PARALLEL COMMUNICATING GRAMMAR SYSTEMS: THE REGULAR CASE

GHEORGHE PĂUN and LILA SÂNTEAN

The parallel communicating grammar systems are introduced here as grammar systems working in parallel and sending messages (strings) to a master grammar whose terminal strings constitute the language we look for. Many variants can be considered: in this paper, we restrict to the case when all grammars are regular and we investigate the generative capacity of such devices.

AMS CLASSIFICATION: 68F45, 68F50
C.R. CATEGORY: F 4.2.

1. Introduction

As it is wellknown, the formal language theory has been developed mainly in connection with programming languages and a lot of theoretical problems were raised in this frame. The topic we discuss here is also suggested by a question of our days practical computer science, namely by the important topic of parallel computing. There are many other models of parallel computers (see the algebraic one of [4], the automata theoretical one of [2], and so on). In this paper we propose a grammatical model, trying to involve as few as possible non syntactic components. The parallel communicating grammar systems were introduced in this aim. They consist of n separated usual Chomsky grammars working simultaneously (each of them starts from its own axiom); one of these grammars is distinguished (is a master) and can ask to the other grammars the current strings as they are (terminal or not). The grammar which communicates in this way to the master returns to the axiom and resumes working. The terminal strings generated by the master grammar constitute the generated language. (In some sense, our mechanisms are similar to the distributed grammar systems of [1], but there the grammars do not work simultaneously and they cooperate in quite a different way to the generation of a string.)

Of course, one can define a lot of variants of such systems, depending on the communication protocol, on the type of grammars and so on. We consider here the simplest case, with (i) only one master grammar, (ii) which only asks for messages from other grammars, (iii) with regular grammars, and (iv) returning to the axiom after each communication step. As it is expected, the power of such systems is considerably larger than that of regular grammars.

2. Definitions and examples

For a vocabulary V, denote by V^* the free monoid generated by V under the operation of concatenation and the null element λ. The length of $x \in V^*$ is denoted by $|x|$ and $|x|_U$, $U \subseteq V$, is the length of the string obtained by erasing from x all symbols not in U.

A *parallel communicating grammar system* (*PCGS*, for short) of degree n, $n \geq 1$, is an n-tuple

$$\gamma = (G_1, G_2, \ldots, G_n)$$

where each G_i is a Chomsky grammar, $G_i = (V_{N,i}, V_{T,i}, S_i, P_i)$, $1 \leq i \leq n$, and $V_{N,1}$ includes a set $C = \{A_2, A_3, \ldots, A_n\}$, of distinguished symbols such that $C \cap \bigcup_{i=2}^{n} V_{Gi} = \emptyset$ (as usual, $V_{Gi} = V_{N,i} \cup V_{T,i}$) ($C$ is called the communication set and A_i, $2 \leq i \leq n$, are called communication symbols), and $V_{N,1} \cap \left(\bigcup_{i=2}^{n} V_{T,i}\right) = \emptyset$.

For two n-tuples $(x_1, \ldots, x_n)$, $(y_1, \ldots, y_n)$, $x_i, y_i \in V^*_{Gi}$, $1 \leq i \leq n$, we write $(x_1, \ldots, x_n) \Rightarrow (y_1, \ldots, y_n)$ if either

$|x_1|_C = 0$ and, for each i, $1 \leq i \leq n$, we have $x_i \Rightarrow y_i$ in the grammar G_i or x_i is terminal (according to G_i) and $y_i = x_i$, or

$x_1 = z_1 A_{i_1} z_2 A_{i_2} \ldots z_k A_{i_k} z_{k+1}$, $k \geq 1$, $z_i \in (V_{G_1} - C)^*$, $1 \leq i \leq k+1$,

$A_{i_j} \in C$, $1 \leq j \leq k$, $y_1 = z_1 x_{i_1} z_2 x_{i_2} \ldots z_k x_{i_k} z_{k+1}$, and

$y_i = x_i$ for $i \in \{2, \ldots, n\} - \{i_1, \ldots, i_k\}$,

$y_{i_j} = S_{i_j}$ for $1 \leq j \leq k$.

