$d(i)=\sigma_i$, $i=1,\dots,m$. If $a=[u_1 \dots u_n u_{n+1}]$, $u_1 \dots u_{n+1}$ in normal form, then $\Phi^{-1}(a)$ is computed by means of $\Phi^{-1}(u_1 \dots u_n)=\langle D', \sim', d' \rangle$ of the following algorithm :

```
begin
{Let u_{n+1}=σ_1⊗...⊗σ_m ; slice_n(D)=[y_1]∪...∪[y_p], [y_1]⊆...⊆[y_p]}
D=D' ; ~=~' ; d=d' ;
for i : =1 to m do D_i=Φ ;
i : =1 ; ic : =1 ;
for k : = 1 to p do
      for j : = 1 to arity(d'(y_k)) do
            begin
            if ic > coarity(σ_i) then
                  begin i : = i+1 ;  ic : = 1 ; end
            D_i : = D_i ∪ {xj/x∈[y_k]} ;
            ic : = ic+1 ;
            end ;
      for i : = 1 to m do
            begin
            D : =D ∪ D_i ;
            for each ⟨x, y⟩∈D_i × D_i do x~y ;
            for each x∈D_i do d(x)=σ_i ;
            end
end.
```

Theorem 5.4. *$PDAG(\Sigma)$ is isomorphic with $T(\Sigma)^{\#}$.*

Proof. By Theorem 5.16 of [3] there exist $\Phi^{\#}$ and $\Phi^{-1\#}$ such that the following diagrams comute :

$$\begin{tikzcd} \mathrm{FPDAG}(\Sigma) \arrow[r, "\wr"] \arrow[dr, "\Phi"'] & \mathrm{PDAG}(\Sigma) \simeq \mathrm{FPDAG}(\Sigma)^{\#} \arrow[d, "\Phi^{\#}"] \\ & \mathrm{T}(\Sigma)^{\#} \end{tikzcd} \qquad \begin{tikzcd} T(\Sigma) \arrow[r, "\wr"] \arrow[dr, "\Phi^{-1}"'] & T(\Sigma)^{\#} \arrow[d, "\Phi^{-1\#}"] \\ & \mathrm{PDAG}(\Sigma) \end{tikzcd}$$

Because Φ is bijective if follows that $(\Phi^{\#})^{-1}=\Phi^{-1\#}$.

6. Conclusions. The results proved in this paper transfer the fundamental properties of partial trees [2], [3] to dags. By Theorem 5.3 the set $\mathrm{FPDAG}(\Sigma)$ is structured as a magmoid if we are setting $\delta_1 . \delta_2=\Phi^{-1}(\Phi(\delta_1) . \Phi(\delta_2))$, $\delta_1 \otimes \delta_2=\Phi^{-1}(\Phi(\delta_1) \otimes \Phi(\delta_2))$, $1=\langle \{1\}, \{[1]\}, \{\langle 1,1 \rangle\} \rangle$, $0=\langle \Phi, \Phi, \Phi \rangle$. Now Φ is an isomorphism of ordered magmoids. This isomorphism can be obtained also in other way : we first structure $\mathrm{FPDAG}(\Sigma)$ as a magmoid by defining the operation . and $\otimes$, and then we find the isomorphism between the two magmoids. This way we consider more incovenient.

REFERENCES

1. B r u n o, W. — *Rationalité et reconnaissabilité dans des graphes*, Thèse pour obtenir le titre de docteur, Laboratoire d'Informatique Fondamentale de Lille, 1986.
2. G u e s s a r i a n, I. — *Algebraic Semantics*, Raport LITP 80—13 Université Paris 7, 1980.
3. T h a t c h e r, J. W., W a g n e r, E. G., W r i g h t, J. B. — *Notes on algebraic fundamentals for theoretical computer science* Mathematical Centre Tracts 109, Amsterdam, 1979.

Received 1. VII.1989

Seminarul Matematic „Al. Myller"
Universitatea „Al. I. Cuza" Iași
6600 Iași, România

An. şt. Univ. „Al. I. Cuza" Iaşi
Informatică, 35 (1989)

A HIERARCHY OF UNARY PRIMITIVE RECURSIVE STRING-FUNCTIONS

BY

LILA SÂNTEAN

1. Introduction. Let N be the set of naturals i.e. $N=\{0, 1, 2, \dots\}$. Consider a fixed alphabet $A=\{a_1, a_2, \dots, a_r\}$, $r \geqslant 2$ and denote by A^* the free monoid generated by A under concatenation (with e the null string). The elements of A^* are called strings ; if reffering to strings, $<$ denotes the lexicographical order induced by $a_1<a_2<\dots<a_r$.

