_{m_2})^c \,.\, F_0)^R(y, \max y_i)$$

and then for every $(x_1, \ldots, x_r) \in N^r$:

$$g(x_1, \ldots, x_r) < ((C^0_{m_2})^c \cdot F_0)^R (F_{j-1}(m_1, \max x_i), F_{j-1}(m_1, \max x_i))$$
$$= ((C^0_{m_1})^c \cdot F_{j-1} \cdot \Delta \cdot (C^{0c}_{m_2} \cdot F_0)^R)(\max x_i)$$

Let $f_{m_1, m_2} = (C^0_{m_1})^c \cdot F_{j-1} \cdot \Delta \cdot ((C^0_{m_2})^c \cdot F_0)^R$. It holds that

$$f_{m_1, m_2}(x) = h^{h(m_2)}(S(\ldots(h^{h(m_2)}(S(F_{j-1}(m_1, x))\ldots))))$$
$$\leqq (h^{h(m_2)+1})^{F_{j-1}(m_1,x)}(F_{j-1}(m_1, x))$$
$$= h^R(((C^0_{m_1})^c \cdot F_{j-1} \cdot {}^cO \cdot (S^{h(m_2)+1})^R)(x), F_{j-1}(m_1, x)).$$

Now $(C^0_{m_1})^c \cdot F_{j-1} \cdot {}^cO \cdot (S^{h(m_2)+1})^R \in \mathscr{I}_{\omega+j-1}$ and then by induction hypothesis there exists $m > m_1, m_2$ such that $((C^0_{m_1})^c \cdot F_{j-1} \cdot {}^cO \cdot (S^{h(m_2)+1})^R)(x) < F_{j-1}(m, x)$ 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 $f: N \to N \in \mathscr{I}_{\omega+i}$ there exists $m \in N$ 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 $\mathscr{I}_\omega$ is normal with the function h then $\mathscr{I}_{2\omega}$ is normal with the function $h^* = H_0$.*

Proof: Consider $F^*_0 : N^2 \to N$, $G^*_0 : N \to N$ defined as follows

$$F^*_0 = h^{*c} \cdot {}^c(h^* \cdot S) \cdot h^{*R} \qquad \text{and} \qquad G^*_0 = (C^0_m)^c \cdot F^*_0.$$

By definition $h^* \in \mathscr{I}_{\omega+1} \subsetneqq \mathscr{I}_{2\omega}$, $F^*_0 \in \mathscr{I}_{2\omega+1}$, $G^*_0 \in \mathscr{I}_{2\omega}$. Suppose $g: N^r \to N^s$ and $g \in \mathscr{I}_{2\omega}$. In order to prove that there exists $m \in N$ 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 $n, i \in N$, there exists $m \in N$ such that, for every x, $F_i(n, x) < F^*_0(m, x)$. By the definitions of F_i and F^*_0 it results

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

and

$$F^*_0(m, x) = (\Delta \cdot h^c \cdot {}^ch \cdot {}^cS \cdot h^R)^R (F_0(m, m), F_0(x, x)+1).$$

Let $g = \Delta \cdot h^c \cdot {}^ch \cdot {}^cS \cdot h^R$. It is easy to prove by induction on n that, for every x, $F_0(n, x) < g^R(n, F_0(x, x)+1) \leqq F^*_0(m, x)$ where m is the least integer such that $n \leqq F_0(m, m)$. 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, $n \in N$ it results $F_i(n, x) < F^*_0(m, x)$, where m is the least integer such that $F_0(m, m) \geqq n+i$.

LEMMA 3.4: *$\mathscr{I}_\omega$ is normal with $h = \Delta \cdot S^R$.*

Proof: By induction on $\mathscr{I}_\omega$. We have shown that if $f \in \mathscr{I}_0$ then

$$f(x_1, \ldots, x_r) = (x_{i_1} + m_1, \ldots, x_{i_s} + m_s)$$

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

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

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

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

Suppose the thesis is true for $j < i$. Consider $f \in \mathscr{I}_i$. If $f \in \mathscr{I}_i$ then $f = f_1 . f_2^R$ with $f_1 \in \mathscr{I}_{i-1}$, $f_2 \in \mathscr{I}_0$. Then there exist m_j', m_2 such that

$$(f_1 . I_j^{r+1})(x_1, \ldots, x_r) < F_0(m_j', \max x_i) \leqq F_0(m_1, \max x_i)$$

where $m_1 = \max m_j'$ and $f_2^R(y, y_1, \ldots, y_r) < F_0(m_2, \max(y, y_i))$ and finally

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

The strict containment of the class $\mathscr{I}_{2\omega+i-1}$ in $\mathscr{I}_{2\omega+i}$ 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 $\mathscr{I}_{3\omega}$ can be proved to be normal with h^{**} where $h^{**} = H_0^* = \Delta . F_0^*$. By repeating the same reasoning up to $\mathscr{I}_{\omega^2}$ the following theorem can be stated.

