ON A THEOREM OF GÜNTER ASSER

by CRISTIAN CALUDE and LILA SÂNTEAN in Bucharest (Romania)

1. Introduction

Recently, G. Asser [2] has obtained two interesting characterizations of the class of unary primitive recursive string-functions over a fixed alphabet as Robinson algebras. Both characterizations use a somewhat artificial string-function, namely the string-function lexicographically associated with the number-theoretical excess-over-a-square function. Our aim is to offer two new and natural Robinson algebras which are equivalent to Asser's algebras.

Let \( N \) denote the set of naturals, i.e. \( N = \{0, 1, 2, \ldots\} \), and \( N^+ = N \setminus \{0\} \). We consider a fixed alphabet \( A = \{a_1, a_2, \ldots, a_r\} \), and denote by \( A^* \) the free monoid generated by \( A \) under concatenation (with \( e \) as the null-string). By length \( (w) \) we denote the length of the string \( w \) (length \( (e) = 0 \)). For every \( w \in A^* \) and \( m \in N \) let \( w^m = w \ldots w \) (\( m \) times), in case \( m > 0 \), and \( w^0 = e \). By \( Fnc \) (respectively, \( Fnc_A \)) we denote the set of all unary number-theoretical (respectively, string) functions. By \( Z, Succ, E, C, Pd \) we denote the following number-theoretical functions:

\[
Z(x) = x; \quad Succ(x) = x + 1; \quad E(x) = x - 1, \quad \text{where} \quad x \preceq y = \max(x - y, 0).
\]

By \( Z_A, Succ_A, C_A, Pd_A \) we denote the following string-functions:

\[
Z_A(w) = w; \quad Succ_A(w) = w_i (1 \leq i \leq r); \quad C_A(w) = u; \quad \text{where} \quad a(w) = a_i, \sigma(w) = wa_i+1, \text{if} \ 1 \leq i < r, \text{and} \ \sigma(w_0) = \sigma(w) a_1; \quad \delta(e) = e, \ \delta(w) = w (1 \leq i \leq r); \quad \pi(e) = e, \ \pi(\sigma(w)) = w.
\]

Furthermore we use the bijections \( c: A^* \rightarrow N, \check{c}: N \rightarrow A^* \) given by \( c(e) = 0, c(wa_i) = r \cdot c(w) + i, 1 \leq i \leq r, \text{and} \ \check{c}(0) = e, \ \check{c}(m) = \sigma(\check{c}(m)); \) obviously \( c(\check{c}(m)) = m \) and \( \check{c}(c(w)) = w \).

To each \( f \in Fnc \) we associate the string-function \( s(f) \in Fnc_A \) defined by \( s(f)(w) = \check{c}(f(c(w))) \); conversely, to each string-function \( g \) we associate the number-theoretical function \( n(g) \in Fnc \) defined by \( n(g)(x) = c(g((c(x))) \). It is easily seen that for every \( f \in Fnc \) and \( g \in Fnc_A \) one has \( n(s(f)) = f \) and \( s(n(g)) = g \). For example, \( s(C_m) = C_{\check{c}(m)}, s(Succ) = \sigma, \ s(I) = I^A, s(Pd) = \pi \).

For every \( F \subseteq Fnc \) and \( G \subseteq Fnc_A \), we put \( s(F) = \{s(f) | f \in F \} \) and \( n(G) = \{n(g) | g \in G \} \). A mapping from \( Fnc^n \) in \( Fnc \) (\( n \in N^+ \)) is called an \( n \)-ary operator in \( Fnc \), and analogous for \( Fnc_A \). We consider the following operators in \( Fnc \) and \( Fnc_A \):