Denote by Fnc (respectively Fnc_A) the set of all unary number-theoretical (respectively, string) functions. By I, Succ, C_m, Pd we denote the following number-theoretical functions : $I(x)=x$, $\mathrm{Succ}(x)=x+1$, $C(x)=m$, $\mathrm{Pd}(x)=x \dot{-} 1$, where $x \dot{-} y=\max(x-y, 0)$. By I^A, Succ_i^A, C_u^A, σ^n, π, we denote the following string-functions: $I^A(w)=w$, $\mathrm{Succ}_i^A(w)=wa_i$ $(1 \leqslant i \leqslant r)$, $C_u^A(w)=u$, $\sigma(e)=a_1$, $\sigma(wa_i)=wa_{i+1}$ if $1 \leqslant i<r$ and $\sigma(wa_r)=\sigma(w)a_1$, $\pi(e)=e$, $\pi(\sigma(w))=w$. Furthermore one uses the primitive recursive bijections $c : A^* \to N$, $\bar{c} : N \to A^*$ given by $c(e)=0$, $c(wa_i)=rc(w)+i$, $1 \leqslant i \leqslant r$, and $\bar{c}(0)=e$, $\bar{c}(m+1)=\sigma(\bar{c}(m))$.

To each f in Fnc one associates the string-function $s(f) \in \mathrm{Fnc}_A$ defined by $s(f)(w)=\bar{c}(f(c(w)))$ and for each g in Fnc one associates the number-theoretical function $n(g)$ defined by $n(g)(x)=c(g(\bar{c}(x)))$. It is easily seen that for every string-function g, $s(n(g))=g$ and for every number-theoretical function f, $n(s(f))=f$. For example, $s(\mathrm{Succ})=\sigma$, $n(I^A)=I$, $s(\mathrm{Pd})=\pi$.

A mapping from Fnc^n to Fnc is called an operator in Fnc, and analogously for Fnc_A. We consider the following operators in Fnc and Fnc_A :

$\mathrm{sub}(f, g)=h$ iff $f, g, h \in \mathrm{Fnc}$ and $f(g(x))=h(x)$;

$\mathrm{diff}(f, g)=h$ iff $f, g, h \in \mathrm{Fnc}$ and $h(x)=f(x) \dot{-} g(x)$;

$\mathrm{it}_x(f)=h$ iff $f, h \in \mathrm{Fnc}$ and $h(0)=x$, $h(y+1)=f(h(y))$;

$\mathrm{sub}_A(f, g)=h$ iff $f, g, h \in \mathrm{Fnc}_A$ and $f(g(w))=h(w)$;

$\sigma-\mathrm{it}_{A,w}(f)=h$ iff $f, h \in \mathrm{Fnc}_A$ and $h(e)=w$, $h(\sigma(u))=f(h(u))$.

To every operator φ in Fnc, $s(\varphi)(f)=s(\varphi(n(f)))$, for every $f \in \mathrm{Fnc}_A$; analogously, for every operator θ in Fnc_A, $n(\theta)(g)=n(\theta(s(g)))$, for every $g \in \mathrm{Fnc}$. For example, $s(\mathrm{it}_x)=\sigma-\mathrm{it}_{A,\bar{c}(x)}$, $n(\sigma-\mathrm{it}_{A,w})=\mathrm{it}_{c(w)}$.

2. Ackermann-Peter string-function. The primitive recursive string-functions were introduced by A s s e r [1] and studied by various authors (see [4], [6], [8]).

In order to study the complexity of such functions, we use as a measure of complexity the growth relatively to the lexicographical order. To this aim we use the string-version of the Ackermann-Peter unary function defined by W e i c h r a u c h [8]. The function, denoted by $A : A^* \to A^*$, is given by means of the following three equations:

$$A_0(x) = (x) \qquad \text{(i)}$$

$$A_{n+1}(e) = A_n(a_1) \qquad \text{(ii)}$$

$$A_{n+1}(\sigma(x)) = A_n(A_{n+1}(x)) \qquad \text{(iii)}$$

The following technical results concern the monotonicity properties of the function A; they generalize the monotonicity properties of the number-theoretical Ackermann-Peter functions (see [4]).

