A Hierarchy of Unary Primitive Recursive String-functions

Lila Santean*
Institute for Informatics
8-10 Miciurin blvd.
71316 Bucharest 1, Romania

Abstract

Using a recent result of G.Asser, an extention of Ackermann-Peter hierarchy of unary primitive recursive functions to string-functions is obtained. The resulting hierarchy classifies the string-functions according to their lexicographical growth.

1 Introduction

Let $\mathbf{N}$ be the set of naturals i.e. $\mathbf{N} = \{0, 1, 2, \ldots\}$. Consider a fixed alphabet $A = \{a_1, a_2, \ldots, a_r\}, r \geq 2$ 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 $a_1 < a_2 < \ldots < a_r$. 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:

$$\begin{aligned} I(x) &= x, \\ Succ(x) &= x+1, \\ C_m(x) &= m, \\ Pd(x) &= x \dot{-} 1, \text{ where } x \dot{-} y = max\{x-y, 0\}, \\ &\quad \text{for all } x, m, y \in \mathbf{N}. \end{aligned}$$

By $I^A, Succ_i^A, C_u^A, \sigma, \pi$, we denote the following string-functions:

$$I^A(w) = w,$$

*Present address: Department of Mathematics, University of Turku, 20500 Turku, Finland

$$
\begin{aligned}
Succ_i^A(w) &= wa_i(1 \leq i \leq r), \\
C_u^A(w) &= u, \\
\sigma(e) &= a_1, \sigma(wa_i) = wa_{i+1} \text{ if } 1 \leq i < r \text{ and } \sigma(wa_r) = \sigma(w)a_1 \\
\pi(e) &= e, \pi(\sigma(w)) = w, \\
& \quad \text{for all } w, u \in A^*
\end{aligned}
$$

Furtheron one uses the primitive recursive bijections $c : A^* \longrightarrow \mathbf{N}, \overline{c} : \mathbf{N} \longrightarrow A^*$ given by

$$
\begin{aligned}
c(e) &= 0, c(wa_i) = r \cdot c(w) + i, 1 \leq i \leq r, w \in A^*, \\
\overline{c}(0) &= e, \overline{c}(m+1) = \sigma(\overline{c}(m)), m \in \mathbf{N}.
\end{aligned}
$$

To each f in Fnc one associates the string-function $s(f) \in Fnc_A$ defined by $s(f)(w) = \overline{c}(f(c(w)))$ and for each g in Fnc_A one associates the number-theoretical function $n(g)$ defined by $n(g)(x) = c(g(\overline{c}(x)))$. 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, $s(Succ) = \sigma, n(I^A) = I, s(Pd) = \pi$. 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 :

$$
\begin{aligned}
sub(f,g) &= h \Longleftrightarrow f,g,h \in Fnc, f(g(x)) = h(x); \\
\mathit{diff}(f,g) &= h \Longleftrightarrow f,g,h \in Fnc, h(x) = f(x) \dot{-} g(x); \\
it_x(f) &= h \Longleftrightarrow f,h \in Fnc, h(0) = x, h(y+1) = f(h(y)); \\
sub_A(f,g) &= h \Longleftrightarrow f,g,h \in Fnc_A, f(g(w)) = h(w); \\
\sigma - it_{A,w}(f) &= h \Longleftrightarrow f,h \in Fnc_A, h(e) = w, h(\sigma(u)) = f(h(u)).
\end{aligned}
$$

For every operator φ in Fnc, $s(\varphi)(f) = s(\varphi(n(f)))$, for every $f \in Fnc$; analogously, for every operator θ in $Fnc_A, n(\theta)(g) = n(\theta(s(g)))$, for every $g \in Fnc$. For example, $s(it_x) = \sigma - it_{A,c(x)}, n(\sigma - it_{A,w}) = it_{\overline{c}(w)}$.

