Intern. J. Computer Math., Vol. 34, pp. 187–203
Reprints available directly from the publisher
Photocopying permitted by license only

FURTHER REMARKS ON PARALLEL COMMUNICATING GRAMMAR SYSTEMS

GHEORGHE PĂUN and LILA SÂNTEAN

Institute of Mathematics, Str. Academiei 14, Bucureşti, Romania

(*Received 19 December 1989; in final form 20 January 1990*)

We continue the study of parallel communicating grammar systems introduced in Păun and Sântean [7] as a grammatical model of parallel computing. The investigated topics are: closure properties, the efficiency of generating a (linear) language by such a system compared with usual grammars, hierarchies.

KEY WORDS: Context-free grammars, parallel computing, grammar systems

C.R. CATEGORIES: F.4.3, G.2.1

1. INTRODUCTION

The parallel communicating grammar systems (PCGS, for short) have been introduced in Păun and Sântean [7] and investigated, from various points of view, in Păun [4, 5, 6]. They prove to be a grammatical model of parallel computing with quite interesting properties and raising many appealing mathematical problems. As architecture, they are similar to the cooperating/distributed grammar systems in Csuhaj-Varju and Dassow [1] and Dassow and Păun [2], but there the grammars work sequentially, not in parallel as in a PCGS.

The theory of PCGS (of grammar systems, in general) is very young, therefore many questions are still unsettled in this area. We consider here three of them: closure properties, hierarchies, efficiency.

2. DEFINITIONS

We assume the reader familiar with basic facts in formal language theory (from Salomaa [8], for instance), and we specify only some notations and the definitions related to PCGS.

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

We denote by REG, LIN, CF, CS the classes of regular, linear, context-free, context sensitive grammars, respectively, without using λ-rules. For a class X of grammars as above, X_λ is the class of arbitrary grammars of type X (that is using λ-rules) and $\mathscr{L}(X)$ is the family of languages generated by grammars in X.

A *PCGS* (of degree n, $n \geqq 1$) is a system

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

where $G_i = (V_{N,i}, V_{T,i}, S_i, P_i)$, $1 \leqq i \leqq n$, are Chomsky grammars such that $V_{T,i} \subseteq V_{T,1}$, $2 \leqq i \leqq n$, and there is a set $K \subseteq \{Q_1, Q_2, \ldots, Q_n\}$, of special symbols (*query* symbols), $K \subseteq \bigcup_{i=1}^{n} V_{N,i}$, used in derivations as follows.

For $(x_1, \ldots, x_n)$, $(y_1, \ldots, y_n)$, $x_i, y_i \in V_{G_i}^*$, $1 \leqq i \leqq n$ $(V_{G_i} = V_{N,i} \cup V_{T,i})$, we write $(x_1, \ldots, x_n) \Rightarrow (y_1, \ldots, y_n)$ if one of the next two cases holds:

i) $|x_i|_K = 0$ for all i, $1 \leqq i \leqq n$, and $x_i \Rightarrow y_i$ in the grammar G_i, or $x_i \in V_{T,i}^*$, $x_i = y_i$, $1 \leqq i \leqq n$;

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

In words, an n-tuple $(x_1, \ldots, x_n)$ directly yields $(y_1, \ldots, y_n)$ if either no query symbol appears in $x_1, \ldots, x_n$, and then we have a componentwise derivation, $x_i \Rightarrow y_i$ in G_i, for each i, $1 \leqq i \leqq n$, 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}, providing 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 rewriting. If some query symbols are not satisfied at a given communication step, then they will be satisfied at the next one (providing 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

$$L(\gamma) = \{x \in V_{T,1}^* \mid (S_1, \ldots, S_n) \stackrel{*}{\Rightarrow} (x, \alpha_2, \ldots, \alpha_n), \alpha_i \in V_{G_i}^*, 2 \leqq i \leqq n\}.$$

A derivation consists of repeated rewriting and communication steps, starting from $(S_1, \ldots, S_n)$; 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 $K \cap V_{N,i} = \emptyset$, $2 \leqq i \leqq n$, then γ is called *centralized* (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 previous definition we erase the words "and $y_{i_j} = S_{i_j}$, $1 \leqq i \leqq t$", 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}).