Lemma 1. *For all naturals n and for all strings x over A, we have $A_n(x) > x$.*

Proof. We proceed by induction on n. For $n=0$ we have $A_0(x)=\sigma(x)>x$. We assume that $A_n(x)>x$ and we prove the inequality $A_{n+1}(x)>x$ by induction on x. For $x=e$, $A_{n+1}(e)=A_n(a_1)>a_1>e$. Suppose now that $A_{n+1}(x)>x$. We use (iii) and the first induction hypothesis to get $A_{n+1}(\sigma(x))=A_n(A_{n+1}(x))>A_{n+1}(x)$. Finally, by the second induction hypothesis, $A_{n+1}(x)\geq\sigma(x)$, we obtain $A_{n+1}(\sigma(x))>\sigma(x)$.

Lemma 2. *For all naturals n and for all strings x over A, we have: $A_n(x)<A_n(\sigma(x))$.*

Proof. For $n=0$, $A_0(x)=\sigma(x)<\sigma(\sigma(x))<A_0(\sigma(x))$.
Assume that $A_n(x)<A_n(\sigma(x))$. In view of (iii) and Lemma (1) we have $A_{n+1}(\sigma(x))=A_n(A_{n+1}(x))>A_{n+1}(x)$.

Corollary 3. *For all naturals n and all strings x, y from A^*, if $x<y$, then $A_n(x)<A_n(y)$.*

Lemma 4. *For all naturals n and for all strings x over A, we have: $A_n(x)<A_{n+1}(x)$.*

Proof. We proceed by double induction on n and x.
For $n=0$ we have $A_0(x)=\sigma(x)<\sigma(\sigma(x))=A_1(x)$. Assume now that $A_n(x)<A_{n+1}(x)$ and we prove that $A_{n+1}(x)<A_{n+2}(x)$ by induction on x. For $x=e$, in view of (ii) and the first induction hypothesis, we get $A_{n+1}(e)=A_n(a_1)<A_{n+1}(a_1)=A_{n+2}(e)$. In view of a new induction hypothesis, $A_{n+1}(x)<A_{n+2}(x)$, we deduce the relations $A_{n+1}(\sigma(x))=A_n(A_{n+1}(x))<A_n(A_{n+2}(x))<A_{n+1}(A_{n+2}(x))=A_{n+2}(\sigma(x))$ (we have also used the first induction hypothesis, relation (iii) and Corollary (3)).

Corollary 5. *For all naturals n and m, if $n<m$, then $A_n(x)<A_m(x)$.*

Lemma 6. *For all strings x over A we have: $A_2(x)=\sigma^{2.c(e)+3}(e)$.*

Proof. We proceed by induction on x.
For $x=e$, in view of (ii) we have $A_2(e)=A_1(a_1)=\sigma(\sigma(a_1))=\sigma^3(e)=\sigma^{2.c(e)+3}(e)$. Assuming that $A_2(x)=\sigma^{2.c(x)+3}(e)$, we have to prove that $A_2(\sigma(x))=\sigma^{2.c(\sigma(x))+3}(e)$. Indeed, using (iii) and the equality $c(\sigma(x))=c(x)+1$, we get: $A_2(\sigma(x))=A_1(A_2(x)=A_1(\sigma^{2.c(x)+3}(e))=\sigma^{2.c(x)+5}(e)=\sigma^{2.c(\sigma(x))+3}(e)$.

Lemma 7. For all naturals k and $n\geq 1$, there exists a natural i (which depends on k) such that $A_n(\sigma^k(x))<A_{n+1}(\pi^k(x))$, for every string x with $c(x)>i$.

Proof. We first notice that for every string x with $c(x)>3k-1$, we have $\sigma^k(x)<A_2(\pi^{k+1}(x))$. Indeed by Lemma (6) we have $A_2(\pi^{k+1}(x))=\sigma^{2.c(\pi^{k+1}(x))+3}(e)=$ $=\sigma^{2(c(x)+k+1)+3}(e)=\sigma^{2.c(x)+2k+1}(e)>\sigma^{k+c(x)}(e)=\sigma^k(\sigma^{c(x)}(e))=\sigma^k(x)$. Consequently, using Corollary (3) and Corollary (5), $A_n(\sigma^k(x))<A_n(A_2(\pi^{k+1}(x)))<A_n(A_{n+1}$ $\pi(^{k+1}(x)))=A_{n+1}(\pi^{k+1}(x))$, for all strings x with $c(x)>3k\dot{-}1$. In conclusions, we can take $i=3k\dot{-}1$.