\[
\begin{align*}
\text{sub}(f, g) &= h \ \text{iff} \ f, g, h \in Fnc \ \text{and} \ h(x) = f(g(x)); \\
\text{it}_{\sigma}(f) &= h \ \text{iff} \ f, h \in Fnc \ \text{and} \ h(0) = x, \ h(y + 1) = f(h(y)); \\
\text{add}(f, g) &= h \ \text{iff} \ f, g, h \in Fnc \ \text{and} \ h(x) = f(x) + g(x); \\
\text{diff}(f, g) &= h \ \text{iff} \ f, g, h \in Fnc \ \text{and} \ h(x) = f(x) - g(x); \\
\text{sub}_{\sigma}(f, g) &= h \ \text{iff} \ f, g, h \in Fnc_A \ \text{and} \ h(w) = f(g(w)); \\
\sigma-\text{it}_{\alpha}(f) &= h \ \text{iff} \ f, h \in Fnc_A \ \text{and} \ h(e) = u, \ h(\sigma(w)) = f(h(w)); \\
\text{it}_{\alpha}(f_1, \ldots, f_r) &= h \ \text{iff} \ f_i, \ldots, f_r, h \in Fnc_A \ \text{and} \ h(e) = u, \ h(\sigma(w)) = f_i(h(w)), 1 \leq i \leq r; \\
\text{con}_{\alpha}(f, g) &= h \ \text{iff} \ f, g, h \in Fnc_A \ \text{and} \ h(w) = f(w) g(w).
\end{align*}
\]
For every operator $\varphi$ in $Fnc$,

$$s(\varphi)(f) = s(\varphi(n(f))), \text{ for every } f \in Fnc;$$

analogously, for every operator $\theta$ in $Fnc$,

$$n(\theta)(g) = n(\theta(s(g))), \text{ for every } g \in Fnc.$$

For example, $s(it_x) = \sigma-it_{\lambda,e(x)}$, $n(\sigma-it_{\lambda,e}) = it_w$.

Finally, for every subset $F \subseteq Fnc$ and every set $X$ of operators in $Fnc$, $[F; X]$ denotes the smallest subset of $Fnc$ which contains $F$ and is closed under the operators belonging to $X$, and analogously for $Fnc_\Lambda$.

A simple, but useful, result in Asser [2] establishes the following relations:

1. For every $F \subseteq Fnc$ and for every set $X$ of operators in $Fnc$, $s([F; X]) = [s(F); s(X)]$.
2. For every $G \subseteq Fnc_\Lambda$ and for every set $Y$ of operators in $Fnc_\Lambda$, $n([G; Y]) = [n(G); n(Y)]$.

2. Main results

The primitive recursive string-functions were introduced by Asser [1] and studied by various authors (see Eilenberg and Elgot [6], Brainard and Landweber [4], Calude [5]). A famous result of R. M. Robinson [9] gives the following characterisation of the class $Prim^1$ of unary primitive recursive number-theoretical functions:

$$Prim^1 = \{\{Succ, E\}; \{\text{sub}, it_0, \text{add}\}\}.$$  

In Asser [2] the following characterizations of the class $Prim_\Lambda^1$ of unary primitive recursive string-functions are obtained:

3. $Prim_\Lambda^1 = \{\{\sigma, s(E)\}; \{\text{sub}_\Lambda, \sigma-it_{\lambda,e}, s(\text{add})\}\}$,

4. $Prim_\Lambda^1 = \{\{\text{Succ}_1, \ldots, \text{Succ}_\Lambda, s(E)\}; \{\text{sub}_\Lambda, it_{\lambda,e}, \text{con}_\Lambda\}\}$.

These characterizations use the somewhat artificial string-function $s(E)$ and the operator $s(\text{add})$ in $Fnc_\Lambda$ is also rather artificial.

In Georgieva [7] (see also Calude [5]) one finds the following result:

$$Prim^1 = \{\{\text{Succ}\}; \{\text{sub}, \text{diff}\} \cup \{it_x | x \in \mathbb{N}\}\}.$$  

This formula can be simplified as follows:

5. $Prim^1 = \{\{\text{Succ}\}; \{\text{sub}, \text{diff}, it_0\}\}$.

