BULLETIN DE L'ACADÉMIE
POLONAISE DES SCIENCES
Série des sciences math., astr.
et phys. — Vol. XXII, No. 6, 1974

MATHEMATICS
(COMPUTER SCIENCE)

Not Programmable Function Defined by a Procedure

by

W. DAŃKO

Presented by A. MOSTOWSKI on January 12, 1974

Summary. There are relational systems for which the class of functions computable by means of programs with procedures is essentially larger than the class of functions computable by means of programs without procedures.

We give examples of relational systems with the following property: the class of functions defined as the least solution of the system of equations contains functions which do not belong to the class of functions computable by means of Scott-Engeler programs.

Since some systems of equations may be regarded as procedures, we can derive a conclusion interesting for the theory of programming (see Summary).

First we will give a few definitions helpful in the construction of such systems and in the proofs of their properties.

Let A be an alphabet consisting of: an infinite set of variables V; zero-argument functional symbols a, b; a two-argument functional symbol P and a two-argument relational symbol $=$.

We assume the sets T, F and S of terms, formulas and substitutions to be defined in the normal way.

We denote by **Pr** a set of Scott—Engeler programs constructed with the use of:

— a countable set of labels E,
— a set of formulas F,
— a set of substitutions S.

We shall define the relational system $\mathscr{S}$ and the realization of the language Pr of programs in $\mathscr{S}$.

Let us denote by B the following set of symbols: $\{p, (\ ,\ ), \alpha, \beta,\ ,\}$.

Let T be the smallest subset of B^* satisfying the conditions:

1) $\alpha, \beta \in T$,
2) if $\tau_1, \tau_2 \in T$ then $p(\tau_1, \tau_2) \in T$.

In other words, we can say that T is a set of terms constructed with the use of symbol p and constants α, β.

Let us define the function $\boldsymbol{P}$ over the set T (we consider T to be the domain of the system $\mathscr{S}$) as follows:

$$\boldsymbol{P}[\tau_1, \tau_2] = p(\tau_1, \tau_2).$$

We shall also define the zero-argument functions α, β and the identity relation id.

By the relational system S we mean the set T together with the functions α, β, $\boldsymbol{P}$ and the identity relation id;

$$S = \langle T, \alpha, \beta, \boldsymbol{P}, \mathrm{id}\rangle .$$

We define a realization R of the language of programs in a relational system S, as follows:

— the zero-argument functional symbols a, b represent the elements of α, β (respectively),
— the two-argument functional symbol P represents the function $\boldsymbol{P}$,
— the two-argument relational symbol—represents the identity relation.

Definition 1. For every term $\tau \in T$ a natural number defined recursively as follows:

1) terms α and β are of degree 0,
2) the degree of the term $p(\tau_1, \tau_2)$ is $\max(\deg \tau_1, \deg \tau_2)+1$ will be called the degree of the term τ.

Definition 2. T_s is a set of symmetrical terms, the smallest subset of T satisfying the conditions:

1) $\alpha, \beta \in T_s$,
2) if $\tau_1, \tau_2 \in T_s$, $\deg \tau_1 = \deg \tau_2$ then $p(\tau_1, \tau_2) \in T_s$.

Definition 3. A term τ' is a subterm of the rank 1 of a term τ if there is a term τ'' such that: $\tau = p(\tau', \tau'')$ or $\tau = p(\tau'', \tau')$.

We shall say that term τ' is a subterm of the rank $n-1$ of a term τ if τ' is a subterm of the rank 1 of some subterm of the rank n of the term τ.

Definition 4. A program $K \in \mathrm{Pr}$ is said to be simple if the formulas appearing in its instructions are of the form: $x=y$; $x, y \in V$ and the substitutions are of the form: $[x/a]$, $[x/b]$, $[x/y]$, $[x/p(y, z)]$, where $x, y, z \in V$ and $x \neq y, x \neq z$.

For any program K there exists an effectively contructible simple program K' such that:

$$K_R(v) = K'_R(v)$$

for any valuation v and any realization R. The obvious proof is left to the reader.

