Information Processing Letters 25 (1987) 311-315
North-Holland
10 July 1987

SUPER-EXPONENTIALS NONPRIMITIVE RECURSIVE, BUT RUDIMENTARY

Cristian CALUDE

Department of Mathematics, University of Bucharest, Str. Academiei 14, R-70109 Bucharest, Romania

Communicated by A.P. Ershov
Received 3 December 1986

We prove that every super-exponential function of Ackermann type (i.e., nonprimitive recursive) in the sense of Porto and Matos (1980) is Smullyan rudimentary (Smullyan, 1961).

Keywords: Ackermann function, nonprimitive recursive function, rudimentary function, loop-program

1. Preliminaries

Let $\mathbb{N} = \{0, 1, \dots\}$ be the set of natural numbers. All functions in this paper are of the form $f : \mathbb{N}^k \to \mathbb{N}$, $k > 0$.

Ackermann–Peter's function $A : \mathbb{N}^2 \to \mathbb{N}$ is defined by the equations (see [1,13]):

$$A(0, x) = x + 1, \tag{1}$$

$$A(n+1, 0) = A(n, 1), \tag{2}$$

$$A(n+1, x+1) = A(n, A(n+1, x)). \tag{3}$$

It is well known that A is recursive and nonprimitive recursive. Skolem [18] showed that A has a primitive recursive graph; in [17,20] it is proved that A has a context-sensitive graph, whereas in [5] it is proved that the graph of A is elementary in Kalmar's sense.

We assume the reader to be familiar with the monotonicity properties of A; in particular, (i) A is strictly increasing in each argument, (ii) the value of A is greater than any of its arguments, and (iii) the first argument has a greater influence on the value of A than its second argument [13]. The last property can more exactly be stated as follows. For each natural n, denote by $\exp_n$ (respectively EXP_n) the unary functions $\exp_n(x) = A(n, x)$ (respectively $EXP_n(x) = A(x, n)$). Each $\exp_n$ has a super-exponential growth for $n > 3$, and so has each EXP_n, for $n \geqslant 0$; every $\exp_n$ is primitive recursive, but no EXP_n has this property.

A binary function satisfying (3) for all $n \geqslant n_0$ and $x \geqslant x_0$ (n_0 and x_0 are fixed naturals) is called a super-exponential of Ackermann's type (see [14]).

A predicate is called *rudimentary* (or *bounded arithmetic*) if it is obtainable from the predicates $z = x + y$ and $z = x \cdot y$, by a finite number of applications of the Boolean operations, finite quantifications (both existential and universal) and explicit transformations [19]. The Boolean operations will be denoted by $\neg$, $\vee$, &; relations $\rightarrow$ and $\leftrightarrow$ respectively denote implication and equivalence. A finite quantification is of the form $(\forall x < y)$ or $(\exists x < y)$, where y is some free variable of the formula. Explicit transformation is a generalized form of substitution which includes composition, identification, permutation of variables, etc. [19] (see [4,9,10,11] for various characterizations of rudimentary predicates). Furthermore, it is well known that the rudimentary predicates constitute a subclass of Grzegorczyk's $\mathscr{E}^0_*$ class [8]. A function $f : \mathbb{N}^k \to \mathbb{N}$ is called *rudimentary* in Smullyan sense [19] if the predicate $z = f(x_1, \dots, x_k)$ is rudimentary.

Gödel [7] proved that the exponential x^y is arithmetic. Bennett [4] strengthened this result by showing that x^y is rudimentary (see also [10,15]). Finkelstein [6] proved that each primitive recursive super-exponential $\exp_n$ is rudimentary.