All that remains to prove (5) is the inclusion

$$Prim^1 \subseteq P = \{\{\text{Succ}\}; \{\text{sub}, \text{diff}, it_0\}\},$$  

i.e. the closure of $P$ under the operators $it_x$ for $x \in \mathbb{N}_*$. First we note that $P$ contains the functions $sg = it_0(C_1)$, $\bar{sg} = \text{diff}(C_1, sg)$ and that $P$ is closed under sum and product. Now let $f$ in $P$ and $h = it_x(f)$, $x \in \mathbb{N}_*$. If, for every natural $k > 0$, $f^k(x) = f(f(...(f(x))...)) \neq 0$ ($k$ times), then $h = \text{sub}(h^*, \text{Succ})$, where $h^* = it_0(g)$ and $g(y) = x \cdot \bar{sg}(y) + f(y) \cdot sg(y)$ ($\cdot$ denotes the product). In case there exists a natural $k > 0$ such that $f^k(x) = 0$, say the minimal one, then

$$h(y) = h^*(\text{Succ}(y)) \cdot \bar{sg}(t(x)) + h_*(t(x)) \cdot sg(t(x)),$$

where $t = \text{diff}(I, C_1)$, $h_* = it_0(f)$. 
In view of (5) and (1), as a slight improved form of (3) we obtain:

(6) \[ \text{Prim} \left( \{ \sigma \} ; \{ \text{sub}_A, \sigma-\text{it}_A, e, s(\text{diff}) \} \right). \]

The string-function \( s(E) \) is dropped, but the unpleasant operator \( s(\text{diff}) \) in \( \text{Fnc}_A \) is still present. To overcome this difficulty we shall present our first result:

(7) \[ \text{Prim} \left( \{ \text{Succ}^\Lambda, \pi \} ; \{ \text{sub}_A, \text{it}_A, e, \text{con}_A \} \right). \]

For this, we denote by \( F \) the right-hand side of (7). In order to prove (7) we will show that (i) \( \sigma \in F \), (ii) \( F \) is closed under \( \sigma-\text{it}_A, e \), (iii) \( F \) is closed under \( s(\text{diff}) \).

As in the proof of Proposition 2 in Asser [2] we begin with displaying a list of string-functions belonging to \( F \):

a) \( \text{C}^\Lambda_e(w) = e \): \( \text{C}^\Lambda_e(\pi, \ldots, \pi) \).

b) \( \text{x}_i(w) = a_i \) (1 \( \leq i \leq r) \): \( \text{x}_i = \text{sub}_A(\text{Succ}^\Lambda, \text{C}^\Lambda_e). \)

c) \( \text{I}^\Lambda(w) = w \): \( \text{I}^\Lambda = \text{it}_A(\text{Succ}^\Lambda, \ldots, \text{Succ}^\Lambda) \).

d) \( \text{sg}^\Lambda(e) = e \), \( \text{sg}^\Lambda(w) = a_i \), for \( w \neq e \) (1 \( \leq i \leq r) \): \( \text{sg}^\Lambda = \text{it}_A(e, \ldots, e) \).

e) \( \text{succ}^\Lambda(w) = a_i w \) (1 \( \leq i \leq r) \): \( \text{succ}^\Lambda = \text{con}_A(\text{x}_i, \text{I}^\Lambda) \).

f) \( \text{mir}(e) = e \), \( \text{mir}(w) = \text{mir(u) mir(w)} \): \( \text{mir} = \text{it}_A(\text{succ}^\Lambda, \ldots, \text{succ}^\Lambda) \).

g) \( \lambda_i(w) = a_i \) (1 \( \leq i \leq r) \): \( \lambda_i = \text{it}_A(\text{Succ}^\Lambda, \ldots, \text{Succ}^\Lambda) \).