Lemma 8. *For all naturals n and strings x we have $A_{n+1}(x)=A^{c(x)+1}(a_1)$.*

Proof. We proceed by induction on x. For $x=e$, using (ii) we obtain $A_{n+1}(e)=A_n(a_1)=A_n^{c(e)+1}(a_1)$. Assuming that $A_{n+1}(x)=A_n^{c(n)+1}(a_1)$ we prove that $A_{n+1}(\sigma(x))=A_n^{c(\sigma(x))+1}(a_1)$. Indeed, using (iii) we get $A_n^{c(\sigma(x))+1}(a_1)=A_n^{c(x)+2}(a_1)=$ $=A_n(A_n^{c(x)+1}(a_1))=A_n(A_{n+1}(x))=A_{n+1}(\sigma(x))$.

The monotonicity properties of the string Ackermann-Peter function will be freely used in what follows.

3. A hierarchy of unary primitive recursive string functions. We are going to define an increasing sequence $(C_n)_{n\geq 0}$ of string-function classes whose union equals the class of the one-argument primitive recursive string-functions.

Definition 9. We say that the function $f: A^*\to A^*$ is defined by *limited iteration at e* (shortly, *limited iteration*) from the functions $g: A^*\to A^*$ and $h: A^*\to A^*$ if it satisfies the following equations

$$f(e)=e,\ f(\sigma(x))=g(f(x)),\ f(x)\leq h(x), \text{ for every } x \text{ in } A^*.$$

Definition 10. For a natural n we define C_n to be the smallest class of unary primitive recursive string-functions which contains the functions A_0, A_n and is closed under composition, limited iteration and s(diff) (the string-function operation corresponding to the arithmetical difference).

Lemma 11. *For all naturals n, the class C_n contains the functions C_e^A, I^A, π and the functions $l_i(1\leq i\leq r)$,* sg *and* $\overline{\text{sg}}$ *defined by*: $l_i(w)=a_i$, $\forall 1\leq i\leq r$, $\forall w\in A^*$

$$\text{sg}(w)=\begin{cases} e & \text{if } w=e \\ a_1 & \text{if } w\neq e\end{cases} \qquad \overline{\text{sg}}(w)=\begin{cases} a_1 & \text{if } w=e \\ e & \text{if } w\neq e.\end{cases}$$

Proof. It follows from the following equalities

$$C_e^A=s(\text{diff})\,(A_0,\ A_0),\quad l_i=A_0^i(e),\quad 1\leq i\leq r$$

$$I^A=s(\text{diff})\,(A_0,\ l_1),\quad \overline{\text{sg}}=s(\text{diff})\,(l_1,\ I^A),$$

$$\text{sg}=s(\text{diff})\,(l_1\ \overline{\text{sg}}),\quad \pi=s(\text{diff})\,(I^A,\ l_1),$$

and from the Definition (10).

Proposition 12. *For all naturals n, $C_n\subseteq C_{n+1}$.*

Proof. We shall prove by induction on n that for all naturals n and k, $A_n\in C_{n+k}$.

If $n=0$, by Definition (10), $A_0\in C_m$, for every natural m. Assume that $A_n\in C_{n+k}$. We shall prove that $A_{n+1}\in C_{n+k+1}$.

Assertion. For every string x, $A_{n+1}(x)=f(\sigma(x))$, where $f(e)=e$, $f(\sigma(x))=$ $=A_n(g(f(x)))$ and $g(x)=s(\text{diff})\,(\sigma(x),\ \text{sg}(x))$. The equalities will be proven by induction on the string x. If $x=e$, from the definitions of the functions A_n and s (diff) we deduce:

$$f(\sigma(e)) = A_n(g(f(e))) = A_n(g(e)) = A_n(s(\mathrm{diff})\,(\sigma(e), \mathrm{sg}\,(e))) = A_n(s(\mathrm{diff})\,(a_1, e)) =$$
$$= A_n(a_1) = A_{n+1}(e).$$

Supposing now that $A_{n+1}(x) = f(\sigma(x))$, we shall show that $A_{n+1}(\sigma(x)) = f(\sigma^2(x))$.

