KYBERNETIKA — VOLUME 28 (1992), NUMBER 2, PAGES 155–166

ON THE SYNTACTIC COMPLEXITY OF PARALLEL COMMUNICATING GRAMMAR SYSTEMS

GHEORGHE PĂUN

Paper dedicated to Professor Solomon Marcus, on his 65th birthday.

We compare the complexity of generating a language by a context-free grammar or by a parallel communicating grammar system (*PCGS*), in the sense of Gruska's measures *Var*, *Prod*, *Symb*. Then we define a specific measure for *PCGS*, *Com*, dealing with the number of communication symbols appearing in a derivation. The results are the expected ones: the *PCGS* are definitely more efficient than context-free grammars (the assertion will receive a precise meaning in Section 2), the parameter *Com* introduces an infinite hierarchy of languages, is incomparable with *Var*, *Prod*, *Symb*, and cannot be algorithmically computed.

1. PARALLEL COMMUNICATING GRAMMAR SYSTEMS

The main problem of the classical formal language theory is to study the way a language can be generated/recognized by a (hence *one*) grammar/automaton. However, in the present-day computer science a lot of circumstances there exist when we deal with more "processors" concerned with the same task: computer nets, distributed data bases, parallel computers, distributed expert systems, computer conferencing and so on. Thus, a natural research topic is to consider "systems of grammars", working together in a well defined way and generating *one* language.

Two classes of such grammar systems can be defined, depending on the working protocol: sequential (in each moment only one grammar is enabled to work), or parallel (the components work simultaneously, in a synchronized manner). The former type is considered in [2] (and investigated in a series of subsequent papers). The later leads to parallel communicating grammar systems (*PCGS*, for short). They were introduced in [11] and were investigated in [8], [9], [10], [14], from various (theoretical) points of view. Details about motivation and a survey of results can be found in [13].

Informally speaking, a *PCGS* consist of n usual Chomsky grammars, working simultaneously, each on its own sentential form, and communicating each other by sending, on request, the correct sentential form, from one component to another; the language generated in this way by a "master" component of the system is considered the language generated by the whole system.

Beside being a natural grammatical model of parallel computing, the $PCGS$ prove to be also a mathematically appealing topic, rich in (often difficult) theoretical problems. Here we investigate two basic variants: centralized and non-centralized query-only systems.

Before presenting their definition, we specify some notations.

For a vocabulary V, denote by V^* the free monoid generated by V, by λ the null element of V^*, by $|x|$ the length of x and by $|x|_U$ the length of the string obtained by erasing from x all symbols not in U, $U \subseteq V$; $V^+ = V^* - \{\lambda\}$. For a Chomsky grammar $G = (V_N, V_T, S, P)$, V_N is the nonterminal vocabulary, V_T is the terminal one, S is the axiom and P is the set of rewriting rules; $V_G = V_N \cup V_T$.

For other notions and notations in formal language theory, the reader is referred, for instance, to [12].

A *parallel communicating grammar system* (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, $G_i = (V_{N,i}, V_{T,i}, S_i, P_i)$, $1 \leq i \leq n$, such that $V_{T,i} \cap V_{N,j} = \emptyset$, $1 \leq i, j \leq n$ and there is a set $K \subseteq \{Q_1, Q_2, \ldots, Q_n\}$, of special symbols (called *query symbols*), $K \subseteq \bigcup_{i=1}^{n} V_{N,i}$, used in derivations as follows.

For $(x_1, x_2, \ldots, x_n)$, $(y_1, y_2, \ldots, y_n)$, $x_i, y_i \in V_{G_i}^*$, $1 \leq i \leq n$, we write $(x_1, x_2, \ldots, x_n) \Longrightarrow (y_1, y_2, \ldots, y_n)$ if one of the next two cases holds:

(i) $|x_i|_K = 0$, $1 \leq i \leq n$, and for each i, $1 \leq i \leq n$, we have $x_i \Longrightarrow y_i$ in the grammar G_i or $x_i \in V_{T,i}^*$, $x_i = y_i$;

(ii) If $|x_i|_K > 0$ for some i, $1 \leq i \leq n$, then for each such i we write $x_i = z_1 Q_{i_1} z_2 Q_{i_2} \cdots z_t Q_{i_t} z_{t+1}$, $t \geq 1$, $|z_j|_K = 0$, for $1 \leq j \leq t+1$; if $|x_{i_j}|_K = 0$, $1 \leq j \leq t$, then $y_i = z_1 x_{i_1} z_2 x_{i_2} \cdots x_{i_t} z_{t+1}$ and $y_{i_j} = S_{i_j}$, $1 \leq j \leq t$; when, for some j, $1 \leq j \leq t$, $|x_{i_j}|_K > 0$, then $y_i = x_i$. For all i, $1 \leq i \leq n$, for which y_i was not defined as above, we put $y_i = x_i$.