(The communication has priority and during a communication step, besides replacing each A_{i_j} by the corresponding string x_{i_j} and each x_{i_j} by S_{i_j}, no further rewritings are done. When no communication symbol appears, the derivation is a usual componentwise one, one step in each grammar G_i, $1 \leq i \leq n$.)

The language generated by γ is

$$L(\gamma) = \{x \in V^*_{T,1} \mid (S_1, \ldots, S_n) \overset{*}{\Rightarrow} (x, \alpha_2, \ldots, \alpha_n),$$
$$\alpha_i \in V^*_{Gi}, 2 \leq i \leq n\}$$

where $\overset{*}{\Rightarrow}$ is the reflexive transitive closure of the relation $\overset{*}{\Rightarrow}$.

Example 1. Let $\gamma_1 = (G_1, G_2)$ be the *PCGS* with

$$G_1 = (\{S_1, S_2, A_2\}, \{a, b, c\}, \{S_1 \to aS_1, S_1 \to aA_2,$$
$$S_2 \to aS_1, S_2 \to c\}),$$

$$G_2 = (\{S_2\}, \{b\}, S_2, \{S \to bS_2\})$$

A terminal derivation in γ_1 has the following form :

$$(S_1, S_2) \overset{*}{\Rightarrow} (a^{k_1}S_1, b^{k_1}S_2) \Rightarrow (a^{k_1+1}A_2, b^{k_1+1}S_2) \Rightarrow$$

$$(a^{k_1+1}b^{k_1+1}S_2, S_2) \Rightarrow (a^{k_1+1}b^{k_1+1}aS_1, bS_2) \overset{*}{\Rightarrow} \ldots$$

$$\ldots \overset{*}{\Rightarrow} (a^{k_1+1}b^{k_1+1} \ldots a^{k_r+1}b^{k_r+1}aS_1, bS_2) \Rightarrow$$

$$\Rightarrow (a^{k_1+1}b^{k_1+1} \ldots a^{k_r+1}b^{k_r+1}a^{k_{r+1}+1}A_2,\ b^{k_{r+1}+1}S_2) \Rightarrow$$

$$\Rightarrow (a^{k_1+1}b^{k_1+1} \ldots a^{k_{r+1}+1}b^{k_{r+1}+1}S_2,\ S_2) \Rightarrow$$

$$\Rightarrow (a^{k_1+1}b^{k_1+1} \ldots a^{k_{r+1}+1}b^{k_{r+1}+1}c,\ bS_2),\ r \geqslant 0,\ k_i \geqslant 0,\ 1 \leqslant i \leqslant r+1,$$

hence

$$L(\gamma_1) = \{a^{k_1}b^{k_1} \ldots a^{k_r}b^{k_r}c \mid r \geqslant 1,\ k_i \geqslant 1,\ 1 \leqslant i \leqslant r\}$$

This language is not linear, although G_1, G_2 are regular grammars.

Example 2. Consider $\gamma_2 = (G_1, G_2)$ with

$$G_1 = (\{S_1, A_2\}, \{a, b, c, d\}, S_1, \{S_1 \to cS_1d,\ S_1 \to cA_2d\})$$

$$G_2 = (\{S_2\}, \{a, b\}, S_2, \{S_2 \to aS_2b,\ S_2 \to ab\})$$

One can easily see that

$$L(\gamma_2) = \{c^m a^n b^n d^m \mid m \geqslant n \geqslant 1\}$$

and this language is not context-free (mark the occurrences of the symbol a and use Ogden lemma) although G_1, G_2 are linear grammars.