Lemma 1. *For any natural number n there is a symmetrical term τ_n, $\deg \tau_n \geqslant n$ with all subterms of the rank n distinct, such that:*

(w) *for any program K with n variables and for any valuation v_0 such that for $y \in V(K)$*

$\deg v_0(y) \leqslant \deg \tau_n - n$, *or*

$\deg v_0(y) \geqslant \deg \tau_n$ *and if* $\deg v_0(y) = \deg \tau_n$ *then* $v_0(y) \neq x_n$

the relation $K_R(v_0)(z) \neq \tau_n$, $z \in V(K)$ *holds.*

Moreover, if η is the term obtained from the term τ_n by applying the substitutions: ρ_1 for α, ρ_2 for β, where $\rho_1 \neq \rho_2$ and both the terms $\rho_1, \rho_2 \in T_s$ are of some degree $k \geqslant 0$ then condition (w) *is also fulfilled for η.*

Proof. For $n=1$ in the program K there is only one variable x_0. Consequently, it does not contain instructions other than:

k: if $x=x$ then go to p else go to q,

l: do s then go to p; where $s=[x/a]$, $[x/b]$.

Let $\tau_1 = p(\alpha, \beta)$. By the assumption $v_0(x) \neq \tau_1$. Hence any superposition (possibly the empty one) of substitutions of the type $[x/a]$, $[x/b]$ always results in a term different than τ_1. That proves the Lemma for $n=1$.

Let us assume the Lemma to be true for m and let $\tau = \tau_m$ be a term with property (w).

Let τ' be a term resulting from the term τ after substituting ρ_1 for α and ρ_2 for β and let τ'' be a term resulting from the term τ after substituting the term ρ_3 for α and ρ_4 for β (we assume here all the terms $\rho_1, \rho_2, \rho_3, \rho_4$ to be distinct and of some degree $k>0$ $\rho_i \in T_s$ for $i=1, 2, 3, 4$).

Let us denote by μ the term $p(\tau', \tau'')$. We shall prove that the term μ satisfies condition (w) for $n=m+1$ (of course $\mu \in T_s$ is the term of degree $\deg \tau + k + 1$ and all its subterms of degree $m+1$ are distinct).

Suppose that the term μ does not satisfy (w), i.e. there is a program K with at most $m+1$ variables and a valuation v_0 (satisfying for $y \in V(K)$

$\deg v_0(y) \leqslant \deg \mu - (m+1)$, or

$\deg v_0(y) \geqslant \deg \mu$ and if $\deg v_0(y) = \deg \mu$ then $v_0(y) \neq u$)

such that for some $x_0 \in V(K)$ the following relation holds:

$$K_R(v_0)(x_0) = \mu .$$

Let $(e_0, v_0), \ldots, (e_p, v_p)$ be the computation of program K for the valuation v_0. With each of the labels e_i, $i=0, 1, \ldots, p-1$ we shall associate a substitution s_i as follows:

if the instruction labeled by e_i is conditional, then $s_i = [\ ]$,

if this instruction is operational and has the form e_i: do s, then go to e_{i+1} then $s_i = s$. Of course $(s_0 \circ \ldots \circ s_{p-1})_R (v_0)(x_0) = \mu$.

With every valuation v_i and every variable $z \in V(K)$ we shall associate a valuation v_{i-1} and a set of variables (empty, one- or two-element) as follows:

if s_{i-1} is of the form $[z/a]$, $[z/b]$ then we associate $(v_{i-1}, \emptyset)$,

if s_{i-1} is of the form $[\ ]$, $[z/z]$ then we associate $(v_{i-1}, \{z\})$,

if s_{i-1} is of the form $[z/y]$ then we associate $(v_{i-1}, \{y\})$,

if s_{i-1} is of the form $[z/_{p(x,y)}]$ then we associate $(v_{i-1}, \{x, y\})$,

if s_{-1} is of the form $[x/\ ]$ where $x \neq z$ then we associate $(v_{i-1}, [\{z\})$.

Let us denote by F the above-constructed correspondence. Let i be the smallest number with the property: the set $F^i(v_p, x_0)$ has two elements, i.e. $F^i(v_p, x_0) =$

$=(v_r, \{z_1, z_2\})$. Then $v_r(z_1)=\tau'$, $v_r(z_2)=\tau''$ (and of course the number with the above property exists by the assumption that $v_0(y)\neq\mu$ for $y\in V(K)$ and all the subterms of the rank 1 of the term μ(i.e. τ', τ'') are distinct.

In general, once we have determined $v_{r_{i_1,\dots,i_{j-1}}}$ for $j<m+1$ and $z_{i_1,\dots,i_{j-1},t}$ $t=1,2$ then $(v_{r_{i_1,\dots,i_j}}, \{z_{i_1,\dots,i_j,1}\, z_{i_1,\dots,i_j,2}\})=F^{m_j}(v_{r_{i_1,\dots,i_{j-1}}}, z_{i_1,\dots,i_j})$ where m_j is the smallest number with the property that the set of variables $F^{m_j}(v_{r_{i_1,\dots,i_{j-1}}}, z_{i_1,\dots,i_j})$ has two elements.