2. Main result

Our aim is to show that the Ackermann–Peter function is rudimentary. We shall use the injective function $OP : \mathbb{N}^2 \to \mathbb{N}$ given by $OP(y, z) = (y+z)^2 + y + 1$, and Gödel's β function [7]. For completeness we give some details here (see [16]). The function $\beta : \mathbb{N}^2 \to \mathbb{N}$ is defined by the formula

$$\beta(a, i) = (\mu x < a \dot{-} 1)(\exists y < a)(\exists z < a)(a = OP(y, z) \;\&\; \mathrm{Div}(y, 1 + (1 + OP(x, i))z)),$$

where Div(y, x) means "x divides y", $x \dot{-} y = \max\{0, x - y\}$ is the arithmetical difference and μ refers to the (bounded) minimization operator. Given a sequence $a_0, a_1, \ldots, a_{n-1}$ of natural numbers and an injective rudimentary unary function f, there exists a natural t such that, for every $i < n$, $\beta(t, f(i)) = a_i$, and, for every j,

$$\beta(t, j) < t \dot{-} 1 \;\leftrightarrow\; (\exists i < n)(j = f(i)).$$

Such a t can be obtained as follows: take

$$c = \max\{1 + OP(a_i, f(i))\}, \qquad r = c!, \qquad y = \Pi((1 + (1 + OP(a_i, f(i)))r)\,; i = 0, \ldots, n-1),$$

and finally put $t = OP(y, r)$. All of the above manipulations are rudimentary.

Equations (1)–(3) will be used in the *rightmost way*, i.e., they are always applied at the innermost level of the nest. We continue by making estimations concerning the growth of some functions involved in the rightmost computation of A(n, x).

First, in view of the inequality $A(i, j) \geqslant A(i-1, j+1)$, one has

$$i + j \geqslant A(n, x) \;\rightarrow\; A(i, j) > A(n, x). \tag{4}$$

Second, for every pair (n, x) we define the finite set

$$\mathit{useful}(n, x) = \{(n, x)\} \cup \{(i, j) \in \mathbb{N}^2 ; A(i, j) \textit{ is} \text{ used in the rightmost computation of } A(n, x)\}$$

and the binary function length(n, x) = card(*useful*(n, x)). Clearly, if (i, j) is in *useful*(n, x), then $A(i, j) \leqslant A(n, x)$. So, by (4),

$$\mathrm{length}(n, x) \leqslant (A(n, x))^2 - 1. \tag{5}$$

Third, we need two more estimations, namely, for every $y > 0$,

$$b(x) = x^{x^x} < A(4, x), \tag{6}$$

and, for every $n > 4$,

$$A(n, x) = A(4, y) \;\rightarrow\; y \geqslant 2n - 1. \tag{7}$$

Given a pair (n, x) we encode the rightmost computation of the value $z = A(n, x)$ by the sequence

$$(A(i_1, j_1), OP(i_1, j_1)), \ldots, (A(i_k, j_k), OP(i_k, j_k)), \tag{8}$$

where

$$k = \text{length}(n, x), \qquad \textit{useful}(n, x) = \{(i_1, j_1), \ldots, (i_k, j_k)\},$$
$$A(i_1, j_1) \leqslant A(i_2, j_2) \leqslant \cdots \leqslant A(i_k, j_k),$$

and if $A(i_u, j_u) = A(i_{u+1}, j_{u+1})$, then $OP(i_u, j_u) < OP(i_{u+1}, j_{u+1})$. Finally, sequence (8) can be coded by a natural t (depending upon n and x) by means of Gödel's β function. More precisely, t satisfies the relations

$$A(i, j) = \beta(t, OP(i, j)) \quad \text{for every } (i, j) \in \textit{useful}(n, x), \tag{9}$$
$$(\forall w)(\beta(w, t) < t \dot{-} 1) \;\leftrightarrow\; (\exists (i, j) \in \textit{useful}(n, x))(w = OP(i, j)). \tag{10}$$