In words, an n-tuple $(x_1, x_2, \ldots, x_n)$ directly yields $(y_1, y_2, \ldots, y_n)$ if either no query symbol appears in $x_1, x_2, \ldots, x_n$, and then we have a componentwise derivation, $x_i \Longrightarrow y_i$ in G_i for each i, $1 \leq i \leq n$, or, in the case of query symbols appearing, we perform a *communication step*, as these query symbols impose: each occurrence of Q_{i_j} in x_i is replaced by x_{i_j}, provided x_{i_j} does not contain query symbols; more exactly, a component x_i is modified only when all its occurrences of query symbols refer to strings without query symbols occurrences. After a communication operation, the communicated string x_{i_j} replaces the query symbol Q_{i_j} whereas the grammar G_{i_j} resumes working from its axiom. The communication has priority over the effective rewriting. If some query symbols are not satisfied at a given communication step, then they will be satisfied at the next one (provided they ask for strings without query symbols in that moment) and so on. No rewriting is possible when at least a query symbol is present. This implies that when a circular query appears, the work of the system is blocked. Similarly, the derivation is blocked when no query symbol appears but some nonterminal component x_i cannot be further rewritten in G_i.

The language generated by γ is

$$L(\gamma) = \left\{ x \in V_{T,1}^* \mid (S_1, S_2, \ldots, S_n) \overset{*}{\Longrightarrow} (x, \alpha_2, \ldots, \alpha_n),\ \alpha_i \in V_{G_i}^*,\ 2 \leq i \leq n \right\}.$$

A derivation consists of repeated rewriting and communication steps, starting from $(S_1, S_2, \ldots, S_n)$; we retain in $L(\gamma)$ the string generated in this way on the first component, terminal with respect to G_1, without care about the strings generated by $G_2, \ldots, G_n$ (G_1 is the *master* grammar of the system).

A $PCGS$ as above is called *non-centralized*; when $K \cap V_{N,i} = \emptyset,\ 2 \leq i \leq n$, then γ is called a *centralized* $PCGS$ (only G_1 may ask for the strings generated by other grammars in the system).

A further classification can be considered, according to the following criterion: the $PCGS$ as above are called *returning, to the axiom*; when in point (ii) of the above definition we erase the words "and $y_{i_j} = S_{i_j},\ 1 \leq j \leq t$", then we obtain a *non-returning* $PCGS$ (after communicating a string x_{i_j} to some x_i, the grammar G_{i_j} does not return to S_{i_j}, but continues to process the current string x_{i_j}).

Four classes of $PCGS$ are obtained in this way: *RCPC, CPC, RPC, PC*, where *R* stands for *returning*, *C* for *centralized* and *PC* for *parallel communicating* grammar systems. When only systems of degree at most n are considered, we add the subscript n: $RCPC_n$, CPC_n etc. According to the type of grammars $G_1, G_2, \ldots, G_n$, a $PCGS$ can be regular, linear, context-free, λ-free etc. (We can write $RCPC\,(REG)$, $RCPC\,(CF)$, and so on, for distinguishing such classes.) Here we consider only λ-free context-free $PCGS$, hence *RCPC, CPC, RPC, PC* will refer to such systems. The family of languages generated by a class X of $PCGS$ is denoted by $\mathcal{L}(X)$.

Here are some simple *examples*, in order to clarify the above definitions and to point out the considerable generative capacity of $PCGS$.

$$\begin{aligned} \gamma_1 &= (G_1, G_2) \\ G_1 &= (\{S_1, S_2, Q_2\},\ \{a, b, c\},\ S_1,\ \{S_1 \longrightarrow a\,S_1,\ S_1 \longrightarrow a^2\,Q_2,\ S_2 \longrightarrow bc\}) \\ G_2 &= (\{S_1\},\ \{a, b\},\ S_2,\ \{S_2 \longrightarrow bS_2c\}). \end{aligned}$$

We have a centralized $PCGS$. The language generated both in the returning and the non-returning mode is

$$L(\gamma_1) = \{a^n b^n c^n \mid n \geq 2\}.$$

Indeed, let us examine a derivation in γ_1:

$$\begin{aligned} (S_1, S_2) &\overset{*}{\Longrightarrow} (a^k S_1,\ b^k S_2 c^k) \Longrightarrow (a^{k+2} Q_2,\ b^{k+1} S_2 c^{k+1}) \\ &\Longrightarrow (a^{k+2} b^{k+1} S_2 c^{k+1}, \alpha_2) \Longrightarrow (a^{k+2} b^{k+2} c^{k+2}, \alpha_2'), \qquad k \geq 0, \end{aligned}$$

with $\alpha_2 = b^{k+1}S_2c^{k+1}$, $\alpha_2' = b^{k+2}S_2c^{k+2}$ in the non-returning case, $\alpha_2 = S_2$, $\alpha_2' = b\,S_2c$ in the returning case.

Note that G_1, G_2 are linear grammars and $L(\gamma_1)$ is not a context-free language.

$$\begin{aligned}\gamma_2 &= (G_1, G_2)\\ G_1 &= (\{S_1, Q_2\}, \{a,b,c\}, S_1, \{S_1 \longrightarrow S_1,\ S_1 \longrightarrow Q_2cQ_2\})\\ G_2 &= (\{S_2\}, \{a,b\}, S_2, \{S_2 \longrightarrow a\,S_2,\ S_2 \longrightarrow b\,S_2,\ S_2 \longrightarrow a,\ S_2 \longrightarrow b\}).\end{aligned}$$

We obtain

$$(S_1, S_2) \stackrel{*}{\Longrightarrow} (S_1, y) \Longrightarrow (Q_2cQ_2,\ x) \Longrightarrow (xcx,\ z)$$

for $z \in \{S_2, x\}$. If $x \in \{a,b\}^*$, then the derivation is terminal, hence both in the returning and the non-returning case we have

$$L(\gamma_2) = \{xcx \mid x \in \{a,b\}^+\}$$

again a non-context-free language. (A similar $PCGS$ can be written for $\{(xc)^r \mid x \in \{a,b\}^+$ $r \geq 1$: replace $S_1 \longrightarrow Q_2cQ_2$ in G_1 by the rule $S_1 \longrightarrow (Q_2c)^r$.)