Here $v_{r_{i_1,\dots,i_j}}(z_{i_1,\dots,i_j,t})$, $t=1,2$ are the subterms of the rank 1 of the term $v_{r_{i_1\dots,i_{j-1}}}(z_{i_1,\dots,i_j})$ and of course the number m_j always exists because $v_0(y)\neq$ $\neq v_{r_{i_1,\dots,i_{j-1}}}(z_{i_1,\dots,i_j})$ for any $y\in V(K)$ and all terms under consideration are distinct (deg $v_{r_{i_1,\dots,i_{j-1}}}(z_{i_1,\dots,i_j})$ equals deg $\mu-j+1$ and thus it is greater than deg $\mu-(m+1)$ and smaller than deg μ).

Let us denote by r_0 the smallest of the numbers $r_{i_1,\dots,i_m}$, $r_0=r_{i_1^0,\dots,i_m^0}$.

Clearly $(s_{r_0}\circ\dots\circ s_{p-1})_R(v_{r_0})(v_0)=\mu$. Let us denote by $s_{r_0}\dots s_{p-1}\,x_0$ the result of a successive application of the substitution operation for the variable x_0.

As it is known, $(s_{r_0}\circ,\dots,\circ s_{p-1})_R(v_{r_0})(x_0)=(s_{r_0}\dots s_{p-1}\,x_0)_R(v_{r_0})$.
The occurrence of y such that deg $v_r(y)>$ deg $\mu-(m+1)$ in the inscription $s_r\dots s_{p-1}\,x_0$ would imply that $v_{r_0}(y)$ is the subterm of the rank $<m+1$ of the term μ, contrarily to the definition of r_0.

Hence for the valuation v constructed for the given v_{r_0} as follows:

$$v(y)=\begin{cases}\alpha & \text{if } y\in V(K) \text{ and } \deg v_r\ (y)>\deg\mu-(m+1), \text{ and } \deg v_{r_0}(y)<\deg\mu,\\ v_{r_0}(y) & \text{otherwise}\end{cases}$$

the following relation holds:

$$(s_{r_0}\circ\dots\circ s_{p-1})_R(v_{r_0})(x_0)=(s_{r_0}\circ\dots\circ s_{p-1})_R(v)(x_0).$$

Furthermore, for every $j\,(s_{r_0}\circ\dots\circ s_{r_{i_1,\dots,i_{j-1}}})\,R(v_{r_0})(z_{i_1,\dots,i_j})=(s_{r_0}\circ\dots\circ s_{r_{i_1,\dots,i_{j-1}}})_R$ $(v)(z_{i_1,\dots,i_j})$.
The proof is by induction.

The terms $v_{r_{i_1^0,\dots,i_m^0}}(z_{i_1^0,\dots,i_m^0,t})=v_t$ $t=1,2$ are the subterms of the rank m of one of the terms τ', τ'', say τ'.

Let us notice that for every i satisfying $r_0<i\leq p$ there is a variable z_i such that the terms $v_j\,j=1,2$ are the subterms of $v_i(z_i)$ (this being a consequence of the definition of r_0).

We shall define the new sequence of substitutions $\bar{s}_{r_0}\dots,\bar{s}_{p-1}$ and the sequence of valuations $\bar{v}_{r_0},\dots,\bar{v}_p$ such that: $\bar{v}_{i-1}=\bar{s}_{i_R}(\bar{v}_i)$ and $\bar{v}_{r_0}=v$ and for every i, $r_0<i<p+1$ there is such a permutation π_i of the set of variables $V(K)$, that $(s_{r_0}\circ,\dots,\circ s_{i-1})_R(v)$ $(y)=\bar{v}_i(\pi_i(y))$ and for $z_0=z_{i_1^0,\dots,i_m^0}$ the terms v_1, v_2 are the subterms of $\bar{v}_i(z_0)$ for $r_0<i<p$. In particular $\bar{v}_p(\pi_p(x_0))=\mu$, $\bar{v}_r(\pi_r(z_2))=\tau''$.