Example 3. Consider the system $\gamma_3 = (G_1, G_2, G_3)$ with

$$G_1 = (\{S_1, S_2, S_3, A_2, A_3\}, \{a, b, c\}, S_1, \{S_1 \to aS_1,$$

$$S_1 \to a^3A_2,\ S_2 \to b^2A_3,\ S_3 \to c\})$$

$$G_2 = (\{S_2\}, \{b\}, S_2, \{S_2 \to bS_2\})$$

$$G_3 = (\{S_3\}, \{c\}, S_3, \{S_3 \to cS_3\})$$

A derivation in γ_3 is of the form

$$(S_1, S_2, S_3) \stackrel{*}{\Rightarrow} (a^kS_1, b^kS_2, c^kS_3) \Rightarrow$$

$$\Rightarrow (a^{k+3}A_2, b^{k+1}S_2, c^{k+1}S_3) \Rightarrow (a^{k+3}b^{k+1}S_2, S_2, c^{k+1}S_3)$$

$$\Rightarrow (a^{k+3}b^{k+3}A_3, bS_2, c^{k+2}S_3) \Rightarrow (a^{k+3}b^{k+3}c^{k+2}S_3, bS_2, S_3)$$

$$\Rightarrow (a^{k+3}b^{k+3}c^{k+3}, b^2S_2, cS_3)\ k \geqslant 0,$$

hence

$$L(\gamma_3) = \{a^nb^nc^n \mid n \geqslant 3\}$$

a language which is not context-free; note that G_1, G_2, G_3 are regular grammars.

As we have said, we investigate here only *PCGS's* with regular components, that is with rules of the forms $X \to aY$, $X \to a$, X, Y nonterminals, a terminal. We denote by $\mathcal{R}(n)$ the family of languages generated in this way, $n \geqslant 1$. As usual, $\mathcal{L}_i$, $i = 0, 1, 2, 3$, will denote the four families in Chomsky hierarchy and $\mathcal{L}_{lin}$ will be the family of linear languages. Two languages will be considered identical if they differ at most in the null string λ.

3. The generative capacity of regular PCGS's

Lemma 1. $\mathscr{R}(n) \subseteq \mathscr{R}(n+1)$, $n \geq 1$.

Proof. Given $\gamma = (G_1, \ldots, G_n)$ a regular *PCGS*, we construct $\gamma' = (G_1', G_2, \ldots, G_n, G_{n+1})$, with $G_1' = (V_{N,1} \cup \{A_{n+1}\}, V_{T,1}, S_1, P_1)$ and $G_{n+1} = (\{S_{n+1}\}, \{a\}, S_{n+1}, \{S_{n+1} \to a\})$. As P_1 does not introduce the symbol A_{n+1}, we have $L(\gamma) = L(\gamma')$, hence the inclusion $\mathscr{R}(n) \subseteq \mathscr{R}(n+1)$ follows.

Theorem 1. (i) $\mathscr{R}(n) - \mathscr{L}_{lin} = \emptyset$, $n \geq 2$; (ii) $\mathscr{R}(n) - \mathscr{L}_2 \neq \emptyset$, $n \geq 3$.

Proof. Follows from Lemma 1 and Examples 1, 3, respectively.

Theorem 2. $\mathscr{L}_{lin} - \mathscr{R}(n) \neq \emptyset$ *for all* $n \geq 1$.

Proof. Let *mi* denote the mirror image and consider the language

$$L = \{xc\ mi(x) \mid x \in \{a, b\}^*\}$$

The linear grammar $G = (\{S\}, \{a, b, c\}, S, \{S \to aSa, S \to bSb, S \to c\})$ generates L, hence $L \in \mathscr{L}_{lin}$. Suppose that $L \in \mathscr{R}(n)$ for some n, $L = L(\gamma)$, $\gamma = (G_1, \ldots, G_n)$. The communication steps do not increase the number of terminal symbols introduced in an n-tuple $(x_1, \ldots, x_n)$, $x_i \in V_{Gi}^*$, $1 \leq i \leq n$, hence, in order to generate an arbitrarily long string of L we need arbitrarily many non-communication steps.

Given a string $\alpha = xc\ mi(x)$ in L with a long enough x, we can find a derivation of α of the form

$$D: (S_1, \ldots, S_n) \overset{*}{\Rightarrow} (xcy_1X_1, y_2X_2, \ldots, y_nX_n) \overset{*}{\Rightarrow}$$

$$\overset{*}{\Rightarrow} (xcy_1y_1'X_1, y_2'X_2, \ldots, y_n'X_n) \overset{*}{\Rightarrow} (xc\ mi(x), y_2'', \ldots, y_n'')$$

with $X_1 \in V_{N,1} - C$, $X_i \in V_{N,i} \cup \{\lambda\}$, $2 \leq i \leq n$.