2. THE EFFICIENCY OF $PCGS$

Given a $PCGS$ $\gamma = (G_1, G_2, \ldots, G_n)$ as above, we can define the complexity measures Var, $Prod$, $Symb$ in the similar way as for context-free grammars [4], [5], [6]:

$$\begin{aligned}Var(\gamma) &= \sum_{i=1}^{n} \text{card}\, V_{N,i}\\ Prod(\gamma) &= \sum_{i=1}^{n} \text{card}\, P_i\\ Symb(\gamma) &= \sum_{i=1}^{n} Symb(P_i), \qquad Symb(P_i) = \sum_{r \in P_i} Symb(r), \qquad \text{and}\\ &\quad Symb(r) = |x| + 2 \quad \text{for} \quad r:\ A \longrightarrow x.\end{aligned}$$

For a complexity measure $M : X \longrightarrow \mathbf{N}$, defined for a class of generative mechanisms X, we define $M_X : \mathcal{L}(X) \longrightarrow \mathbf{N}$ by

$$M_X(L) = \inf\{M(G) \mid G \in X,\ L = L(G)\}.$$

Clearly, when $X_1 \subseteq X_2$, we have $M_{X_1}(L) \geq M_{X_2}(L)$, for all $L \in \mathcal{L}(X_1)$. Following [7], if there are languages $L \in \mathcal{L}(X_1)$ such that $M_{X_1}(L) > M_{X_2}(L)$, provided $X_1 \subset X_2$ is a proper inclusion, then we say that M is a *honest* measure. The following refinements of this notion are considered in [7]:

(i) $M_{X_1} >_1 M_{X_2}$ iff there is $L \in \mathcal{L}(X_1)$ such that $M_{X_1}(L) > M_{X_2}(L)$

(ii) $M_{X_1} >_2 M_{X_2}$ iff for every integer p there is $L \in \mathcal{L}(X_1)$ such that $M_{X_1}(L) - M_{X_2}(L) > p$ (arbitrarily large difference)

(iii) $M_{X_1} >_3 M_{X_2}$ iff there is a sequence L_n, $n \geq 1$ of languages in $\mathcal{L}(X_1)$ such that

$$\lim_{n\to\infty} \frac{M_{X_1}(L_n)}{M_{X_2}(L_n)} = \infty$$

(supra-linear difference)

(iv) $M_{X_1} >_4 M_{X_2}$ iff there is a constant p such that for any integer q there is a language $L \in \mathcal{L}(X_1)$ such that $M_{X_1}(L) > q$ and $M_{X_2}(L) \leq p$ (bounded by no mapping difference).

Clearly $>_j$ implies $>_{j-1}$ for each $j = 2, 3, 4$.

Here we are interested in comparing Var, $Prod$, $Symb$ with respect to CF, the class of context-free grammars, with $RCPC$, CPC, RPC, PC (we have the inclusions $CF \subset RCPC \subset RPC$, $CF \subset CPC \subset PC$).

Theorem 1. $Var_{CF} >_4 Var_X$, $X \in \{RCPC, RPC, CPC, PC\}$.

Proof. Let us consider the $PCGS$ $\gamma_n = (G_1, G_2)$ with

$$\begin{aligned} G_1 &= (\{S_1, Q_2\}, \{a, b\}, S_1, \\ &\quad \{S_1 \longrightarrow S_1\} \cup \{S_1 \longrightarrow Q_2^k b^k Q_2 \,|\, 1 \leq k \leq n\}) \\ G_2 &= (\{S_2\}, \{a\}, S_2, \{S_1 \longrightarrow a\,S_2, S_2 \longrightarrow a\}). \end{aligned}$$

Each derivation can contain only one communication step, hence γ_n can be viewed both as a returning and a non-returning $PCGS$, centralized or non-centralized. When using the rule $S_1 \longrightarrow Q_2^k b^k Q_2$, the string generated in G_2 must be a terminal one (G_1 cannot rewrite the symbol S_2); moreover, that string is of arbitrary length. Therefore,

$$L(\gamma_n) = \bigcup_{k=1}^{n} \{a^{ki} b^k a^i \,|\, i \geq 1\}$$

and we have $Var_X(L(\gamma_n)) \leq 3$ (and $Prod_X(L(\gamma_n)) \leq n+3$), $X \in \{RCPC, RPC, CPC, PC\}$.