THEOREM 3.1: $\mathscr{I}_i \subsetneqq \mathscr{I}_{i+1}$ *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:

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

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

(*b*) $X_i \leftarrow X_j$ where X_i and $X_j (i \neq j)$ are input variables or variables introduced before;

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

(*d*) LOOP X_i; I'_1; . . .; 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 $I_i (1 \leqq i \leqq k)$.

DEFINITION 4.2: A function $f : N^r \to N^s \in \mathscr{S}$ 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 $\mathscr{F}$ be the set of functions computed by the LOOP programs.

THEOREM 4.1: $\mathscr{F} = \mathscr{S}$.

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

Let $s = (X_1, \ldots, X_r)$ and $t = (X_{i_1}, \ldots, X_{i_q})$, 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 : IN s; OUT t computes the function $I^r_{i_1} \wedge \ldots \wedge I^r_{i_q}$ (in this case is always $q \leqq r$, $p = 0$).

Case 2: The program P : IN s; $X_i \leftarrow 0$; OUT t computes the function

$$ {}^{c^{i-1}}(C^1_0)^{c^{r-i}} . (I^r_{i_1} \wedge \ldots \wedge I^r_{i_q}), \qquad \text{if} \quad 1 \leqq i \leqq r, $$

and the function

$$ {}^{c^r} O . (I^{r+1}_{i_1} \wedge \ldots \wedge I^{r+1}_{i_q}), $$

otherwise.

The program P : IN s; $X_i \leftarrow X_j$; OUT t computes the function $T^r_{i,j} . (I^r_{i_1} \wedge \ldots \wedge I^r_{i_q})$.

The program P : IN s; $X_i \leftarrow X_i + 1$; OUT t computes the function ${}^{c^{i-1}}S^{c^{r-i}} . (I^r_{i_1} \wedge \ldots \wedge I^r_{i_q})$.

Case 3: Consider the program P : IN s; I_1; . . .; I_k; OUT t where I_i are instructions of the types *a, b, c, d*. Take the programs P_i : IN s_i; I_i; OUT t_i $(1 \leqq i \leqq k)$ with $s_1 = s$, $s_i = (s, s'_{i-1})$, for $1 < i \leqq k$, 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 $\mathscr{S}$. Then the function f computed by the program P can be written as composition of functions in $\mathscr{S}$ and therefore $f \in \mathscr{S}$.

Case 4: Consider the program P : IN s; LOOP X_i; I_1; . . . ; I_k; END; OUT t where the instructions I_i are of the types a, b, c, d. The function f computed by the program P is

$$ {}^{c^r}O \,.\, \ldots \,.\, {}^{c^{r+p-1}}O \,.\, {}^{c^{i-1}}\Delta^{c^{r+p-i}} \,.\, \Theta_i^{r+p+1} \,.\, f'^R \,.\, (I_{i_1}^{r+p} \hat{} \ldots \hat{} I_{i_q}^{r+p}), $$

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

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

Programs computing the functions of Σ 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 $f_1 . f_2$ 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. □

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_i; I; END where the instruction I contains at most a nested instructions of the type d. Let $M_{\omega a} = \bigcup_{i<\omega} M_{\omega (a-1)+1}$ for $a \geqq 1$.

We consider the following classes of primitive recursive sequence functions: $\mathcal{M}_i = \{ f \mid f \in \mathcal{S}$ and there exists $P \in M_i$ such that f computable by P $\}$ for $i < \omega^2$.

By the proof of theorem 4.1 (part b) and definition 4.3 we obtain that if $f_1 \in \mathcal{M}_{\omega a+b}$ and $f_2 \in \mathcal{M}_{\omega a+c}$ then $f = f_1 . f_2 \in \mathcal{M}_{\omega a+b+c}$ and if $f \in \mathcal{M}_{\omega a}$ then $f^R \in \mathcal{M}_{\omega (a+1)+1}$ for every $a \geqq 0$.