Note that the symbol's c occurrence in this derivation was already produced and that the subderivation $(xcy_1X_1, y_2X_2, \ldots, y_nX_n) \overset{*}{\Rightarrow} \overset{*}{\Rightarrow} (xcy_1y_1'X_1, y_2'X_2, \ldots, y_n'X_n)$ can be iterated. Moreover, $y_1' \neq \lambda$ (G_1 is a regular λ-free grammar). Denote by D' the derivation obtained from D by repeating two times the above subderivation. Clearly, D' is a correct derivation in γ and the generated string is of the form xcy, with $|y| > |x|$. This is not a string in L, contradiction, therefore $L \notin \mathscr{R}(n)$.

Corollary. $\mathscr{L}_{lin}$ *is incomparable with all* $\mathscr{R}(n)$, $n \geq 2$, *and* $\mathscr{L}_2$ *is incomparable with all* $\mathscr{R}(n)$, $n \geq 3$.

Proof. Combine the above two theorems.

Theorem 3. $\mathscr{R}(2) \subset \mathscr{L}_2$, *strict inclusion.*

Proof. In view of Theorem 2, it is enough to prove the inclusion. For, let $\gamma = (G_1, G_2)$ be a *PCGS* and construct the context-free grammar $G = (V_N, V_{T,1}, S, P)$ with

$$V_N = V_{N,1} \cup V_{N,1} \times (V_{N,2} \cup \{*\}) \cup \{X' \mid X \in V_{N,1} \cap V_{N,2}\} \cup \{S\}$$

and P containing the next rules:

1) $S \to S_1$

2) $X \to aY$, for $X \to aY \in P_1$, $Y \neq A_2$
   $X \to a$, for $X \to a \in P_1$

(We simulate in this way the derivations in G_1 which involve no communication step.)

3) $S \to (S_1, Z)\, Z'$, $Z \in V_{N,1} \cap V_{N,2}$

$S \to (S_1, *)$

$(X, Y) \to a(X', Y')\, b$, for $X \to aX' \in P_1$, $a \in V_{T,1}$,

$Y' \to bY \in P_2$, $b \in V_{T,2}$

$(X, Y) \to ab$, for $X \to aA_2 \in P_1$, $a \in V_{T,1}$,

$S_2 \to bY \in P_2$, $b \in V_{T,2}$

$(X, *) \to a(X', *)$, for $X \to aX' \in P_1$, $a \in V_{T,1}$

$(X, *) \to a(X', Y)b$, for $X \to aX' \in P_1$, $a \in V_{T,1}$,

$Y \to b \in P_2$, $b \in V_{T,2}$

$(X, *) \to ab$, for $X \to aA_2 \in P_1$, $a \in V_{T,1}$,

$S_2 \to b \in P_2$, $b \in V_{T,2}$

(The derivations in G_1, G_2 are simultaneously simulated, that in G_1 in the usual way and that in G_2 reversed, from the right to the left. The derivation in G_2 may be terminal or shorter than that in G_1 — this is the role of the symbol $*$. The communication is simulated by the rules $(X, Y) \to ab$, $(X, *) \to ab$.)

4) $Z' \to (Z, *)$

$Z' \to (Z, Y)Y'$, $Z, Y \in V_{N,1} \cap V_{N,2}$

(If the derivation in G_2 produces a string xZ and we have to do a communication step, then the derivation can continue only when $Z' \in V_{N,1} \cap V_{N,2}$).

From the above explanations it is easy to see that $L(\gamma) = L(G)$, hence $L(\gamma) \in \mathscr{L}_2$ and the proof is over.

We shall establish now the relation between families $\mathscr{R}(n)$, $n \geq 1$, and $\mathscr{L}_1$, the family of context sensitive languages.

Theorem 4. Each language $L \in \mathscr{R}(n)$, $n \geq 1$, is letter equivalent to a regular language.