h) \( \gamma_i(w) = a_i \) (1 \( \leq i \leq r) \):

\[ \gamma_i = \text{it}_A(\text{con}_A(\text{I}^\Lambda \ldots \text{I}^\Lambda, \text{x}_i), \text{con}_A(\text{I}^\Lambda \ldots \text{I}^\Lambda, \text{x}_i), \ldots, \text{con}_A(\text{I}^\Lambda \ldots \text{I}^\Lambda, \text{x}_i)) \]

(the \( k \)-th place of the operator \( \text{it}_A, e \) is \( \text{con}_A(\text{I}^\Lambda \ldots \text{I}^\Lambda, \text{x}_i) \) with \( r \) concatenations of \( \text{I}^\Lambda \) and \( k \) concatenations of \( \text{x}_i; \) 1 \( \leq k \leq r) \).

i) \( \alpha_i(w) = u \) if \( w = u a_i a_{i+1} \ldots a_r \) and \( u \) does not terminate with \( a_i \) (1 \( \leq i \leq r) \):

\[ \alpha_i(\text{sub}_A(\text{mir}, \text{sub}_A(\text{it}_A, (\text{Succ}^\Lambda, \ldots, \text{Succ}^\Lambda)))). \]

j) \( \beta_i(w) = a_i a_{i+1} \ldots a_r \) if \( w = \alpha(w) a_i a_{i+1} \ldots a_r \) (1 \( \leq i \leq r) \): \( \beta_i = \text{it}_A(\text{con}_A(\text{I}^\Lambda \ldots \text{I}^\Lambda, \text{x}_i), \ldots, \text{con}_A(\text{I}^\Lambda \ldots \text{I}^\Lambda, \text{x}_i)) \).

k) \( \text{sg}^\Lambda(e) = a_i \), \( \text{sg}^\Lambda(w) = e \), for \( w \neq e \) (1 \( \leq i \leq r) \):

\[ \text{sg}^\Lambda = \text{sub}_A(\text{sg}^\Lambda, \text{sub}_A(\text{I}^\Lambda, \text{sg}^\Lambda)) \).

l) \( \psi_i(e) = a_{i+1}, \psi_i(w) = a_i \), for \( w \neq e \) (1 \( \leq i < r) \): \( \psi_i = \text{con}_A(\text{sg}^\Lambda, \text{sg}^\Lambda) \);

\[ \psi_i(e) = a_1, \psi_i(w) = a_r \), for \( w \neq e \): \( \psi_r = \text{con}_A(\text{sg}^\Lambda, \text{sg}^\Lambda) \).

m) \( \chi(e) = e \), \( \chi(u a_i) = u a_{i+1} \), for \( 1 \leq i < r \): \( \chi(u a_r) = u a_i \):

\[ \chi = \text{sub}_A(\text{mir}, \text{sub}_A(\text{con}_A(\text{I}^\Lambda, \psi_1), \ldots, \text{con}_A(\text{I}^\Lambda, \psi_r)) \).

We are now in a position to prove (i): \( \sigma \in F \). Indeed,

\[ \sigma = \text{con}_A(\text{sub}_A(\text{sg}^\Lambda, \alpha), \text{con}_A(\text{sub}_A(\chi, \alpha), \text{sub}_A(\chi, \beta))) \).
Passing to (ii) we note that

\[ (8) \quad \sigma_{it_{A^e}}(f) = sub_A(it_{A^e}(f, \ldots, f), y) \],

i.e. \( F \) is closed under the operator \( \sigma_{it_{A^e}} \).

To finish the proof we recall that, for every \( f \in F_{nc} \) and \( n \in \mathbb{N} \), \( f^0 = I^A \) and \( f^n = sub_A(f, sub_A(f, \ldots, sub_A(f, f), \ldots)) \), \( n \) times. Using a double lexicographical induction one proves the equality

\[ \pi_{diff}(u) = c(diff(c(u), c(w))) \text{ for } u, w \in \mathbb{N}^* \],

which enables us to write the formula

\[ (9) \quad s(diff)(f, g) = sub_A(it_{A^e}(\sigma, \pi, I^A, \ldots, I^A), con_A(sub_A(\gamma_1, f), sub_A(\gamma_2, g))) \],

for all \( f, g \in F_{nc} \), thus proving (iii). This ends the proof of (7).