In this paper we deal with returning *PCGS*. We shall denote by *PC* their class and by *CPC* the class of centralized ones. When only systems of degree at most n are considered, we add the subscript n (PC_n, CPC_n). According to the type of grammars $G_1, \ldots, G_n$, a *PCGS* can be regular, linear, context-free, λ-free etc. We can write $PC(REG)$, $CPC(CF)$ and so on, for distinguishing such classes.

Here is a simple example: $\gamma = (G_1, G_2, G_3)$, with

$$G_1 = (\{S_1, S'_1, S_2, S_3, Q_2, Q_3\}, \{a, b, c\}, S_1,$$

$$\{S_1 \to abc, S_1 \to a^2b^2c^2, S_1 \to a^3b^3c^3,$$

$$S_1 \to aS'_1, S'_1 \to aS'_1, S'_1 \to a^3Q_2,$$

$$S_2 \to b^2Q_3, S_3 \to c\})$$

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

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

This is a regular centralized *PCGS* of degree 3 and it is easy to see that (both in the returning and the non-returning mode of derivation) we have

$$L(\gamma) = \{a^n b^n c^n \mid n \geqq 1\}$$

which is a non-context-free language.

3. ON HIERARCHIES

In Păun [5] and Păun and Sântean [7] only the power of centralized *PCGS* is investigated. Clearly $CPC_n(X) \subseteq PC_n(X)$, hence $\mathscr{L}(CPC_n(X)) \subseteq \mathscr{L}(PC_n(X))$, for all $n \geqq 1$, X as above, and it is expected that these inclusions are proper. We shall prove here that this is the case for regular *PCGS*. For other classes of grammars the problem remains open.

THEOREM 1 $\mathscr{L}(PC_3(REG)) - \mathscr{L}(CPC(REG)) \neq \varnothing$ *(hence all inclusions* $\mathscr{L}(CPC_n(REG)) \subset \mathscr{L}(PC_n(REG))$, $\mathscr{L}(CPC(REG)) \subset \mathscr{L}(PC(REG))$ *are proper*, $n \geqq 3$).

Proof Consider the language

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

($mi(x)$ is the mirror image of x). In Păun and Sântean [7] it is proved that $\{xc\, mi(x) \mid x \in \{a, b\}^*\}$ is not in $\mathscr{L}(CPC(REG))$; the same proof shows that L, the previous language, is not in $\mathscr{L}(CPC(REG))$.

On the other hand, this language can be generated by the non-centralized *PCGS* $\gamma=(G_1, G_2, G_3)$ with

$$G_1=(\{S_1, S_2, S_3, B_2, B_3, C, Z, Q_2, Q_3\}, \{a, b, c, d\}, S_1,$$

$$\{S_1\to cC, S_1\to Q_2, S_1\to Q_3, C\to aB_2, C\to bB_3,$$

$$S_2\to Z, S_3\to Z, C\to d, B_2\to Z, B_3\to Z\})$$

$$G_2=(\{S_2, S_3, B_2, B_3, C, Z, Q_1\}, \{a, b, c\}, S_2,$$

$$\{S_2\to S_2, S_2\to aQ_1, B_2\to C, B_3\to Z, S_3\to Z, C\to Z\})$$

$$G_3=(\{S_2, S_3, B_2, B_3, C, Z, Q_1\}, \{a, b, c\}, S_3,$$

$$\{S_3\to S_3, S_3\to bQ_1, B_3\to C, B_2\to Z, S_2\to Z, C\to Z\}).$$

As one can see, Z is a trap-symbol; after introducing it, at most communication steps can be performed and then the derivation is blocked.