Proof: Let $L \in \mathscr{R}(n)$ be a language generated by a *PCGS* $\gamma = (G_1, \ldots, G_n)$. We construct a matrix grammar (see [3], [5], for definitions) $G = (V_N, V_T, S, M)$ with

$$V_N = \bigcup_{i=1}^{n} \{X^{(i)} \mid X \in V_{N,i}\} \cup \{U_i \mid 2 \leq i \leq n\} \cup V_{N,1} \cup \{S, Z\}$$

$$V_T = V_{T,1} \cup \{c_i \mid 1 \leq i \leq n\} \cup \{c'_i \mid 1 \leq i \leq n\}$$

and M consisting of the following matrices:

1) $(S \to c_1 S_1^{(1)} c_2 S_2^{(2)} \ldots c_n S_n^{(n)})$

2) $(X_1^{(1)} \to a_1 Y_1^{(1)}, X_2^{(2)} \to a_2 Y_2^{(2)}, \ldots, X_n^{(n)} \to a_n Y_n^{(n)})$,

where $X_1 \to a_1 Y_1$ is a rule in P_1

and, for $i = 2, \ldots, n$, one of the next cases holds :

i) $X_i \to a_i Y_i \in P_i$

ii) $X_i \to a_i \in P_i$, $a_i \in V_{T,i}$, and $Y_i^{(i)} = U_i$

iii $a_i = \lambda$, $X_i^{(i)} = Y_i^{(i)} = U_i$

(We simulate in this way a non-communication step in a derivation in γ. The derivation in G_i starts from $S_i^{(i)}$ and the obtained string is bounded by the „brackets" c_i, c'_i. When the derivation in some G_i is finished, a special nonterminal, U_i, is introduced, encoding that.)

3) $(A_i^{(1)} \to c'_1, X_i^{(i)} \to c'_i c_1 X_i^{(1)} c_i S_i^{(i)})$, $2 \leqslant i \leqslant n$,

$(A_i^{(1)} \to c'_1, U_i \to c'_i Z)$, $2 \leqslant i \leqslant n$

(These matrices simulate a communication step in γ. When a symbol A_i appears and the derivation in G_i was not terminal, the corresponding symbol X_i is prepared for further derivations in G_1 — it is replaced by $X_i^{(1)}$, with c_1 in its left hand side — and G_i resumes rewriting from $S_i^{(i)}$. When the derivation in G_i was terminal, the whole derivation ends, as G_1 cannot do further rewritings. The symbol Z is introduced in order to encode this information. At each communication step, the right „brackets" c'_1 and c'_i are introduced.)

4) $(Z \to Z, X_i^{(j)} \to \lambda)$, $X_i \in V_{N,i}$, $2 \leqslant i \leqslant n$,

$(Z \to \lambda)$

(In the presence of Z, all nonterminal occurrences are erased, then Z is removed too.)

5) $(S \to c_1 S_1 c'_1)$

$(X \to aY)$, for $X \to aY \in P_1$, $Y \notin \{A_i \mid 2 \leqslant i \leqslant n\}$,

$(X \to a)$, for $X \to a \in P_1$, $a \in V_{T,1}$

(The non-communication derivations in G_1 are simulated in this way.)

As one can easily see, each sentential form of a derivation in G contains at most n nonterminal symbols. Consequently, G is a matrix grammar of finite index.

From the above explanations, we find that the strings of $L(G)$ are of the form

$$\alpha = c_1 x_1 c'_1 y_1 c_{i_1} x_{i_1} c'_{i_1} y_2 c_{i_2} x_{i_2} c'_{i_2} y_3 \ldots y_m c_{i_m} x_{i_m} c'_{i_m} y_{i_{m+1}}$$