THEOREM 4.2: *$\mathcal{M}_{\omega a+b} = \mathcal{I}_{\omega a+b}$ 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 $f = f_1 . f_2$ with $f_1 \in \mathcal{I}_{\omega c+n-1}$, $f_2 \in \mathcal{I}_{\omega c+1}$ and by induction hypothesis there exist $P_1 \in M_{\omega c+n-1}$ and $P_2 \in M_{\omega c+1}$ computing f_1 and f_2 respectively. Then the program P computing f is a program in $M_{\omega c+n}$. Therefore $\mathcal{I}_{\omega c+n} \subseteqq \mathcal{M}_{\omega c+n}$.

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 $P_1 \in M_{\omega c+n-1}$ and $P_2 \in M_{c+1}$. 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 $f = f_1 . f_2 \in \mathcal{I}_{\omega c+n}$. Therefore $\mathcal{M}_{\omega c+n} \subseteqq \mathcal{I}_{\omega c+n}$. □

DEFINITION 4.4: Let P a program IN s; I; OUT s'' with $I=I_1; \ldots; I_n (n \geqq 0)$. 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

$$^{c^r}O \, . \, ^{c^{r+1}}O \, . \, \ldots \, . \, ^{c^q}O \, . \, t \, . \, I_{q+1}^{q+1}$$

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

(a)	$t=I_{q+1}^{q+1}$	if $n=0$;
(b)	$t={}^{c^{i-1}}S^{c^{q-i}} \, . \, {}^{c^q}S$	if $I : X_i \leftarrow X_i+1$;
	$t={}^{c^{i-1}}(C_0^1)^{c^{q-i}} \, . \, {}^{c^q}S$	if $I : X_i \leftarrow 0$;
	$t=T_{i,j}^q \, . \, {}^{c^q}S$	if $I : X_i \leftarrow X_j$
(c)	$t=t_1 \, . \, t_2$	if $I : I_1; I_2$

(d) $t={}^{c^{i-1}}\Delta^{c^{q-i}} \, . \, \Theta_i^{q+1} \, . \, (t_1 \, . \, {}^{c^q}(S \, . \, S))^R \, . \, {}^{c^q}(S \, . \, S)$ if I : LOOP X_i; I'; END.

LEMMA 4.1: *If* $P \in M_{\omega a+b}$ *then* $t_P \in \mathscr{I}_{\omega a+b}$ *for* $a, b \geqq 0$.

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 of 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 LOOP-END; finally T is the only output register. □

Now for the classes $M_{\omega a+b}$ 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 $g \in \mathscr{I}_{\omega m+n}$ 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* $P \in M_{\omega a+b}$. *Suppose that* $t_P(u) < F_n^{*m}(p, \max x_i)$ *for* $p \in N$ *and* $1 \leqq m < a$, $b > 0$. *Then there exists a program* $P' \in M_{\omega m+n'}$ *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 $M_{\omega+1}$ 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* $f \in \mathscr{S}$ *and a program* P *which computes* f, $f \in \mathscr{I}_{\omega a+b'}$ *iff* $t_P(u) < F_b^{*a}(p, \max x_i)$ *for a proper* p, *with* $b'=b+1$ *for* $a=1$, $b>0$ *and with* $b'=b$ *for* $a>1$, $b \geqq 0$.

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 $B=\{S=\lambda x.(x+1),\ O=(0),\ I_i^r=\lambda x_1, \ldots, x_r.(x_i)\}$.

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 $E_{\omega a+b}$, 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 $\mathscr{E}_i$, 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 $n \geqq 3$, and closed with respect to substitution and limited recursion (*see* [11, 15]).

The class $\mathscr{L}_i$ 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 $\mathscr{I}_i'$, for $i<\omega^2$, be the subclass $\mathscr{I}_i$ containing only functions with output sequence of length one, i.e., $\mathscr{I}_i'=\{f.I_j^s \mid f: N^r \to N^s \in \mathscr{I}_i,\ r \geqq 0,\ s>0\}$.

By the theorem 4.2 the following lemma holds.

LEMMA 5.1: *$\mathscr{L}_i=\mathscr{I}_{\omega i}'$ for $i<\omega$.*

The following lemmas express the relationship between the classes $\mathscr{I}_i'$ and the classes $\mathscr{E}_i$ and E_i.