Consider now a reduced context-free grammar $G = (V_N, V_T, S, P)$ generating $L(\gamma_n)$ and suppose there is a symbol $A \in V_N$ such that $A \overset{*}{\Longrightarrow} uAv$, $uv \neq \lambda$, in G. None of u, v can contain the symbol b (otherwise strings with arbitrarily many occurrences of b can be produced). If $A \overset{*}{\Longrightarrow} w$, $w \in \{a\}^*$, then $uwv \in \{a\}^*$, hence this is a substring of the prefix $a^{ki}b$ or of the suffix ba^i of some string $a^{ki}b^k a^i$ in $L(\gamma_n)$. But $u^r w v^r$ is such a substring too, for all $r \geq 1$. If $a^{ki}b^k a^i = xuwvyb^k a^i$, then, for $r > ni$, $|xu^r wv^r y| > ni$, hence $xu^r wv^r yb^k a^i \notin L(\gamma_n)$. If $a^{ki}b^k a^i = a^{ki}b^k xuwvy$, then, for $r > ki$, $|xu^r wv^r y| > ki$, hence $a^{ki}b^k xu^r wv^r y \notin L(\gamma_n)$. Consequently, $w = a^r b^k a^s$ for all such derivations $A \stackrel{*}{\Longrightarrow} uAv \stackrel{*}{\Longrightarrow} uwv$. Assume $u = a^p$, $v = a^q$ and consider a derivation $S \stackrel{*}{\Longrightarrow} a^g A a^h \stackrel{*}{\Longrightarrow} a^g a^{pi} A a^{qi} a^h \stackrel{*}{\Longrightarrow} a^g a^{pi} a^r b^k a^s a^{qi} a^h$ for an arbitrary $i \geq 1$. We must have $g + pi + r = k(s + qi + h)$, hence $p = kq$ and the derivation $A \stackrel{*}{\Longrightarrow} uAv \stackrel{*}{\Longrightarrow} uwv$ is of the form $A_k \stackrel{*}{\Longrightarrow} a^{kq} a^r b^k a^s a^q$. As each set $\{a^{ki} b^k a^i \mid i \geq 1\}$ is infinite, when generating it we have to use recursive derivations, hence a nonterminal A_k and a derivation as above there exists in G. Suppose now that $A_k = A_{k'}$, for $k \neq k'$, $1 \leq k$, $k' \leq n$. We can obtain a derivation

$$\begin{aligned} S &\stackrel{*}{\Longrightarrow} a^{t_1} A_k a^{t_2} \stackrel{*}{\Longrightarrow} a^{t_1} a^{kqr} A_k a^{qr} a^{t_2} \\ &\stackrel{*}{\Longrightarrow} a^{t_1} a^{kqr} a^{k'q's} A_k a^{q's} a^{qr} a^{t_2} \\ &\stackrel{*}{\Longrightarrow} a^{t_1} a^{kqr} a^{k'q's} a^{t_3} b^{k'} a^{t_4} a^{q's} a^{qr} a^{t_2} \end{aligned}$$

for arbitrary r, s. Therefore, $t_1 + kqr + k'q's + t_3 = k'(t_4 + q's + qr + t_2)$, for arbitrary r, s, which implies $kqr + k'q's = k'(q's + qr)$. However, this leads to $k = k'$, contradiction.

For each k, $1 \leq k \leq n$, we have a distinct A_k as above, therefore $Var(G) \geq n + 1$ (no one of A_k can be the axiom of G), $Var_{CF}(L(\gamma_n)) \geq n + 1$, and the proof is over. □

Corollary. $Prod_{CF} >_2 Prod_X$, $Symb_{CF} >_1 Symb_X$, X as above.

Proof. In the above proof we obtain $Prod(G) \geq 3n$: we need a derivation $S \stackrel{*}{\Longrightarrow} xA_k y$, one $A_k \stackrel{*}{\Longrightarrow} uA_k v$, and a terminal one, $A_k \stackrel{*}{\Longrightarrow} w$, each of them involving at least a rule, for each k, $1 \leq k \leq n$. Consequently, $Prod_{CF}(L(\gamma_n)) \geq 3n$, hence $Prod_{CF} >_2 Prod_X$ (as we have pointed out, $Prod_X(L(\gamma_n)) \leq n + 3$).

In the case $n = 2$, the above $PCGS$ γ_2 has $Symb(\gamma_2) = 22$, hence $Symb_X(L(\gamma_2)) \leq 22$. However, as it easily follows from the previous proof, a context-free grammar G for $L(\gamma_2)$ must contain at least six rules, of the forms $S \longrightarrow x_1 A_2 y_1$, $S \longrightarrow x_2 A_2 y_2$, $A_1 \longrightarrow a^i A_1 a^i$, $i \geq 1$, $A_2 \longrightarrow a^{2i} A_2 a^i$, $i \geq 1$, $A_1 \longrightarrow u_1 b v_1$, $A_2 \longrightarrow u_2 b^2 v_2$. Consequently, $Symb(G) \geq 24$, that is $Symb_{CF} >_1 Symb_X$, X as above. □

For $Prod$ we can find a stronger result.

Theorem 2. $Prod_{CF} >_4 Prod_X$, $X \in \{RCPC, RPC\}$.