where $x_1 \in V_{T,1}^*$, y_j, $x_{i_j} \in V_{T,k}^*$, for some k, $1 \leqslant k \leqslant n$, and all strings x_1, x_{i_j}, $1 \leqslant j \leqslant m$, correspond either to strings generated by G_1 or to strings generated by some G_k, $2 \leqslant k \leqslant n$, and communicated to G_1 in a correct derivation in γ; the strings y_i, $1 \leqslant i \leqslant m$ (that is the substrings not bounded by "brackets" c_r, c'_r, $1 \leqslant r \leqslant n$), correspond to strings generated by some G_k, $2 \leqslant k \leqslant n$, which were not communicated to G_1 in the γ derivation. Clearly, if we remove from α all markers c_i, c'_i as well as all the parasitic substrings

y_i, then we obtain a permutation of a string in $L(\gamma)$. These removings can be done by the following *gsm* :

$$g = (\{s_i \mid 0 \leqslant i \leqslant n\}, V_T, V_{T,1}, s_0, \{s_0\}, P)$$

with P containing the next rules for all i, $1 \leqslant i \leqslant n$:

$$s_0 c_i \to s_i$$

$$s_i a \to a s_i, \ a \in V_{T,1}$$

$$s_i c'_i \to s_0$$

$$s_0 c'_i \to s_0$$

$$s_0 a \to s_0, \ a \in V_{T,1}$$

Consequently, $L(\gamma)$ is letter equivalent to $g(L(G))$. The family of finite index matrix languages is closed under arbitrary *gsm* mappings ([3]), hence $L(\gamma)$ is letter equivalent to a finite index matrix language; in turn, these languages are letter-equivalent to regular languages ([2]), which completes the proof.

Corollary. (i) *Each language in* $\mathscr{R}(n)$, $n \geqslant 1$, *is semilinear (its image through Parikh mapping is a semilinear set).*

(ii) *Each one-letter language in* $\mathscr{R}(n)$, $n \geqslant 1$, *is regular.*

Theorem 5. *Each family* $\mathscr{R}(n)$, $n \geqslant 1$, *is strictly included in* $\mathscr{L}_1$.

Proof. In view of the previous result, it is enough to prove the inclusion. This will be done by a direct construction: we synchronously simulate derivations in grammars G_i of a *PCGS* γ, and, using the possibilities offered by a type — 0 grammar, we move the strings asked for by G_1 and erase the parasitic strings and symbols (markers). As the grammars G_i are regular (each rule introduces a terminal), the workspace of this type—0 grammar will be linearly bounded, hence the generated language will be context sensitive.

Here are the details of this construction.

Let $\gamma = (G_1, \ldots, G_n)$ be a regular *PCGS* and consider the type—0 grammar $G = (V_N, V_{T,1}, S, P)$ with

$$V_N = \bigcup_{i=1}^{n} V_{N,i} \cup \{X_i \mid 1 \leqslant i \leqslant n+1\} \cup \{D_i \mid 1 \leqslant i \leqslant n+1\} \cup$$

$$\cup \ \{Z, Z', E, S\} \cup \left\{\alpha' \mid \alpha \in \bigcup_{i=2}^{n} V_{G_i}\right\}$$

and P containing ths next rules :

1) $S \to X_1 Z S_1 X_2 S_2 \ldots X_n S_n X_{n+1}$

$Za \to aZ, \ a \in V_{T,1}$

$ZB \to D_1 B, \ B \in V_{N,1} - \{A_2, \ldots, A_n\}$

(We check whether the string between X_1, X_2 is not terminal and contains no communication symbol. In the affirmative case one introduces the symbol D_1.)

2) $D_iBX_{i+1} \to \alpha X_{i+1} D_{i+1}$, for $B \to \alpha \in P_i$, $1 \leqslant i \leqslant n$

$D_iX_{i+1} \to X_{i+1} D_{i+1}$, $1 \leqslant i \leqslant n$

$D_ia \to aD_i$, $a \in V_{T,i}$, $1 \leqslant i \leqslant n$

(The symbols D_i, $1 \leqslant i \leqslant n$, simulate a synchronous derivation in γ.)

3) $\alpha D_{n+1} \to D_{n+1}\alpha$, $\alpha \in \bigcup_{i=1}^{n} V_{G_i} \cup \{X_2, \ldots, X_{n+1}\}$

$X_1D_{n+1} \to X_1Z$

(The process can be iterated.)