2 Ackermann-Peter string-function

The primitive-recursive functions were introduced by Asser [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 *Weichrauch* [8]. The function, denoted by $A : A^* \longrightarrow A^*$, is given by means of the following three equations :

$$
\begin{aligned}
A_0(x) &= \sigma(x) && (1) \\
A_{n+1}(e) &= A_n(a_1) && (2) \\
A_{n+1}(\sigma(x)) &= A_n(A_{n+1}(x)). && (3)
\end{aligned}
$$

The following technical results concern the monotonicity properties of the function A; they generalize the monotonicity properties of the number-theoretical Ackermann-Peter function (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 $x = e, A_{n+1}(e) = A_n(a_1) > e$. Suppose now that $A_{n+1}(x) > x$. We use (3) 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, that is $A_{n+1}(x) \geq \sigma(x)$, we obtain $A_{n+1}(\sigma(x)) > \sigma(x)$. $\square$

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 (3) and lemma 1 we have

$$A_{n+1}(\sigma(x)) = A_n(A_{n+1}(x)) > A_{n+1}(x).$$

$\square$

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

Lemma 3 *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 (2) 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 (3) and corollary 2). $\square$

Corollary *For all naturals n and m, and for all strings x in A^*, if $n < m$, then*

$$A_n(x) < A_m(x).$$

Lemma 4 *For all strings x of A^* we have: $A_2(x) = \sigma^{2c(x)+3}(e)$.*

Proof. We proceed by induction on x.

For $x = e$, in view of (2) we have

$$A_2(e) = A_1(a_1) = \sigma(\sigma(a_1)) = \sigma^3(e) = \sigma^{2c(e)+3}(e).$$

Assuming that $A_2(x) = \sigma^{2c(x)+3}(e)$, we prove that $A_2(\sigma(x)) = \sigma^{2c(\sigma(x))+3}(e)$. Indeed, using (3) and the equality $c(\sigma(x)) = c(x) + 1$, we get:

$$A_2(\sigma(x)) = A_1(A_2(x)) = A_1(\sigma^{2c(x)+3}(e)) = \sigma^{2c(x)+5}(e) = \sigma^{2c(\sigma(x))+3}(e).$$

□

Lemma 5 *For all naturals k and $n \geq 1$, there exists a natural i (which depends upon k) such that*

$$A_n(\sigma^k(x)) < A_{n+1}(\pi^k(x)),$$

for every string x in A^* with $c(x) > i$.

Proof. We first notice that for every string x with $c(x) > 3k \dot{-} 1$, we have $\sigma^k(x) < A_2(\pi^{k+1}(x))$.

Indeed, by lemma 4 we have

$$\begin{aligned} A_2(\pi^{k+1}(x)) &= \sigma^{2c(\pi^{k+1}(x))+3}(e) = \sigma^{2(c(x) \dot{-} k \dot{-} 1)+3}(e) = \sigma^{2c(x) \dot{-} 2k+1}(e) \\ &> \sigma^{k+c(x)}(e) = \sigma^k(\sigma^{c(x)}(e)) = \sigma^k(x). \end{aligned}$$

Consequently, using corolary 2 and corollary 2,

$$A_n(\sigma^k(x)) < A_n(A_2(\pi^{k+1}(x))) < A_n(A_{n+1}(\pi^{k+1}(x))) = A_{n+1}(\pi^k(x)),$$

for all strings x with $c(x) > 3k \dot{-} 1$. In conclusion, we can take $i = 3k \dot{-} 1$. □

Lemma 6 *For all naturals n and strings x in A^* we have*

$$A_{n+1}(x) = A_n^{c(x)+1}(a_1).$$

Proof. We proceed by induction on x.

For $x = e$, using (2) we obtain

$$A_{n+1}(e) = A_n(a_1) = A_n^{c(e)+1}(a_1).$$

Assuming that $A_{n+1}(x) = A_n^{c(x)+1}(a_1)$ we prove the equality

$$A_{n+1}(\sigma(x)) = A_n^{c(\sigma(x))+1}(a_1).$$

Indeed, using (3) 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 1 We say that the function $f : A^* \longrightarrow A^*$ is defined by *limited iteration at* e (shortly, *limited iteration*) from the functions $g : A^* \longrightarrow A^*$ and $h : A^* \longrightarrow A^*$ if it satisfies the following equations:

$$\begin{aligned} f(e) &= e, \\ f(\sigma(x)) &= g(f(x)), \\ f(x) &\leq h(x), \end{aligned}$$

for every x in A^*.

Definition 2 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 $s(\mathit{diff})$ (the string-function operation corresponding to the arithmetical difference).