The sequence $\bar{s}_i$ is defined as follows:

$$\bar{s}_{r_0}=s_{r_0},\qquad \pi_{r_0}=\text{id}.$$

Assume that for a given i we have already defined $\bar{s}_i$ and π_i. If s_{i+1} is of the form $[\ ]$, $[x/y]$, $[x/a]$, $[x/b]$, $[x/P(y,z)]$ where $x \neq z_0$, then s_{i+1} is of the form

$$[\ ],\ [\pi_i(x)/\pi_i(y)],\ [\pi_i(x)/a],\ [\pi_i(x)/b],\ [\pi_i(x)/P(\pi_i(y), \pi_i(z))]$$

and $\pi_{i+1} = \pi_i$.

If s_{i+1} is of the form

1) $[z_0/a]$, $[z_0/b]$,
2) $[z_0/y]$,
3) $[z_0/P(y,z)]$

then in the case 2) where v_1, v_2 are the subterms of $v_i(y)$ and also in the case 3) when v_1, v_2 are the subterms of either $v_i(y)$ or $v_i(z)$ we put $\bar{s}_{i+1}$ to be of the form

$$[\pi_i(z_0)/\pi_i(y)], \quad [\pi_i(z_0)/P(\pi_i(y), \pi_i(z))], \quad \pi_{i+1} = \pi_i.$$

In the remaining case we put $\bar{s}_{i+1}$ to be of the form

$$[z_0/z_{i+1}, z_{i+1}/a], \quad [z_0/z_{i+1}, z_{i+1}/b], \quad [z_0/z_{i-1}, z_{0+1}/\pi_i(y)],$$

$$[z_0/z_{i+1}, z_{i+1}/P(\pi_i(y), \pi_i(z))],$$

respectively, where z_{i+1} is such a variable that v_1, v_2 are subterms of $v_{i+1}(\pi_i^{-1}(z_{i+1}))$; of course in such a case $z_0 \neq z_{i+1}$. Then also $\pi_{i+1} = f\pi_i$, where:

$$f(y) = \begin{cases} z_0 & \text{for } y = z_{i+1}, \\ z_{i+1} & \text{for } y = z_0, \\ y & \text{otherwise.} \end{cases}$$

One can easily verify that the above-constructed sequence has the required properties.

As we have already noticed, $(\bar{s}_{r_0} \circ \ldots \circ \bar{s}_r)_R(v)(\pi_r(z_2)) = \tau''$.

The terms v_1 and v_2 are not subterms of the term τ'' (all the subterms of the rank $m+1$ of the term μ are distinct) and therefore the expression $\bar{s}_{r_0} \ldots \bar{s}_r \bar{z}_2$ cannot contain z_0 (here we denote $\pi_r(z_2)$ by $\bar{z}_2$).

Let us create the sequence $\bar{s}_{k_1}, \ldots, \bar{s}_{k_n}$ from the sequence $\bar{s}_{r_0}, \ldots, \bar{s}_r$ sequence all the substitutions containing the variable z_0.

In virtue of the above

$$(\bar{s}_{r_0} \circ \ldots \circ \bar{s}_r)_R(v)(\bar{z}_2) = (\bar{s}_{k_1} \circ \ldots \circ \bar{s}_{k_n})_R(v)(\bar{z}_2) = \tau''.$$

That in turn implies the existence of a program K' (the sequence of substitutions $\bar{s}_{k_1}, \ldots, \bar{s}_{k_n}$) with at most m variables (we have excluded z_0) and such that for the valuation v (satisfying:

$\deg v(y) \leqslant \deg \tau'' - m$, or

$\deg v(y) \geqslant \deg \tau''$ and if $\deg v(y) = \deg \tau''$ then $\tau'' \neq v(y)$)

it fulfils the condition $K'_R(v)(\bar{z}_2) = \tau''$.

This conclusion together with the fact that τ'' is a substitution of τ contradicts the induction hypothesis that the Lemma is true for m.

That completes the proof of the Lemma.

Definition 5. A program K computes the function $F: T \to T$ if for some fixed $x_0 \in V(K)$ $v(x_0)=\tau$ implies $K_R(v)(x_0)=F(\tau)$.

Let $F: T \to T$ be the least of the functions satisfying the system of equations

$$\begin{cases} f(a)=b, \\ f(b)=a, \\ f(P(x,y))=P(f(x),f(y)). \end{cases}$$

One can easily notice that F is the function which in any term substitutes α for β and vice versa. Therefore the entire set T constitutes the set of values of F.

Let us suppose that there is a program K which computes the function F, and let n be the number of variables in such a program.

Let τ denote the term $F^{-1}(\tau_n)$ and

$$v(y)=\begin{cases} \alpha & \text{for } y \neq x_0, \\ \tau & \text{for } y = x_0. \end{cases}$$

Then the fact that K computes the function F (i.e. the term τ_n is the value of $K_R(v)(x_0)$) contradicts the preceding Lemma.