Indeed, $f(\sigma(\sigma(x))) = A_n(g(f(\sigma(x)))) = A_n(g(A_{n+1}(x))) = A_n(s(\mathrm{diff})\,(\sigma(A_{n+1}(x)), \mathrm{sg}(A_{n+1}(x))) = A_n(s(\mathrm{diff})\,(\sigma(A_{n+1}(x)), a_1) = A_n(\bar{c}(\mathrm{diff}(c(\sigma(A_{n+1}(x))), c(a)))) = A(\bar{c}(\mathrm{diff}\,(c(A_{n+1}(x) + 1, 1))) = A_n(c(c(A_{n+1}(x)))) = A_n(A_{n+1}(x)) = A_{n+1}(\sigma(x))$.

Using now Definition (10), Lemma (11) and the relations $f(x) = A_{n+1}(\pi(x)) \leqslant A_{n+1}(x) \leqslant A_{n+k+1}(x)$, $x \in A^*$, we deduce that A_{n+1} is in C_{n+k+1} as being obtained by limited iteration from f and A_{n+k+1}, both belonging to C_{n+k+1}.

Lemma 13. *For all naturals n and all functions f in C_n, there exists a natural k such that $f(x) < A_n^k(x)$, for every string x in A^*.*

Proof. We shall make use of the inductive definition of C_n. If $f(x) = A_0(x)$ then $f(x) < A_0(A_c(x)) \leqslant A_n(A_n(x))$ and we can take $k = 2$. If $f(x) = A_n(x)$, then $f(x) \leqslant A_n(A_n(x))$ and we can also take $k = 2$. If $f(x) < A_n^p(x)$ and $g(x) < A_n^q(x)$, for all strings x in A^* then $(f \circ g)\,(x) = f(g(x)) < A_n^p(g(x)) < A_n^{p+q}(x))$; $s(\mathrm{diff})\,(f, g)\,(x) \leqslant f(x) < A_n^p(x)$;

Finally, if f is obtained by limited iteration from g and h, $h(x) < A_n^k(x)$ then $f(x) \leqslant h(x) < A_n^k(x)$.

Theorem 14. *For every class C_n, $n \geqslant 1$, and every f in C_n, there exists a natural i (depending on f) such that $f(x) < A_{n+1}(x)$ for every string x satisfying $c(x) \geqslant i$.*

Proof. Assume that f is a function in C_n, $n \geqslant 1$. In view of Lemma (13), we can find a natural $k \geqslant 2$ (which depends on f) such that, for every string x, $f(x) < A_n^k(x)$. We shall show that the requested inequality holds for $i = 3k$.

From the monotonicity properties of Ackermann-Peter string-function, one can deduce the following relations: $A_n^k(x) = A_n^{k-1}(A_n(x)) \leqslant A_n^{k-1}(A_n(\sigma^{k+1}(x)))) < A_n^{k-1}(A_{n+1}(\pi^{k-1}(x)))$, for every string x with $c(x) > 3k + 1$.

Intermediate step. $A_{n+1}(x) = A_n^{k-1}(A_{n+1}(\pi^{k-1}(x)))$, for every string x with $c(x) \geqslant k$. We shall prove the equality by induction on x. If $c(x) = k$, then we have $A_n^{k-1}(A_{n+1}(\pi^{k-1}(x))) = A_n^{k-1}(A_{n+1}(\pi^{k-1}(\sigma^{c(x)}((e))) = A_n^{k-1}(A_{n+1}(\pi^{k-1}(\sigma^k(e))) = A_n^{k-1}(A_{n+1}(a_1)) = A_n^{k-1}(A_n^2(a_1)) = A_n^{k+1}(a_1) = A_n^{c(x)+1}(a_1) = A_{n+1}(x)$.

If the equality holds for x, we can prove that $A_{n+1}(\sigma(x)) = A_n^{k-1}(A_{n+1}(\pi^{k-1}(x))))$. Indeed, $A_n^{k-1}(A_{n+1}(\pi^{k-1}(\sigma(x)))) = A_n^{k-1}(A_{n+1}(\sigma(\pi^{k-1}(x))) = (A_n^{k-1} A_n(A_{n+1}(\pi^{k-1}(x))) = A_n(A_n^{k-1}(A_{n+1}(\pi^{k-1}(x)))) = A_n(A_{n+1}(x)) = A_{n+1}(\sigma(x))$, and the intermediate step is proved.