Lemma 7 *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 $\overline{sg}$ defined by:*

$$\begin{aligned} l_i(w) &= a_i, 1 \leq i \leq r, \\ sg(w) &= \begin{cases} e & \text{if } w = e \\ a_1 & \text{if } w \neq e \end{cases} \\ \overline{sg}(w) &= \begin{cases} a_1 & \text{if } w = e \\ e & \text{if } w \neq e, \end{cases} \\ & \quad \text{for all } w \in A^*. \end{aligned}$$

Proof. It follows from the following equalities:

$$\begin{aligned} C_e^A &= s(\mathit{diff})(A_0, A_0) \\ l_i &= A_0^i(e), 1 \leq i \leq r \\ I^A &= s(\mathit{diff})(A_0, l_1) \\ \overline{sg} &= s(\mathit{diff})(l_1, I^A) \\ sg &= s(\mathit{diff})(l_1, \overline{sg}) \\ \pi &= s(\mathit{diff})(I^A, l_1) \end{aligned}$$

and from the definition 2. □

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

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

If $n = 0$, by definition 2, $A_0 \in C_m$, for every natural m. Assume that $A_n \in C_{n+k}, \forall k \in \mathbf{N}$. We shall prove that $A_{n+1} \in C_{n+k+1}, \forall k \in \mathbf{N}$.

Assertion: For every string $x, A_{n+1}(x) = f(\sigma(x))$, where

$$\begin{aligned} f(e) &= e, \\ f(\sigma(x)) &= A_n(g(f(x))), \text{ and} \\ g(x) &= s(\mathit{diff})(\sigma(x), sg(x)). \end{aligned}$$

The equalities will be proved by induction on the string x. If $x = e$, from the definitions of the functions A_n and $s(\mathit{diff})$ we deduce:

$$\begin{aligned} f(\sigma(e)) &= A_n(g(f(e))) = A_n(g(e)) = A_n(s(\mathit{diff})(\sigma(e), sg(e))) \\ &= A_n(s(\mathit{diff})(a_1, e)) = A_n(a_1) = A_{n+1}(e). \end{aligned}$$

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

Indeed,

$$\begin{aligned} f(\sigma(\sigma(x))) &= A_n(g(f(\sigma(x)))) = A_n(g(A_{n+1}(x))) \\ &= A_n(s(\mathit{diff})(\sigma(A_{n+1}(x)), sg(A_{n+1}(x)))) \\ &= A_n(s(\mathit{diff})(\sigma(A_{n+1}(x)), a_1)) \\ &= A_n(\overline{c}(\mathit{diff}(c(\sigma(A_{n+1}(x))), c(a_1)))) \\ &= A_n(\overline{c}(\mathit{diff}(c(A_{n+1}(x)) + 1, 1))) \\ &= A_n(\overline{c}(c(A_{n+1}(x)))) = A_n(A_{n+1}(x)) \\ &= A_{n+1}(\sigma(x)). \end{aligned}$$

Using now definition 2, lemma 7, the induction hypothesis and the relations

$$f(x) = A_{n+1}(\pi(x)) \leq A_{n+1}(x) \leq A_{n+k+1}(x), x \in A^*,$$

we deduce that A_{n+1} is in C_{n+k+1} being obtained from functions belonging to C_{n+k+1}, using composition, limited iteration and $s(\mathit{diff})$. □

Lemma 8 *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

$$f(x) < A_0(A_0(x)) \leq A_n(A_n(x))$$

and we can take $k = 2$.

If $f(x) = A_n(x)$, then

$$f(x) \leq A_n(A_n(x))$$

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

$$\begin{aligned}(f \circ g)(x) &= f(g(x)) < A_n^p(g(x)) < A_n^{p+q}(x)),\\ s(\mathit{diff})(f,g)(x) &\leq f(x) < A_n^p(x).\end{aligned}$$

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

Theorem 2 *For every class $C_n, n \geq 1$, and every f in C_n, there exists a natural i (depending upon f) such that $f(x) < A_{n+1}(x)$ for every string x in A^* satisfying $c(x) \geq i$.*

Proof. Assume that f is a function in $C_n, n \geq 1$. In view of lemma 8, we can find a natural $k \geq 2$ (which depends upon 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:

$$A_n^k(x) = A_n^{k-1}(A_n(x)) \leq A_n^{k-1}(A_n(\sigma^{k-1}(x))) < A_n^{k-1}(A_{n+1}(\pi^{k-1}(x))),$$

for every string x with $c(x) > 3k \dot{-} 1$.