Now let $f: \mathbb{N}^3 \to \mathbb{N}$ be the rudimentary function given by $f(n, x, y) = \beta(y, OP(n, x))$ and consider the ternary predicates P and R defined as follows:

$$\begin{aligned} P(n, x, y) \leftrightarrow\ & (f(n, x, y) < y \dot{-} 1) \& \{[(n = 0) \& (f(n, x, y) = x + 1)] \\ & \vee [(n > 0) \& (x = 0) \& (f(n \dot{-} 1, x + 1, y) < y \dot{-} 1) \& (f(n, x, y) = f(n \dot{-} 1, x + 1, y))] \\ & \vee [(nx > 0) \& (f(n, x \dot{-} 1, y) < y \dot{-} 1) \& (f(n \dot{-} 1, f(n, x \dot{-} 1, y), y) < y \dot{-} 1) \\ & \quad \& (f(n, x, y) = f(n \dot{-} 1, f(n, x \dot{-} 1, y), y))]\}, \end{aligned}$$

$$\begin{aligned} R(n, x, y) \leftrightarrow\ & (f(n, x, y) < y \dot{-} 1) \& (\forall u < y)(\forall i < u)(\forall j < u) \\ & [(u = OP(i, j)) \& (f(i, j, y) < y \dot{-} 1) \to P(i, j, y)]. \end{aligned}$$

Using equations (1)–(3) one can easily prove, by induction on i and j, the following statement: for all n, x, y,

$$R(n, x, y) \& (u = OP(i, j)) \& (f(i, j, y) < y \dot{-} 1) \;\to\; A(i, j) = f(i, j, y). \tag{11}$$

Finally, notice that for every pair (n, x) the least natural t satisfying predicate R(n, x, t) *does exist* and, in view of (9) and (10),

$$z = A(n, x) \;\leftrightarrow\; (\exists t)(z = f(n, x, t) \& R(n, x, t)). \tag{12}$$

In what follows we shall restrict ourselves to $n > 4$ and $x > 0$ because it is well known that the $\exp_n$ $(n \leqslant 4)$ are rudimentary. Using (3) one gets

$$\begin{aligned} z = A(n, x) &= A(n - 1, A(n, x - 1)) = A(n - 1, z_n) \\ &= A(n - 2, A(n - 1, z_n - 1)) = A(n - 2, z_{n-1}) \\ &\ \vdots \\ &= A(4, A(5, z_6 - 1)) = A(4, z_5), \end{aligned}$$

where

$$z_n = A(n, x - 1), \tag{13}$$
$$z_i = A(i, z_{i+1} - 1), \quad 5 \leqslant i \leqslant n - 1. \tag{14}$$

On the basis of (13) and (14) we may write the equivalence

$$\begin{aligned} z = A(n, x) \;\leftrightarrow\; & (\exists z_n) \ldots (\exists z_5)((z_n = A(n, x - 1)) \& \cdots \& (z_i = A(i, z_{i+1} - 1)) \\ & \& \cdots \& (z_5 = A(5, z_6 - 1)) \& (z = A(4, z_5))). \end{aligned} \tag{15}$$

The next step consists in the estimation of the domain of variation of the variables $t, z_n, \ldots, z_5$ occurring in (12) and (15). We use the notations established in the paragraph that describes Gödel's β function. An elementary analysis gives

$$\begin{aligned} c &= \max\{1 + OP(A(i, j), OP(i, j)) ; (i, j) \in \mathit{useful}(n, x)\} \\ &\leqslant 1 + OP(A(n, x), A(n, x)) = 1 + OP(z, z) \leqslant z^3, \\ r &= c! \leqslant z^{3z^3}, \\ y &= \Pi((1 + (1 + OP(A(i, j), OP(i, j)))r) ; (i, j) \in \mathit{useful}(n, x)) \\ &\leqslant (1 + (1 + OP(z, z))r)^{\text{length}(n,x)+1} \leqslant (1 + (1 + OP(z, z))r)^{z^2} \leqslant 2z^{4z^5} \end{aligned}$$

(we have used (5)), and

$$t = OP(y, r) \leqslant 5y^2 \leqslant 20z^{8z^5} \leqslant z^{z^z} = b(z),$$

because $z \geqslant 20$. Accordingly, formula (12) can be written as follows: for $n > 4$ and $x > 0$,

$$z = A(n, x) \quad\leftrightarrow\quad (\exists t \leqslant b(z))[(z = f(n, x, t))\&R(n, x, t)]. \tag{16}$$