The first step of a derivation is of the form $(S_1, S_2, S_3)\Rightarrow(\alpha_1, \alpha_2, \alpha_3)$, $\alpha_1\in\{cC, Q_2, Q_3\}$, $\alpha_2\in\{S_2, aQ_1\}, \alpha_3\in\{S_3, bQ_1\}$. If $\alpha_1=Q_2$ or $\alpha_1=Q_3$, then either the derivation is blocked by circularity when $\alpha_2=aQ_1$, or $\alpha_3=bQ_1$, or we communicate S_2, S_3 to G_1 and, at the next step, we have to use one of the rules $S_2\to Z$, $S_3\to Z$. Consequently, we must have $\alpha_1=cC$. As both G_2 and G_3 contain the rule $C\to Z$, the derivation is blocked also when one of α_2, α_3 contains Q_1. Thus, the beginning of a derivation must be of the form $(S_1, S_2, S_3)\Rightarrow(cC, S_2, S_3)\Rightarrow(c\beta_1, \beta_2, \beta_3)$, $\beta_1\in\{d, aB_2, bB_3\}$, $\beta_2\in\{S_2, aQ_1\}$, $\beta_3\in\{S_3, bQ_1\}$. If $\beta_1=d$, then the derivation ends by producing the string cd. If $\beta_1=aB_2$ (or $\beta_1=bB_3$) and $\beta_2=S_2$ ($\beta_3=S_3$, respectively), then at the next step G_1 introduces the symbol Z. If both $\beta_2=aQ_1$, $\beta_3=bQ_1$ (and, for example, $\beta_1=aB_2$), then after communicating caB_2 to G_3 we have to introduce Z in G_3. Consequently, we must have $(cC, S_2, S_3)\Rightarrow(caB_2, aQ_1, S_3)\Rightarrow(S_1, acaB_2, S_3)$ or $(cC, S_2, S_3)\Rightarrow(cbB_3, S_2, bQ_1)\Rightarrow(S_1, S_2, bcbB_3)$. We continue here the former path; the latter is similar. $(S_1, acaB_2, S_3)\Rightarrow(\delta_1, acaC, \delta_3)$, $\delta_1\in\{cC, Q_2, Q_3\}$, $\delta_3\in\{S_3, bQ_1\}$. At the next step we must have a communication, otherwise G_2 introduces the symbol Z. Therefore $\delta_1=Q_2$ and we have $(Q_2, acaC, \delta_3)\Rightarrow(acaC, S_2, \delta_3)$. If $\delta_3=bQ_1$, then C will be replaced in G_3 by Z. We started from (cC, S_2, S_3) and we obtained $(acaC, S_2, S_3)$. The process can be iterated and, after each such four steps, we add to the terminal string in G_1 a prefix σ and a suffix σ, $\sigma\in\{a, b\}$. In this way we can obtain each string of the form $xc\,mi(x)$, $x\in\{a, b\}^*$, therefore $L(\gamma)=L$ and the proof is over.

As we have said, it is highly probably that also $\mathcal{L}(CPC(CF))\subset\mathcal{L}(PC(CF))$ is a strict inclusion. For instance, we believe that the language LL, with $L=\{a^nb^nc^n \mid n\geq 1\}$, is not in $\mathcal{L}(CPC(CF))$ (for generating $a^nb^nc^na^mb^mc^m$ with unrelated arbitrarily large n, m, we have to generate "independently" $a^nb^nc^n$ and $a^mb^mc^m$, which implies we need derivations of the form $A\stackrel{*}{\Rightarrow}A$ in at least two grammars, and this can lead to parasitic derivations producing strings $a^n b^m c^p$, with different n, m, p). If this conjecture would be proved, then it would follow that $\mathscr{L}(CPC(CF))$ is not closed under concatenation. Note, in contrast, that the family $\mathscr{L}(PC(CF))$ is closed under concatenation (Theorem 5).