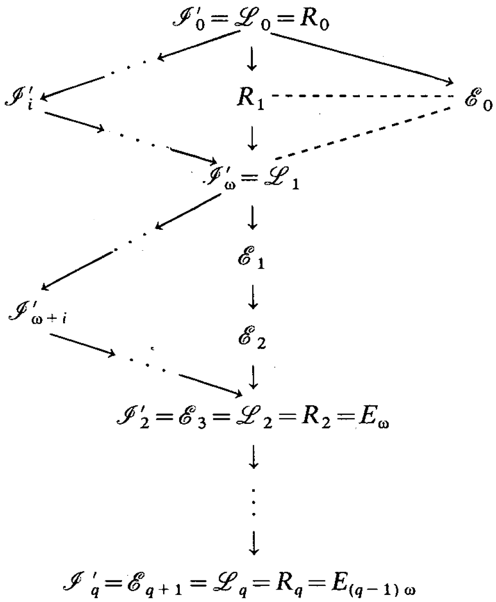
LEMMA 5.2: *$\mathscr{I}_{\omega i}'=\mathscr{E}_{i+1}$ for $i>1$.*

Proof: From [13] we have $\mathscr{L}_i=\mathscr{E}_{i+1}$ for $i>1$. From lemma 5.1 the thesis follows immediately. □

LEMMA 5.3: *$\mathscr{I}_{\omega i+j}'=E_{\omega(i-1)+j}$ for $i>1, j \geqq 0$.*

Proof: From [4] we have $E_{\omega i} = \mathscr{E}_{i+2}$ 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 →·B means that the class A is strictly enclosed in the class B; A --- B means that the two classes are not comparable.



Note that the classes $E_i\,(0 \leqq i < \omega)$ are not comparable with $\mathscr{E}_i\,(i < 3)$. The class $\mathscr{I}'_0 = \mathscr{L}_0 = R_0 \subsetneqq E_0$, but $\mathscr{L}_1$, R_1 are not comparable with $E_i\,(i < \omega)$, because the predecessor function is in $\mathscr{L}_1 \cap R_1$ and not in E_0, and the product function is in E but not in $\mathscr{L}_1$ or in R_1. The classes $\mathscr{I}'_i$ are contained in the classes E_i, for $0 \leqq i < \omega^2$.

The class of functions computed by Beck class $\bar{L}^i_j$ 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 $\mathscr{I}'_{\omega i + j}$, for $i, j \geqq 0$.

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 $\overline{\mathscr{I}}_0 = \mathscr{C}(\Sigma')$ and define new classes $\overline{\mathscr{I}}_i$ with a construction analogous to the one used for $\mathscr{I}_i$. The following lemma can be proved easily by induction.

LEMMA 5.4: *$E_i = \overline{\mathscr{I}}'_i$ 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 implicitely and the following assertion is proved: "if $f_i: N^r \to N \in E_{\omega a+b}$, then $f = (f_1, \ldots, f_s): N^r \to N^s \in E^{r,s}_{\omega a+b}$ for $a, b < \omega$". Now in lemma 2.4 we proved that if $f \in \mathscr{I}_{\omega a+b}$ then $(f_1\hat{}\ldots\hat{}f_s) \in \mathscr{I}_{\omega a+sb}$ but we can prove also the following lemma.

LEMMA 5.5: *If $f_i \in \mathscr{I}_{\omega a+b}$, for $a \geqq 2$, $b \geqq 0$, $1 \leqq i \leqq s$, then $f_1\hat{}\ldots\hat{}f_s \in \mathscr{I}_{\omega a+b}$.*

Proof: Consider the LOOP programs $\mathrm{P}_i \in \mathrm{M}_{\omega a+b}$ computing the functions $f_i \in \mathscr{I}_{\omega a+b}$. It is easy to construct a program P computing $f = f_1\hat{}\ldots\hat{}f_s$ 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 succesive loops each containing a nested loops, the program P will consist of sb successive loops each containing a nested loops, i.e. $\mathrm{P} \in M_{\omega a+sb}$. For $a \geqq 2$ and $b \geqq 0$ 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° $$\mathscr{I}_{\omega(a+1)+b} = \bigcup_{r,s} E^{r,s}_{\omega a+b} \quad \textit{for} \quad a \geqq 1,\ b \geqq 0.$$

2° $$\overline{\mathscr{I}}_{\omega a+b} = \bigcup_{r,s} E^{r,s}_{\omega a+b} \quad \textit{for} \quad a,\ b \geqq 0.$$

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 $\mathscr{I}_1$ and the product is in $\mathscr{I}_{\omega+1}$, 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^i_j (*see* [3, 13, 17]) are generalized immediately to the classes $\mathscr{I}_i$.