Furthermore, for $5 \leqslant i \leqslant n - 1$, by (6) one gets

$$z_i = A(i, z_{i+1} - 1) \geqslant A(i - 1, z_{i+1}) \geqslant A(4, z_{i+1}) > b(z_{i+1}). \tag{17}$$

We conclude our proof as follows. Start with (15) and replace each relation (13) or (14) by the corresponding right-hand side given by (16); by (17) one gets

$$\begin{aligned} z=A(n, x) \quad\leftrightarrow\quad & (\exists z_n)\ldots(\exists z_5)(\exists y_n < b(z_n))\ldots(\exists y_5 < b(z_5)) \\ & [(z_n = f(n, x \dot{-} 1, y_n))\&R(n, x \dot{-} 1, y_n)\& \cdots \\ & \&(z_5 = f(5, z_6 \dot{-} 1, y_5))\&R(5, z_6 \dot{-} 1, y_5)\&(z = A(4, z_5))]. \end{aligned} \tag{18}$$

Now consider the sequence (a_i), $0 \leqslant i \leqslant 2n - 9$, defined by $a_i = y_{n-i}$ if $0 \leqslant i \leqslant n - 5$, $a_i = z_{2n-i-4}$ if $n - 5 < i \leqslant 2n - 9$, and the identity function. By means of Gödel's β function one obtains a code w such that $\beta(w, i) = a_i$, for all $0 \leqslant i \leqslant 2n - 9$. Furthermore, a similar estimation as in the case of t works:

$$c = \max\{1 + OP(a_i, i) ; 0 \leqslant i \leqslant 2n - 9\} = 1 + OP(z_5, 2n - 9) \leqslant 1 + OP(z_5, z_5) \leqslant z_5^3,$$

because, from (7), $z = A(n, x) = A(4, z_5) \to z_5 \geqslant 2n - 1$, $r \leqslant z_5^{3z_5^3}$, $y \leqslant z_5^{2z_5^5}$, $w = OP(y, r) \leqslant 5y^2 \leqslant 5z_5^{z_5^6} \leqslant b(z_5) < A(4, z_5) = z$, by (6). Accordingly, the whole of the right-hand side of equivalence relation (18) can be bounded by z, i.e.,

$$\begin{aligned} z=A(n, x) \quad\leftrightarrow\quad (\exists w < z)[(\beta(w, n \dot{-} 4) = f(n, x \dot{-} 1, \beta(w, 0)))\&R(n, x \dot{-} 1, \beta(w, 0)) \\ \& \cdots \&(\beta(w, 2n - 9) = f(5, \beta(w, 2n \dot{-} 10) - 1, \beta(w, n \dot{-} 5))) \\ \&R(5, \beta(w, 2n \dot{-} 10) - 1, \beta(w, n \dot{-} 5))\&(z = A(4, \beta(w, 2n \dot{-} 9)))], \end{aligned}$$

which shows that A is rudimentary.

3. Concluding remarks

(a) The binary function $\text{step}(n, x) = (\mu t)Q(n, x, t)$, where $Q(n, x, t) \leftrightarrow (z = f(n, x, t))\&R(n, x, t)$ is rudimentary and nonprimitive recursive. Indeed,

$$t = \text{step}(n, x) \quad\leftrightarrow\quad Q(n, x, t)\&((\forall u < t)\neg Q(n, x, u)) \quad\text{and}\quad A(n, x) = f(n, x, \text{step}(n, x))$$

(see (12)).

(b) Each nonprimitive recursive function EXP_n is rudimentary. Furthermore, *our proof* shows that every super-exponential of Ackermann type is rudimentary.