On the other hand, the problem whether $\mathscr{L}(CPC_n(X))$, $\mathscr{L}(PC_n(X))$, $n \geqq 1$, are infinite hierarchies, for $X \in \{REG, LIN, CF, CF_\lambda\}$, is open. (It is known that λ-rules can be eliminated from regular and linear *PCGS* (Păun [5]).) The answer of this important open question is conjectured to be affirmative. Possible languages for proving this would be those of the forms

$$\{a_1^n a_2^n \dots a_k^n \mid n \geqq 1\} \in \mathscr{L}(CPC_k(REG)) - \mathscr{L}(PC_{k-1}(REG))$$

$$\{a_1^n a_2^n \dots a_{2k}^n \mid n \geqq 1\} \in \mathscr{L}(CPC_k(LIN)) - \mathscr{L}(PC_{k-1}(CF))$$

Anyway, languages of the next form are not useful:

$$L_k = \bigcup_{i=1}^{k} \{a_i^n b_i^n \mid n \geqq 1\}.$$

Although apparently a different component is necessary for generating each substring b_i^n, the language L_k can be generated by a regular *PCGS* of degree 2. We shall obtain this as a consequence of a more general result.

Given a *PCGS* $\gamma = (G_1, \dots, G_n)$ as above and a derivation $D: (S_1, \dots, S_n) \Rightarrow (w_{1,1}, \dots, w_{1,n}) \Rightarrow (w_{2,1}, \dots, w_{2,n}) \Rightarrow \cdots \Rightarrow (w_{r,1}, \dots, w_{r,n})$ in γ, denote

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

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

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

$$Com(x, \gamma) = \min\{Com(D) \mid D: (S_1, \dots, S_n) \overset{*}{\Rightarrow} (x, \alpha_2, \dots, \alpha_n)\}$$

Then

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

The parameter *Com* has been introduced in Păun [6], as a sort of cost of producing a string in γ (the total number of query symbols appearing in the derivation).

THEOREM 2 *If γ is a linear (regular) PCGS of degree n such that $Com(\gamma) = 1$, then there is a linear (regular) PCGS γ' of degree 2 such that $L(\gamma) = L(\gamma')$.*

Proof Clearly, if $Com(\gamma)=1$, then γ is a centralized *PCGS*. Moreover, in Păun [5, Theorem 4] it is proved that $\mathscr{L}(CPC_n(REG))=\mathscr{L}(CPC_n(REG_\lambda))$, $\mathscr{L}(CPC_n(LIN))=\mathscr{L}(CPC_n(LIN_\lambda))$. Therefore, we can assume γ to be λ-free.

We shall consider here only linear *PCGS*; the regular case is a particular one.

Assume $\gamma=(G_1,\ldots,G_n)$, $G_i=(V_{n,i},V_{T,i},S_i,P_i)$ $1\leqq i\leqq n$, and construct $\gamma'=(G'_1,G'_2)$ as follows.

$$G'_1=(V'_{N,1},V_{T,1},S'_1,P'_1)$$

where

$$V'_{N,1}=V_{N,1}\cup\{S'_1\}\cup\{[A,i,0]\mid A\in(V_{N,1}\cap V_{N,i})\cup V_{T,i},2\leqq i\leqq n\}$$

$$\cup\{(\alpha,i,j)\mid \alpha\in V_{N,1}\cup\{*\},2\leqq i\leqq n,0\leqq j\leqq n-1\}$$

and P'_1 contains the next groups of rules:

1) $$S'_1\to(*,i,1),\quad 2\leqq i\leqq n,$$

$$(*,i,j)\to(*,i,j+1),\quad 2\leqq i\leqq n,1\leqq j\leqq i-2,$$

$$(*,i,i-1)\to(S_1,i,0),\quad 2\leqq i\leqq n.$$

(Each derivation in G'_1 starts by i steps when no rule in G_1 is involved, $2\leqq i\leqq n$.)

2) $$(A,i,j)\to(A,i,j+1),\quad A\in V_{N,1},\quad 2\leqq i\leqq n,\quad 0\leqq j\leqq n-2,$$

$$(A,i,n-1)\to x(B,i,0)y,\quad \text{for } A\to xBy\in P_1,\quad 2\leqq i\leqq n.$$

(At each step $i+rn$, $r\geqq 1$, of a derivation in G'_1, for given i, $2\leqq i\leqq n$, a rule of P_1 is simulated; the second component of the current nonterminal specifies the chosen i.)