Returning to the proof of the theorem, we can now write $f(x) < A_n^k(x) < A_n^{k-1}(A_{n+1}(\pi^{k-1}(x))) = A_{n+1}(x)$, for all strings x with $c(x) \geqslant 3k + 1$ and taking $i = 3k + 1$, the proof is finished.

Theorem 15. *The set $\bigcup_{n=0}^{\infty} C_n$ coincides with the set of unary primitive recursive string functions.*

Proof. We shall make use of the characterization of the set of unary primitive recursive string-functions obtained in [5], namely as the smallest class of unary string-functions which contains σ and is closed under the operations sub_A, $\sigma - \mathrm{it}_{A,e}$, $s(\mathrm{diff})$.

It is obvious that every function in $\bigcup_{n=0}^{\infty} C_n$ is primitive recursive. For the converse inclusion, all that remains to prove is reduced to the closure of $\bigcup_{n=0}^{\infty} C_n$ to $\sigma - \mathrm{it}_{A,e}$.

We shall show that if $f \in \bigcup_{n=0}^{\infty} C_n$ is obtained by pure iteration from $g \in \bigcup_{n=0}^{\infty} C_n$, there exists a function $h \in \bigcup_{n=0}^{\infty} C_n$ such that f is obtained by limited iteration from g and h and, therefore, f is in $\bigcup_{n=0}^{\infty} C_n$. Indeed, let f be obtained by pure iteration from g in C_m, $m > 0$. We shall prove, by induction on the string x that f is majorated by A_{n+1}. If $x = e$, we have $f(e) = e < A_{n+1}(e)$.

Supposing that $f(x) < A_{n+1}(x)$ and using the definition and the monotonicity properties of Ackermann-Peter function, we get: $f(\sigma(x)) = g(f(x)) < A_n(f(x)) < A_n(A_{n+1}(x)) = A_n(\sigma(x))$.

Theorem 16. *The function $\bar{A} : A^* \to A^*$ defined by $\bar{A}(w) = A_{c(w)}\,(w)$ is not primitive recursive.*

Proof. Assume that $\bar{A}$ is primitive recursive. From Theorem (15) we get a natural n such that $\bar{A} \in C$. By Theorem (14), there exists a natural i such that $A(x) < A_{n+1}(x)$ for every x with $c(x) \geqslant i$. Let x be a word satisfying the condition $c(x) = n + i + 1$. We arrive at a contradiction since $\bar{A}(x) = A_{c(x)}(x) = A_{n+i+1}(x) < A_{n+1}(x)$ (see Corollary (5)). This ends the proof of the theorem.

Acknowledgements. I am grateful to dr. Cristian Calude for drawing my attention to these problems and for many helpful remarks.

REFERENCES

1. Asser, G. — *Rekursive Wortfunktionen*, Z. Math. Logik Grundlag. Math. 6 (1960), 258—278.
2. Asser, G. — *Primitive recursive word-functions of one variable*, in E. Borger (ed.), Computation Theory and Logic, LNCS 270, Springer 1987, 14—19.
3. Asser, G. — *Zur Robinson Charakterisierung der Einstelligen Primitiv Rekursiven Wortfunktionen*, Z. Math. Logik Grundlag. Math., 34 (1988), 317—322.
4. Calude, C. — *Theories of Computational Complexity*, North-Holland, Amsterdam, New York, Oxford, Tokio, 1988.
5. Calude, C., Santean, L. — *On a Theorem of Gunter Asser*, Z. Math. Logik Grundlag. Math., 1 (1990).
6. Henke, F. W. V., Indermark, K., Rose, G., Weichrauch, K. — *On Primitive Recursive Wordfunctions*, Computing 15 (1975), 217—231.
7. Tătărâm, M. — *Darboux Property and one-argument primitive recursive string functions*, Revue Roumaine de Mathematiques Pures et Appliques, XXXII (1987), 79—94.
8. Weichrauch, K. — *Teilklassen primitiv-rekursiver Wortfunktionen*, Berichte der GMD 91 (1974), 1—49.

Received 1.VI.1989

Institute for Computers and Informatics
8—10 Miciurin, 71316
Bucharest 1, Romania