The following assertions hold:

the equivalence problem is recursively unsolvable in $\mathscr{I}_{\omega a+b}$, for $a \geqq 1$, $b \geqq 2$;

the equivalence problem for functions of is recursively solvable $\mathscr{I}_{\omega}$ in $\mathscr{I}_{2\omega}$;

the problem of determining the least a, b such that f belongs to $\mathscr{I}_{\omega a+b}$ is recursively unsolvable for $a \geqq 2$, $b \geqq 1$;

there is an algorithm which for a given function $f \in \mathscr{I}_i (i<\omega)$ determines whether the expression defining f contains a repetition on a constant, so that in such a case $f \in \mathscr{I}_{i-1}$.

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 $\mathscr{I}_i^P$, $\mathscr{I}_i^t$ having as base $\Sigma \cup \{P\}$ and $\Sigma \cup \{t\}$, with $P = \lambda x . (x \dot{-} 1)$ and $t = \lambda x, y, z$. (if $z=0$ *then* x 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 $\mathscr{I}_{\omega}^P$ and $\mathscr{I}_{\omega}^t$.

REFERENCES

1. G. AUSIELLO, *Complessità di calcolo delle funzioni, Boringhieri*, Torino, 1975.
2. P. AXT, *Iteration of Primitive Recursion*, Zeisch. f. math. Logik und Grundl. d. Math., Vol. 9, 1965, pp. 253-255.
3. H. BECK, *Zur Entscheidbarkeit der funktionalen Aquivalenz*, Automata Theory and Formal Languages 2nd GI Conference, Lecture Notes in Computer Science, Vol. 33, 1975, pp. 127-133.
4. J. P. CLEAVE, *A Hierarchy of Primitive Recursive Functions*, Zeitsch. f. math. Logik und Grundl. d. Math., Vol. 9, 1963, pp. 331-345.
5. A. COBHAM, *The Intrinsic Computational Difficulty of Functions*, Proc. Congress on Logic, Methodology and Philosophy of Science, Haifa, Israel, 1964, North-Holland, Amsterdam, 1964, pp. 24-30.
6. S. EILENBERG and C. C. ELGOT, *Iteration and Recursion*, Proc. Nat. Acad. Sci. U.S.A., Vol. 61, 1968, pp. 378-379.
7. S. EILENBERG and C. C. ELGOT, *Recursiveness*, Academic Press, New York, 1970.
8. G. GERMANO and A. MAGGIOLO-SCHETTINI, *Quelques caractérisations des fonctions récursives partielles*, C.R. Acad. Sc. Paris, t. 276, série A, 1973, pp. 1325-1327.
9. G. GERMANO and A. MAGGIOLO-SCHETTINI, *Sequence-to-Sequence Recursiveness*, Information Processing Lett., Vol. 4, 1975, pp. 1-6.
10. G. GERMANO and A. MAGGIOLO-SCHETTINI, *Proving a Compiler Correct: a Simple Approach*, J. Comput. System Sc,. Vol. 10, 1975, pp. 370-383.
11. A. GRZEGORCZYK, *Some Classes of Recursive Functions*, Rozprawy Mathematyczne, Vol. 4, 1953, pp. 1-45.

12. H. HUWIG and V. CLAUS, *Das Äquivalenzproblem für spezielle Klassen von* LOOP-*Programmen*, Theoretical Computer Science 3rd GI Conference, Lecture Notes in Computer Science, Vol. 48, 1977, p. 73-82.

13. A. R. MEYER and D. M. RITCHIE, *The Complexity of* LOOP*Programs*, Proc. 22nd A.C.M. Nat. Conference, Washington, D.C., 1968, pp. 465-469.

14. H. MÜLLER, *Characterization of the Elementary Functions in Terms of Nesting of Primitive Recursions*, Recursive Function Theory Newsletters, Vol. 5, 1973.

15. R. W. RITCHIE, *Classes of Recursive Functions Based on Ackermann's Function*, Pacific J. Math., Vol. 15, 1965, pp. 1027-1044.

16. H. SCHWICHTENBERG, *Rekursionszahlen und die Grzegorczyk Hierarchie*, Arch. Math. Logik Grundlagenforsch., Vol. 12, 1969, pp. 85-97.

17. D. TSICHRITZIS, *A note on Comparison of Subrecursive Hierarchies*, Information Processing Lett., Vol. 1, 1971, pp. 42-44.

18. D. TSICHRITZIS, *The Equivalence Problem of Simple Programs*, J. Ass. Comput. Mach, Vol. 17, 1970, pp. 729-738.