3) $$(A,i,n-1)\to xQ_2y,\quad \text{for } A\to xQ_iy \text{ in } P_1.$$

(Also the queries are simulated at the moments $i+rn$, $r\geqq 1$, $2\leqq i\leqq n$.)

4) $$[A,i,0]\to A,\quad A\in V_{N,1}\cap V_{N,i},\quad 2\leqq i\leqq n,$$

$$A\to x,\quad \text{for } A\to x\in P_1,\quad |x|_K=0.$$

(After communicating a nonterminal string, the derivation can continue in G'_1 as in G_1, without further communications. As the rules $A\to x$ of P_1 without queries are introduced in P'_1, each no communication terminal derivation in G_1 can be reproduced in G'_1 too.)

5) $$[a,i,0]\to a,\quad a\in V_{T,i},\quad 2\leqq i\leqq n.$$

(When a terminal string is communicated, the derivation ends by a rule as above.)

$$G'_2=\left(V'_{N,2}, \bigcup_{i=2}^{n} V_{T,i}, S'_2, P'_2\right)$$

where

$$V'_{N,2}=\{S'_2\}\cup\{[\alpha,i,j] \mid \alpha\in\{*\}\cup V_{N,i}\cup V_{T,i},$$

$$2\leqq i\leqq n,\quad 0\leqq j\leqq n-1\}$$

and P'_2 contains the following groups of rules:

1)
$$S'_2\to[*,i,1],\quad 2\leqq i\leqq n,$$

$$[*,i,j]\to[*,i,j+1],\quad 2\leqq i\leqq n,\quad 1\leqq j\leqq i-2,$$

$$[*,i,i-1]\to[S_i,i,0],\quad 2\leqq i\leqq n.$$

(The derivations in grammars G_i, $2\leqq i\leqq n$, will be simulated in G_2, involving nonterminals having i on the second component.)

2)
$$[A,i,j]\to[A,i,j+1],\quad A\in V_{N,i},\quad 2\leqq i\leqq n,\quad 0\leqq j\leqq n-2,$$

$$[A,i,n-1]\to x[B,i,0]y,\quad \text{for } A\to xBy\in P_i,\quad 2\leqq i\leqq n.$$

(At each moment $i+rn$, $r\geqq 1$, $2\leqq i\leqq n$, in the presence of the component i, a rule of G_i is simulated in G'_2.)

3)
$$[A,i,n-1]\to x'[a,i,0],\quad \text{for } A\to x'a\in P_i,\quad x'\in V^*_{T,i},$$

$$a\in V_{T,i},\quad 2\leqq i\leqq n.$$

(For each terminal rule $A\to x'a$ in P_i—all of them are λ-free—we consider rules as above, in order to make possibly to finish only the derivations in which the queries from G'_1 at moments $i+rn$, $r\geqq 1$, $2\leqq i\leqq n$, are satisfied by derivations in G'_2 simulating derivations in G_i, that is introducing at moments $i+rn$ nonterminals of the form $[\alpha,i,0]$.)

4)
$$[a,i,j]\to[a,i,j+1],\quad a\in V_{T,i},\quad 2\leqq i\leqq n,\quad 0\leqq j\leqq n-2,$$

$$[a,i,n-1]\to[a,i,0],\quad a\in V_{T,i},\quad 2\leqq i\leqq n.$$

(Symbols $[a,i,0]$ are introduced only at moments $i+rn$, $r\geqq 1$, $2\leqq i\leqq n$.)

From the above explanations, it is easy to see that the symbols $[\alpha,i,j]$, $j\neq 0$, cannot be rewritten in G'_1, hence the only successful derivations are those which simulate in G'_2 a derivation in G_i, $2 \leqq i \leqq n$, and G'_1 asks exactly at moments $i+rn$, $r \geqq 1$, for the string generated in G'_2, thus receiving a nonterminal of the form $[\alpha, i, 0]$, $\alpha \in (V_{N,1} \cap V_{N,i}) \cup V_{T,i}$. In conclusion, $L(\gamma) = L(\gamma')$ and the proof is finished.