Proof. In [1] it is proved that $Prod_{CF}(L_n) \geq \log_2(n+1)$ for $L_n = \{a^i b a^j \mid i + j \leq n - 1\}$. However, $Prod_X(L_n) \leq 11$ for all n, as L_n is generated by the $PCGS$ $\gamma = (G_1, G_2, G_3)$, with

$$\begin{aligned} G_1 &= (\{S_1, T, Q_2\}, \{a, b\}, S, \{S_1 \longrightarrow b, S_1 \longrightarrow ab, S_1 \longrightarrow ba, \\ &\qquad S_1 \longrightarrow S_1, S_1 \longrightarrow Q_2 T, T \longrightarrow T, T \longrightarrow bQ_2\}) \\ G_2 &= (\{S_2\}, \{a\}, S_2, \{S_1 \longrightarrow aS_2, S_2 \longrightarrow a\}) \\ G_3 &= (\{S_3, A, B\}, \{a\}, S_3, \{S_3 \longrightarrow A^{n-2}, A \longrightarrow B\}). \end{aligned}$$

Excepting the one-step derivations $S_1 \Longrightarrow x$, $x \in \{a, ab, ba\}$, all derivations in G_1 are of the form $S_1 \stackrel{*}{\Longrightarrow} S_1 \Longrightarrow Q_2 T \stackrel{*}{\Longrightarrow} Q_2 T \Longrightarrow Q_2 b Q_2$. As G_1 cannot rewrite S_2, the communicated strings must be of the form a^i, a^j, hence one generates strings of the form a^iba^j. However, the derivations in G_3 can have at most $n-1$ derivations steps, hence also G_2 can perform at most $n-1$ derivation steps, which implies $i+j \leq n-1$, that is $L(\gamma) = L_n$, which completes the proof. □

For the non-returning case, also the relation for *Symb* can be (slightly) improved.

Theorem 3. $Prod_{CF} >_4 Prod_X$, $Symb_{CF} >_2 Symb_X$, $X \in \{CPC, PC\}$.

Proof. We consider the *PCGS* $\gamma_n = (G_1, G_2, G_3)$, with

$$\begin{aligned} G_1 &= (\{S_1, D, Q_2, Q_3\}, \{a, b\}, S_1, \{S_1 \longrightarrow S_1, S_1 \longrightarrow DQ_3, D \longrightarrow Q_2D, \\ &\quad D \longrightarrow Q_2bQ_2, C \longrightarrow b\}) \\ G_2 &= (\{S_2\}, \{a\}, S_2, \{S_1 \longrightarrow aS_2, S_2 \longrightarrow a\}) \\ G_3 &= (\{S_3, B, C, E\}, \{a\}, S_3, \{S_3 \longrightarrow S_3, S_3 \longrightarrow C^n, C \longrightarrow B, B \longrightarrow E\}). \end{aligned}$$

Each derivation in G_1 starts by $S_1 \overset{*}{\Longrightarrow} S_1 \Longrightarrow DQ_3$. As G_1 cannot rewrite the symbols S_3, B, E, in the moment of introducing DQ_3 in G_1 we must introduce C^n in G_3 too. Thus we have $(S_1, S_2, S_3) \overset{*}{\Longrightarrow} (S_1, \alpha_2, S_3) \Longrightarrow (DQ_3, \alpha_2', C^n)$, α_2, $\alpha_2' \in \{a^i, a^iS_2 \mid i \geq 1\}$. Now, in G_3 we can use at most n times the rule $C \longrightarrow B$ and at most n times the rule $B \longrightarrow E$, therefore the derivation will have at most $2n$ further rewriting steps. In G_1, each C must be replaced by b (n rewriting steps); thus at most n steps can be performed using the rules $D \longrightarrow Q_2D$ and $D \longrightarrow Q_2bQ_2$. At the first use of the rule $D \longrightarrow Q_2D$, the string α_2' generated in G_2 must be terminal (G_1 cannot rewrite S_2), that is of the form a^i. Consequently, all subsequent symbols Q_2 will be replaced by the same string a^i. In conclusion,

$$L(\gamma_n) = \bigcup_{k=1}^{n} \{a^{ki}ba^ib^n \mid i \geq 1\}$$

hence $Var_X(L(\gamma_n)) \leq 9$, $Prod_x(L(\gamma_n)) \leq 11$, $Symb_X(L(\gamma_n)) \leq n + 37$.

Consider now a context-free grammar for $L(\gamma_n)$. As in the proof of Theorem 1, we can find that a derivation $A_k \overset{*}{\Longrightarrow} a^{kq}A_ka^q$ there is for each k, that is $Var_{CF}(L(\gamma_n)) \geq n+1$, $Prod_{CF}(L(\gamma_n)) \geq 3n$, $Symb_{CF}(L(\gamma_n)) \geq 9n$, and the proof is over. □

Open problem. Improve the above results for the measure *Symb*.

3. A SPECIFIC MEASURE

The above measures are borrowed from context-free grammars area; we consider now a specific complexity measure for *PCGS*, which can be interpreted as a dynamical one, as it refers to derivations, not to the "hardware" of a system.

Consider a $PCGS$ $\gamma = (G_1, G_2, \ldots, G_n)$ and a derivation $D: (S_1, S_2, \ldots, S_n) \Longrightarrow (w_{1,1}, w_{1,2}, \ldots, w_{1,n}) \Longrightarrow (w_{2,1}, w_{2,2}, \ldots, w_{2,n}) \cdots \Longrightarrow (w_{k,1}, w_{k,2}, \ldots, w_{k,n})$ in γ. Denote

$$Com\,(w_{i,1}, \ldots, w_{i,n}) = \sum_{j=1}^{n} |w_{i,j}|_K$$

$$Com\,(D) = \sum_{i=1}^{k} Com\,(w_{i,1}, \ldots, w_{i,n}).$$

For $x \in L(\gamma)$ define

$$Com\,(x, \gamma) = \min \left\{ Com\,(D) \mid D: (S_1, \ldots, S_n) \stackrel{*}{\Longrightarrow} (x, \alpha_2, \ldots, \alpha_n) \right\}.$$

Then

$$Com\,(\gamma) = \sup \{ Com\,(x, \gamma) \mid x \in L(\gamma) \}$$

and, for a language L and a class X of $PCGS$,

$$Com_X(L) = \inf \{ Com\,(\gamma) \mid L = L(\gamma),\ \gamma \in X \}.$$

In what follows, we consider only centralized $PCGS$ returning to axiom after each communication, hence we do not specify the class X of $PCGS$ (it is always $RCPC$).