Intermediate step: $A_{n+1}(x) = A_n^{k-1}(A_{n+1}(\pi^{k-1}(x)))$, for every string x with $c(x) \geq k$.

We shall prove the equality by induction on x. If $c(x) = k$, then we have

$$\begin{aligned}A_n^{k-1}(A_{n+1}(\pi^{k-1}(x))) &= A_n^{k-1}(A_{n+1}(\pi^{k-1}(\sigma^{c(x)}(e))))\\ &= A_n^{k-1}(A_{n+1}(\pi^{k-1}(\sigma^k(e)))) = A_n^{k-1}(A_{n+1}(a_1))\\ &= A_n^{k-1}(A_n^2(a_1)) = A_n^{k+1}(a_1) = A_n^{c(x)+1}(a_1)\\ &= A_{n+1}(x).\end{aligned}$$

If the equality holds for x, we can prove that

$$A_{n+1}(\sigma(x)) = A_n^{k-1}(A_{n+1}(\pi^{k-1}(\sigma(x)))).$$

Indeed,

$$\begin{aligned}A_n^{k-1}(A_{n+1}(\pi^{k-1}(\sigma(x)))) &= A_n^{k-1}(A_{n+1}(\sigma(\pi^{k-1}(x))))\\ &= A_n^{k-1}(A_n(A_{n+1}(\pi^{k-1}(x))))\\ &= A_n(A_n^{k-1}(A_{n+1}(\pi^{k-1}(x))))\\ &= A_n(A_{n+1}(x)) = A_{n+1}(\sigma(x)),\end{aligned}$$

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 $c(x) \geq 3k \dot{-} 1$ and taking $i = 3k \dot{-} 1$, the proof is finished.

□

Theorem 3 *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

$$sub, \sigma - it_{A,e}, s(diff).$$

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

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 4 *The function* $\overline{A} : A^* \longrightarrow A^*$ *defined by* $\overline{A}(w) = A_{c(w)}(w)$ *is not primitive recursive.*

Proof. Assume, on the contrary, that $\overline{A}$ is primitive recursive. From theorem 3 we get a natural n such that $\overline{A} \in C_n$. By theorem 2, there exists a natural i such that $A(x) < A_{n+1}(x)$ for every x with $c(x) \geq i$. Let x be a string satisfying the condition $c(x) = n + i + 1$. We arrive at a contradiction since

$$\overline{A}(x) = A_{c(x)}(x) = A_{n+i+1}(x) < A_{n+1}(x)$$

(see corollary 2). This completes the proof of the theorem. □

4 Acknowledgements

We are grateful to Dr. Cristian Calude for drawing our attention to these problems and for many helpful remarks.

References

[1] G.Asser. Rekursive Wortfunktionen *Z.Math. Logik Grundlag.Math.* 6(1960), 258-278.

[2] G.Asser. Primitive recursive word-functions of one variable, in E.Borger (ed.), *Computation Theory and Logic*, LNCS 270, Springer 1987, 14-19.

[3] G.Asser. Zur Robinson Charakterisierung der Einstelligen Primitiv Rekursiven Wortfunktionen, *Z.Math.Logik Grundlag.Math.*, 34(1988), 317-322.

[4] C.Calude. *Theories of Computational Complexity*, North-Holland, Amsterdam, New-York, Oxford, Tokio, 1988.

[5] C.Calude, L.Santean. On a Theorem of Gunter Asser, *Z.Math.Logik Grundlag.Math.*, 1990.

[6] F.W.v.Henke, K.Indermark, G.Rose, K.Weichrauch. On Primitive Recursive Wordfunctions, *Computing* 15(1975), 217-234.

[7] M.Tatarim. Darboux property and one-argument primitive recursive string-functions,*Revue Roumaine des Mathematiques Pures et Appliques*, 1987, 79-94.

[8] K.Weichrauch. Teilklassen primitiv-rekursiver Wortfunktionen, *Berichte der GMD* 91(1974), 1-49.