4) $ZA_i \to Z_i$, $2 \leqslant i \leqslant n$

$Z_i\alpha \to \alpha Z_i$, $\alpha \in \bigcup_{j=2}^{i-1} V_{G_j} \cup \{X_2, \ldots, X_{i-1}\}$, $2 \leqslant i \leqslant n$

$Z_iX_i\alpha \to \alpha' Z_iX_i$, $\alpha \in V_{G_i}$, $2 \leqslant i \leqslant n$

$Z_iX_iX_{i+1} \to Z'X_iS_iX_{i+1}$, $2 \leqslant i \leqslant n$

$\beta\alpha' \to \alpha'\beta$, $\alpha \in \bigcup_{i=2}^{n} V_{G_i}$, $\beta \in \bigcup_{i=2}^{n} V_{G_i} \cup \{X_3, \ldots, X_n\}$

$X_2\alpha' \to \alpha X_2$, $\alpha \in \bigcup_{i=2}^{n} V_{G_i}$

$\alpha Z' \to Z'\alpha$, $\alpha \in \bigcup_{i=1}^{n} V_{G_i} \cap \{X_2, \ldots, X_n\}$

$X_1Z' \to X_1Z$

(When a symbol A_i is present, then the string generated between X_i, X_{i+1} is transported in the left hand of X_2 and S_i is introduced between X_i, X_{i+1}. The symbol Z' checks whether the process is correctly accomplished.)

5) $ZX_2 \to E$

$E\alpha \to E$, $\alpha \in \bigcup_{i=2}^{n} V_{G_i} \cup \{X_3, \ldots, X_{n+1}\}$

$E \to \lambda$

$X_1 \to \lambda$

(The derivation ends when the string between X_1, X_2 is terminal.)

Clearly, $L(\gamma) = L(G)$, and, for each string $x \in L(G)$ we have

$$WS(x, G) \leqslant |x| + (n-1)|x| + n + 2 \leqslant (2n+2)|x|$$

therefore $L(G) \in \mathscr{L}_1$ and the proof is over.

4. Final remarks

First, let us note that all the proofs of this paper — excepting the last one — remain valid (with minor modifications) for right-linear grammars (rules $X \to xY$, $X \to x$, with x terminal string, not necessarily a symbol). The proof of Theorem 5 remains valid only for λ-free $PCGS$, without rules of the form $A \to B$, in the master grammar G_1 (otherwise unbounded erasings can appear, hence a different construction is needed for proving — if possible — the containement into $\mathscr{L}_1$).

Moreover, it is natural to investigate further types of grammars: linear, context-free, etc. (a sister paper, dealig with this topic, is in progress). Besides this research topic, also remained open one specific problem : is the hierarchy $\mathscr{R}(n)$, $n \geqslant 1$, an infinite one? From Theorems 1, 2, 3 we find that $\mathscr{R}(1) \subset \mathscr{R}(2) \subset \mathscr{R}(3)$ are strict inclusions. It is likely that $\mathscr{R}(n) \subseteq \mathscr{R}(n+1)$ is proper for all $n \geqslant 1$.

Of course, a more general problem is that of defining and investigating different variants of $PCGS$'s, maybe starting from "practical" questions. The domain seems to be quite promising.

References

1. E. CSUHAJ-VARJU, J. DASSOW. Distributed grammar systems, *Inform. Sci.* (submitted, 1988).
2. K. ČULIK, J. GRUSKA, A. SALOMAA. *Systolic trellis automata (for VLSI)*, Univ. of Waterloo, Dept. of Computer Science, Res. Rep. CS-81-34 (1981).
3. J. DASSOW, Gh. PĂUN. *Regulated rewriting in formal language theory*, Akademie Verlag, Berlin, 1989.
4. C. A. R. HOARE. Communicating sequential processes, *Comm. ACM*, 1, 8 (1978), 666-677.
5. A. SALOMAA. *Formal languages*, Academic Press, New York, London, 1973.

Received September 1988

The Computing Centre
of the University of Bucharest
Str. Academiei 14
70109 București