The parameter Com evaluates the number of query symbols appearing in a derivation (a sort of cost of producing a string in γ).

A measure $M: \mathcal{L}(X) \longrightarrow \mathbf{N}$ is called *connected* if for each $n \geq n_0$, n_0 a given constant, there is $L_n \in \mathcal{L}(X)$ such that $M(L_n) = n$ (cf. [6]).

Theorem 4. *Com is a connected measure.*

Proof. Consider the languages

$$L_n = \left\{ b \left(a^i b a^i\right)^{2n+1} b \mid i \geq 1 \right\}, \qquad \text{for } n \geq 1.$$

They can be generated by the $PCGS$ $\gamma_n = (G_1, G_2)$, with

$$\begin{aligned} G_1 &= (\{S_1, S_1', S_2', Q_2\}, \{a, b\}, S_1, \{S_1 \longrightarrow bS_1'b, \\ &\quad S_1' \longrightarrow aS_1'a,\ S_1' \longrightarrow a(bQ_2)^n ba,\ S_2' \longrightarrow b\}) \\ G_2 &= (\{S_2, S_2'\}, \{a\}, S_2, \{S_1 \longrightarrow S_2',\ S_2' \longrightarrow a^2 S_2' a^2\}). \end{aligned}$$

A derivation in γ proceeds as follows:

$$\begin{aligned} (S_1, S_2) &\Longrightarrow (bS_1'b,\ S_2') \stackrel{*}{\Longrightarrow} (ba^i S_1' a^i b,\ a^{2i} S_2' a^{2i}) \\ &\Longrightarrow (ba^{i+1}(bQ_2)^n ba^{i+1}b,\ a^{2(i+1)} S_2' a^{2(i+1)}) \\ &\Longrightarrow_{c} \left(ba^{i+1} \left(ba^{2(i+1)} S_2' a^{2(i+1)}\right)^n ba^{i+1}b,\ S_2\right) \\ &\stackrel{*}{\Longrightarrow} \left(ba^{i+1} \left(ba^{2(i+1)} b a^{2(i+1)}\right)^n ba^{i+1}b,\ a^{2(n-1)} S_2 a^{2(n-1)}\right); \end{aligned}$$

hence $L(\gamma_n) = L_n$ indeed, and consequently $Com\,(L_n) \leq n$.

Consider now a $PCGS$ $\gamma = (G_1, G_2, \ldots, G_m)$ generating this language. Each string in L_n contains $2n+3$ occurrences of the symbol b, hence $2n+2$ substrings of the form a^i, a^{2i} bounded by such symbols. Each G_i is a context-free grammar, hence cannot generate strings of the form $x_1ba^ibx_2ba^ibx_3ba^ibx_4$ for arbitrarily many i. Two substrings a^i can be generated in G_1, for the other $2n$ such substrings we need communication steps. Each communication can bring to G_1 at most two substrings a^i, with arbitrarily large i. Therefore n communication steps are necessary, that is $Com\,(\gamma) \geq n$, $Com\ (L(\gamma_n)) \geq n$ hence $Com\ (L(\gamma_n)) = n$.

Clearly, the parameters Var, $Prod$, $Symb$ can be computed for an arbitrary $PCGS$ by a simple counting. The situation is different for the measure Com due to its dynamical character (it is evaluated on an infinite set, that of all terminal derivations).

Theorem 5. $Com\,(\gamma)$ and $Com\,(L(\gamma))$ cannot be algorithmically computed for an arbitrarily given (context-free, centralized and returning) $PCGS$.

Proof. In fact, a more general assertion is true, namely "the context-free-ness of $L(\gamma)$, for an arbitrarily given $PCGS$ γ, is undecidable". On the other hand, $L(\gamma)$ is context-free if and only if $Com\,(L(\gamma)) = 0$.

For, consider an arbitrary context-free grammar $G = (V_N, V_T, S, P)$, with $V_T = \{a, b\}$, and the non-context-free language

$$L = \{c^n d^m c e^m \mid m \geq n \geq 1\}$$

and construct the language

$$L' = L(G)\ \{c, d, c\}^+ \cup \{a, b\}^+ L.$$

If $L(G) = \{a, b\}^+$, then $L' = \{a, b\}^+ \{c, d, e\}^+$, hence it is a regular language. If $L(G) \neq \{a, b\}^+$, then let $w \in \{a, b\}^+ - L(G)$ be an arbitrary string. We have $L' \cap \{w\}\{c, d, e\}^+ = \{w\}L$, and this is not a context-free language. Consequently, L' is context-free (even regular) if and only if $L(G) = \{a, b\}^+$. The equality $L(G) = \{a, b\}^+$ is undecidable for arbitrary context-free grammars, hence it is undecidable whether L' is context-free or not.