(c) In [2], modified loop-programs are defined as follows. A *loop*(0) program is a finite sequence of the following six assignments. $t \leftarrow 0$, $t \leftarrow 1$, $t \leftarrow t \dot{-} 1$, $t \leftarrow sg(s)$, $t \leftarrow t+1$, $t \leftarrow s$ (the last two statements are called principal; $sg(0)=0$, $sg(x+1)=1$). A *loop*(n + 1) program is recursively defined as: (i) a *loop*(n) program, or (ii) of the form LOOP t, P, END, where P is a *loop*(n) program, or (iii) a concatenation of two *loop*(n + 1) programs, subject to the restriction: if a loop-program contains a subprogram Q : LOOP t, P, END, where P contains only assignment statements, then t does not appear as the left-hand side of any principal statement in P or in the scope of any loop directly containing Q. The meaning of these programs should be clear. We stress that, in the case of *deterministic* use of the loop-programs, all the working variables and the output variable have entry zero; in the case of *nondeterministic* use, they may have arbitrary values. The run-time of a program is the number of steps executed until termination. Now, in view of the fact that each rudimentary predicate is in Grzegorczyk's $\mathscr{E}_*^0$ class and $\mathscr{E}_2$ is the set of polynomially computable functions in deterministic modified loop-programs [2, Theorem 1], it follows that the function A can be computed nondeterministically in polynomial time by the modified loop-programs. Notice that the above 'polynomial computation' is not the well-known Turing polynomial computation (see [3,11]). It is an *open* problem (formulated in [12]) to show that A does not have a polynomial-time nondeterministic Turing computable graph.

References

[1] W. Ackermann, Zum Hilbertschen Aufbau der reellen Zahlen, Math. Ann. 99 (1928) 118–133.

[2] A. Amir and Y. Choueka, Loop-programs and polynomially computable functions, Internat. J. Comput. Math. 9 (1981) 195–205.

[3] A. Amir and Y. Choueka, A syntactical definition of the P =? NP problem, Internat, J. Comput. Math. 17 (1985) 217–228.

[4] J.H. Bennett, On spectra, Ph.D. Dissertation, Princeton Univ., 1962.

[5] Ş. Buzeţeanu and C. Calude, Functions having the graph in the *n*th Grzegorczyk class, Found. Control Engrg. 11 (1986) 61–67.

[6] S. Finkelstein, On rudimentary languages, Unpublished manuscript, 1977 (quoted following [9]).

[7] K. Gödel, Über formal unentscheidbare Sätze der Principia Mathematica und verwandte Systeme, Teil I, Monatshefte f. Math. & Physik 38 (1931) 173–198 [English translation in: M. Davis, ed., The Undecidable (Raven Press, New York, 1965) 5–38].

[8] A. Grzegorczyk, Some classes of recursive functions, Rozprawy Math. 4 (1953) 1–45.

[9] K. Harrow, Equivalence of some hierarchies of primitive recursive functions, Z. f. Math. Logik & Grundlag. Math. 25 (1979) 411–418.

[10] V.A. Nepomnjaščiĭ, Rudimentary interpretations of two-tape Turing computations, Kybernetika 2 (1970) 29–35 (in Russian).

[11] V.A. Nepomnjaščiĭ, Rudimentary predicates and Turing computations, Dokl. Akad. Nauk SSSR 195 (1970) 1462–1465.

[12] V.A. Nepomnjaščiĭ, Personal communication, September 1986.

[13] R. Peter, Rekursive Funktionen (Akadémiai Kiadó, Budapest, 1957).

[14] A.G. Porto and A.B. Matos, Ackermann and the superpowers, ACM SIGACT News 12 (3) (1980) 90–95.

[15] P. Pudlak, A definition of exponentiation by a bounded arithmetical formula, Comm. Math. Univ. Carolin. 24 (1983) 667–671.

[16] J.R. Shoenfield, Mathematical Logic (Addison-Wesley, Reading, MA, 1967).

[17] D. Simovici, Context-sensitive languages and Ackermann's function, Found. Control Engrg. 5 (1980) 91–103.

[18] T. Skolem, Remarks on recursive functions and relations, Norske Vid. Selsk. Forh. 17 (1944) 89–92; also in: J.E. Fenstad, ed., Selected Works in Logic (Universitetsforlaget, Oslo, 1970) 483–486.

[19] R.M. Smullyan, Theory of Formal Systems (Princeton University Press, Princeton, NJ, 1961).

[20] M. Tătărâm, Ackermann–Peter's function has primitive recursive graph and range, Found. Control Engrg. 2 (1984) 177–180.