Our second Robinson algebra is the following:

\[ (10) \quad Prim_A^n = \{ \gamma; \{ sub_A, it_{A^e}, con_A \} \} \].

In view of (6) we must prove that the right-hand side of (10) is closed under \( \sigma_{it_{A^e}} \) and \( s(dif) \).

Again we proceed with displaying a sequence of primitive recursive string-functions belonging to the right-hand side of (10):

a) \( I^A = sub_A(\pi, \sigma) \).

b) \( C^A = it_{A^e}(\pi, \ldots, \pi) \).

c) \( \lambda_i = sub_A(\sigma, sub_A(\sigma, \ldots, sub_A(\sigma, C^A))) \),

where the operator \( sub_A \) appears \( i \) times (\( 1 \leq i \leq r \)).

d) \( \gamma_i = it_{A^e}(con_A(M_i(I^A), M_i(\lambda_i)) \text{ con}_A(M_i(I^A), M_i(\lambda_i)), \ldots, \text{ con}_A(M_i(I^A), M_i(\lambda_i))) \),

with \( M_i(f) = \text{ con}_A(f, \text{ con}_A(f, \ldots, \text{ con}_A(f, f), \ldots) \), where \( f \) is any string-function and the operator \( \text{ con}_A \) appears \( j \geq 1 \) times.

The proof of (10) is complete in view of (8) and (9).

Finally, we conjecture the validity of the following formula:

\[ (11) \quad Prim_A^n = \{ \gamma; \{ sub_A, \sigma_{it_{A^e}}, \text{ con}_A \} \}. \]

In view of (2) and \( n(Succ^i)(x) = Succ^i(x), 1 \leq i \leq r, \) (11) holds iff its right-hand side is closed under the operator \( it_{A^e} \).

3. Final remarks

After finishing this paper we have learnt the following new characterizations of \( Prim_A^n \) due to G. Asser [3]:

\[ Prim_A^n = \{ \{ Succ^1, \ldots, Succ^n, \lambda \}; \{ sub_A, it_{A^e}, con_A \} \} \]

\[ = \{ \{ Succ^1, \ldots, Succ^n, \varphi \}; \{ sub_A, it_{A^e}, con_A \} \}, \]

where \( \lambda, \varphi \) are the component functions of the pairing function.
\( \gamma(u, v) = a_1^{\text{length}(u)} a_2 u v a_3^{\text{length}(v)} \),

i.e., if \( w = \gamma(u, v) \) for some strings \( u, v \), then \( \lambda(w) = u \) and \( \varrho(w) = v \), else \( \lambda(w) = \varrho(w) = e \).

Furthermore G. Asser (communication of July 13, 1989) has perceived that in (7) the function \( \pi \) can be replaced by the function \( \delta \), i.e.

\[
\text{(12)} \quad \text{Prim}_A = \left\{ \{ \text{Succ}_A^\wedge, \ldots, \text{Succ}_A, \delta \}, \{ \text{sub}, \text{it}_A, \text{e}, \text{con}_A \} \right\}.
\]

The proof is essentially the same as for (7). Only a) must be replaced by

\[
C_\varepsilon^A = \text{it}_A, \varepsilon (\delta, \ldots, \delta),
\]

and (9) must be replaced by

\[
s(\text{diff})(f, g) = \text{sub}_A(\text{it}_A, \varepsilon (\sigma, \ldots, \sigma), \text{sub}_A(\text{it}_A, \varepsilon (\text{Succ}_A^\wedge, \delta, \ldots, \delta), (\text{con}_A(\text{sub}_A(y_1, f), \text{sub}_A(y_2, g))))).
\]

References