As an immediate corollary we have:

Theorem 1. *There is a relational system $\mathcal{S}=\langle T, a, b, P, =\rangle$ with the property: the class of functions defined as the least solution of the system of equations contains functions which do not belong to the class of functions computable by means of Scott-Engeler programs.*

Let N denote the set of natural numbers and T the previously considered set of terms.

Let $D=N \cup \{\varepsilon\} \cup T$, where $\varepsilon \notin T \cup N$.

In the set D we shall define the zero-argument functions $\mathbf{0}, \mathbf{1}, a, b$. Symbol $\mathbf{0}$ denotes the element $0 \in N$, $\mathbf{1}$ — the element $1 \in N$, a — the element $\alpha \in T$, b — the element $\beta \in T$.

One-argument functions R, L are defined as follows

$$R(x)=\begin{cases} \varepsilon & \text{for } x \in T \cup \{\varepsilon\}, \\ r(x) & \text{for } x \in N, \end{cases}$$

where r denotes the right-side inverse function for the function of the pair of functions,

$$L(x)=\begin{cases} \varepsilon & \text{for } x \in T \cup \{\varepsilon\}, \\ l(x) & \text{for } x \in N, \end{cases}$$

where l denotes the left side inverse function for the function of the pair of functions. We define the two-argument function P as follows:

$$P(x,y)=\begin{cases} x+y & \text{for } x, y \in N; \ +\text{denotes the addition in } N, \\ P(x,y) & \text{for } x, y \in T, \\ \varepsilon & \text{otherwise.} \end{cases}$$

Thus we have defined the system

$$D = \langle D, 0, 1, \alpha, \beta, R, L, P, = \rangle.$$

We assume here the set of programs over an alphabet A_1 and the set of labels E to be defined in a normal way, where:

$$A_1 = V \cup \{\mathbf{0}, \mathbf{1}, a, b\} \cup \{R, L\} \cup \{P\} \cup \{=\}.$$

We define the realization R_1 by associating with every functional (or relational) symbol an identically denoted function (relation).

In an analogous manner one can prove

LEMMA 2. *For every natural n there is a term $\tau_n \in T_s$ of the degree $\geqslant n$ with all subterms of the rank n distinct such that:*

(w) *for every program K with n variables and for any valuation v_0 satisfying: if $v_0(y) \in T$ $(y \in V(K))$ then* $\deg v_0(y) \leqslant \deg \tau_n - n$, *the following relation holds*:

$$K_{R_1}(v_0)(x) \neq \tau_n \quad \text{for } x \in V(K).$$

Let F be the least function $F: D \to D$ satisfying the system of equations

$$\begin{cases} f(\mathbf{0}) = a, \\ f(\mathbf{1}) = b, \\ f(x) = P\big(f(L(x)), f(R(x))\big). \end{cases}$$

We easily notice that F has the domain N and a set of values T.

On the ground of assumptions analogous as in the preceding case we prove that there is no program computing the function F.

We can write the above system of equations in the form:

$$f(x) = \text{ if } x = 0 \text{ then } a \text{ else if } x = 1 \text{ then } b \text{ else } P\big(f(L(x)), f(R(x))\big)$$

identical with the definition of a procedure. Therefore we have:

THEOREM 2. *There is a relational system with the property: the class of functions computable by means of programs with procedures is essentially larger than the class of functions computable by means of programs without procedures.*

Notice that the system under consideration included as a subsystem the set of natural numbers with $0, 1, +, r, l$.

This is what is called a constructive subsystem.

COROLLARY 1. *The fact that the system contains a constructive subsystem does not imply that the class of functions computable by means of programs with procedures is identical with the class of functions computable by means of programs without procedures.*

INSTITUT MATEMATYCZNY, UNIWERSYTET WARSZAWSKI, 00-901 WARSZAWA, PKiN

REFERENCES

[1] H. Rasiowa, R. Sikowski, *The mathematics of metamathematics*, PWN, Warszawa 1963.

[2] A. Salwicki, *On the equivalence of FS-expressions and programs*, Bull. Acad. Polon. Sci., Sér. Sci. Math. Astronom. Phys., **18** (1970), 275—278.

[3] — , *Formalized algorithmic languages*, ibid., 227—233.

В. Данько, **Непрограммированные функции определяемые процедурой**

Содержание. В настоящей работе приводятся примеры реляционных систем, для которых класс функций вычисляемых программами с процедурами существенно шире класса функций вычислоемых программами без процедур.