On the other hand, L' is generated by the $PCGS$ $\gamma = (G_1, G_2)$, with

$$\begin{aligned}
G_1 &= (\{S_1, A, B, C, T, Q_2\} \cup V_N,\ \{a, b, c, d, e\},\ S_1, \\
&\quad \{S_1 \longrightarrow T\} \cup P \cup \{T \longrightarrow T\alpha \mid \alpha \in \{c, d, e\}\} \cup \\
&\quad \{T \longrightarrow S\alpha \mid \alpha \in \{c, d, e\}\} \cup \\
&\quad \{S_1 \longrightarrow AB\} \cup \{A \longrightarrow \alpha A \mid \alpha \in \{a, b\}\} \cup \\
&\quad \{A \longrightarrow \alpha \mid \alpha \in \{a, b\}\} \cup \\
&\quad \{B \longrightarrow cB,\ B \longrightarrow cQ_2,\ C \longrightarrow c\}) \\
G_2 &= (\{S_2, C\},\ \{d, e\},\ S_2,\ \{S_2 \longrightarrow C,\ C \longrightarrow dCe\}).
\end{aligned}$$

(Starting with the rule $S_1 \longrightarrow T$ we produce a string in $L(G)\{c,d,e\}^+$ and starting with $S_1 \longrightarrow AB$ we obtain a string in $\{a,b\}^+ L$.) Consequently, $Com\,(L(\gamma)) = 0$ if and only if $L(\gamma)$ is regular, which is undecidable.

Moreover, let us remark that when $L(G) = \{a,b\}^+$, then the derivations starting with $S_1 \longrightarrow T$ produce all strings in $L(\gamma)$, without involving communications. When $L(G) \neq \{a,b\}^+$, as the language $L(\gamma)$ is not context-free, at least a communication step is done. In conclusion, $Com\,(\gamma) = 0$ if and only if $L(G) = \{a,b\}^+$, hence also the equality $Com\,(\gamma) = 0$ is undecidable. □

Corollary. It is not decidable whether $Com\,(\gamma) = Com\,(L(\gamma))$, for an arbitrarily given $PCGS$ γ.

Proof. For the above considered language L', construct the $PCGS$ $\gamma = (G_1, G_2, G_3)$, with

$$\begin{aligned}
G_1 &= (\{S_1, A, B, C, T, Q_2, Q_3\} \cup V_N,\ \{a,b,c,d,e\},\ S_1,\\
&\quad \{S_1 \longrightarrow ST,\ T \longrightarrow Q_3\} \cup \{T \longrightarrow \alpha T \mid \alpha \in \{c,d,e\}\} \cup P \cup\\
&\quad \{S_1 \longrightarrow AB,\ B \longrightarrow cB,\ B \longrightarrow cQ_2,\ C \longrightarrow c\} \cup\\
&\quad \{A \longrightarrow \alpha A \mid \alpha \in \{a,b\}\} \cup \{A \longrightarrow \alpha \mid \alpha \in \{a,b\}\})\\
G_2 &= (\{S_2, C\},\ \{d,e\},\ S_2,\ \{S_2 \longrightarrow C,\ C \longrightarrow dCe\})\\
G_3 &= (\{S_3\},\ \{c,d,e\},\ S_3,\ \{S_3 \longrightarrow \alpha \mid \alpha \in \{c,d,e\}\}).
\end{aligned}$$

As it easily can be seen, $L(\gamma) = L'$ and each derivation in γ must use either the rule $B \longrightarrow cQ_2$ or the rule $T \longrightarrow Q_3$, hence $Com\,(\gamma) = 1$. On the other hand, $Com\,(L(\gamma)) = 0$ or $Com\,L(\gamma)) = 1$, depending on the equality $L(G) = \{a,b\}^+$, which is undecidable.

Consider now the *compatibility* question [6]: given a measure $M : X \longrightarrow \mathbf{N}$ and a language $L \in \mathcal{L}(X)$, denote

$$M^{-1}(L) = \{G \in X \mid M(G) = M(L), L = L(G)\}$$

(the set of minimal generative mechanisms for L, with respect to M). Two measures M_1, M_2 are said to be *incompatible* if there is a language L such that

$$M_1^{-1}(L) \cap M_2^{-1}(L) = \emptyset$$

(they cannot be simultaneously minimized for at least one language). □

Theorem 6. The measure Com is incompatible with each of Var, $Prod$, $Symb$.

Proof. Consider the language

$$L = \{a^n b^n c b^n c b^n c a^n \mid n \geq 1\}.$$

It can be generated by the $PCGS$ $\gamma = (G_1, G_2, G_3)$, with

$$\begin{aligned} G_1 &= (\{S_1, S_3, S_3, Q_2, Q_3\}, \{a, b, c\}, S_1, \\ &\quad \{S_1 \longrightarrow aS_1a,\ S_1 \longrightarrow aQ_2cQ_3a,\ S_2 \longrightarrow c,\ S_3 \longrightarrow c\}) \\ G_2 &= (\{S_2\}, \{b\}, S_2, \{S_2 \longrightarrow bS_2b\}) \\ G_3 &= (\{S_3\}, \{b\}, S_3, \{S_3 \longrightarrow bS_3\}). \end{aligned}$$

Consequently, $Com(L) \leq 2$.

Consider a $PCGS$ γ such that $L = L(\gamma)$, $Com(\gamma) \leq 2$. Suppose $\gamma = (G_1, G_2)$. Each of G_1, G_2 is context-free and each string in L contains five substrings a^n, b^n with related lengths. This implies $Com(\gamma) \geq 2$. If two communications are performed from G_2 to G_1, then they must be allowed to bring to G_1 strings of the same form (after a communication, the grammar G_2 resumes working from S_2). However, we cannot distinguish in $a^nb^ncb^ncb^nca^n$ two substrings, both of the form a^n or of the form b^nc or cb^n and so on, such that the string obtained by removing them to can be generated in the context-free grammar G_1. In conclusion, either $Com(\gamma) \geq 3$, or γ is of degree at least 3, contradiction.

As we assumed $Com(\gamma) \leq 2$, we have γ of degree at least 3. However, this implies $Var(\gamma) \geq 5$ (we have to use at least S_1, S_2, S_3, Q_2, Q_3), $Prod(\gamma) \geq 5$ (each G_i contains at least a rule, whereas G_1 must contain a terminal rule, one introducing Q_2, Q_3 and a recursive one, which is different from the above two), and $Symb(\gamma) \geq 19$ (in each G_i we have a nonterminal rule, also introducing a symbol a, b – we obtain $Symb \geq 12$ for them – but also c must be introduced by a non-recursive rule, as well as Q_2, Q_3 – two further rules, with $Symb \geq 7$).

On the other hand, $Var(L) \leq 4$, $Prod(L) \leq 4$, $Symb(L) \leq 17$, as L can be generated by the $PCGS$ $\gamma' = (G_1, G_2)$, with

$$\begin{aligned} G_1 &= (\{S_1, S_2, Q_2\}, \{a, b, c\}, S_1, \\ &\quad \{S_1 \longrightarrow aS_1a,\ S_1 \longrightarrow Q_2Q_2Q_2,\ S_2 \longrightarrow c\}) \\ G_2 &= (\{S_2\}, S_2, \{S_2 \longrightarrow bS_2\}) \end{aligned}$$

having $Com(\gamma') = 3$.

4. FINAL REMARKS

Of course, the complexity of $PCGS$ must be more investigated, both considering for them measures used for context-free grammars (grammatical level, index etc. [6]) and defining specific measures. For instance, a natural idea is to consider the number of simultaneously used query symbols: for a derivation D as in the beginning of Section 3, define

$$SCom(w_{i,1}, \ldots, w_{i,n}) = \max\left\{|x_{i,j}|_K : 1 \leq j \leq n\right\}$$

and then define $SCom(D)$, $SCom(x,\gamma)$, $SCom(\gamma)$, $SCom(L)$ as for Com. Similar results as for Com are expected also for this measure. Other such measures can be the maximum length of a communicated string, the degree of non-centralization (the number of grammars introducing query symbols) and so on.

As we already said, the $PCGS$ area seems to be both "practically" motivated and rich in theoretical problems.

(Received October 10, 1990.)

REFERENCES

[1] W. Bucher, K. Čulík II., H. A. Maurer and D. Wotschke: Concise description of finite languages. Theoret. Comput. Sci. *14* (1981), 227 – 246.

[2] E. Csuhaj-Varju and J. Dassow: On cooperating/distributed grammar systems. J. Inform. Process. Cybernet. (EIK) *26* (1990), 49 – 63.

[3] S. Ginsburg: The Mathematical Theory of Context-Free Languages. McGraw-Hill, New York 1966.

[4] J. Gruska: On a classification of context-free languages. Kybernetika *3* (1967), 22 – 29.

[5] J. Gruska: Some classifications of context-free languages. Inform. and Control *14* (1969), 152 – 179.

[6] J. Gruska: Descriptional complexity of context-free languages. In: Proc. Math. Found. Comp. Sci. Symp., High Tatras, 1973, pp. 71 – 84.

[7] Gh. Păun: On the fidelity of some measures of syntactic complexity. Rev. Roum. Lingv. *24* (1987), 77 – 86.

[8] Gh. Păun: Parallel communicating grammar systems: the context-free case. Found. Control Engrg. *14* (1989), 39 – 50.

[9] Gh. Păun: Non-centralized parallel communicating grammar systems. Bulletion of the EATCS *40* (1989), 257 – 264.

[10] Gh. Păun: On the power of synchronization in parallel communicating grammar systems. Stud. Cerc. Matem. *41* (1989), 191 – 197.

[11] Gh. Păun and L. Sântean: Parallel communicating grammar systems: the regular case. Ann. Univ. Buc., Ser. Matem.-Inform. *37* (1989), 55 – 63.

[12] A. Salomaa: Formal Languages. Academic Press, New York – London 1973.

[13] L. Sântean: Parallel communicating grammar systems. Bulletion of the EATCS (The Formal Language Theory Column) *42* (1990), 160 – 171.

[14] L. Sântean and J. Kari: The impact of the number of cooperating grammars on the generative power. Theoret. Comput. Sci. (to appear).

Dr. Gheorghe Păun, Institute of Mathematics, Str. Academiei 14, 70109 Bucuresti. Romania.