COROLLARY *If γ is a regular PCGS such that $Com(\gamma) = 1$, then $L(\gamma) \in \mathscr{L}(CF)$.*

Proof Theorem 3 in Păun and Sântean [7] proves that $\mathscr{L}(CPC_2(REG)) \subset \mathscr{L}(CF)$. Combining with the result of the previous theorem, we obtain the assertion in corollary.

Remark The example in Section 2 proves that the $Com(\gamma) = 1$ consideration in Theorem 1 is essential: for γ as in that example (it is centralized) we have $Com(\gamma) = 2$ and $L(\gamma)$ is not context-free.

Open problem Is the above theorem valid also for *PCGS* with context-free components? (Conjecture: no.)

4. THE EFFICIENCY OF *PCGS*

As in Păun [5] and Păun and Sântean [7] it is proved, the generative capacity of *PCGS* is much larger than that of corresponding grammars (*PCGS* with regular components can generate non-context-free languages, *PCGS* with at least three context-free components can generate non-semilinear languages etc.). Moreover, the syntactic complexity, in the sense of Gruska [3], of context-free languages can be considerably decreased (see Păun [6] for a precise meaning of this assertion). But what about dynamical complexity, about *time* parameter, for instance? Although in formal language theory the time parameter is not investigated for grammars, we can define such a measure in the following way.

Given a grammar $G = (V_N, V_T, S, P)$ and a derivation $D: S \Rightarrow w_1 \Rightarrow \cdots \Rightarrow w_n$, we put

$$Time(D) = n$$

and define, for $x \in L(G)$,

$$Time_G(x) = \min\{Time(D) \mid D: S \overset{*}{\Rightarrow} x\}$$

A mapping $Time_G: L(G) \to \mathbf{N}$ is obtained in this way. A similar definition holds for any other generative device, including a *PCGS* (both the rewriting and the communication steps are counted).

In this frame, a natural question is: given a grammar G and a mapping f, can we construct (algorithmically) a *PCGS* γ such that $L(G) = L(\gamma)$ and $Time_\gamma(x) < f(Time_G(x))$, for (almost) all strings $x \in L(G)$? Of course, such an improvement in the parameter *Time* must not increase "too much" other measures of (syntactic) complexity. Here are some such measures. For a *PCGS* γ as above, define

$$Var(\gamma) = \operatorname{card} \bigcup_{i=1}^{n} V_{N,i}$$

$$Prod(\gamma) = \text{card} \bigcup_{i=1}^{n} P_i$$

$$Symb(\gamma) = \sum_r Symb(r), \quad \text{for all } r \in \bigcup_{i=1}^{n} P_i, \quad \text{where}$$

$$Symb(A \to x) = |x| + 2,$$

$$Length(\gamma) = \max \left\{ |x| \,\middle|\, A \to x \in \bigcup_{i=1}^{n} P_i \right\}.$$

These parameters can be defined in the same way for grammars (particular case).

THEOREM 3 *Given an infinite linear language L and a linear grammar G such that $L = L(G)$, $Var(G) = p$, for each given natural number t there is a centralized PCGS γ such that*

$$L = L(\gamma)$$

$$Var(\gamma) = Var(G) + pt$$

$$Prod(\gamma) = Prod(G) + p(t+1)$$

$$Symb(\gamma) = Symb(G) + 3p(t+1)$$

$$Length(\gamma) = Length(G)$$

and, for each $x \in L(G)$, we have

$$Time_{\gamma}(x) < \frac{1}{t} Time_G(x) + 3t.$$

Proof For $G = (\{A_1, \ldots, A_p\}, V_T, A_1, P)$, we construct the *PCGS* $\gamma = (G_1, \ldots, G_{pt+1})$, where

$$G_1 = (V_{N,1}, V_T, A_1, P_1)$$

$$V_{N,1} = \{A_1, \ldots, A_p\} \cup \{Q_{ij+1} \mid 1 \leqq i \leqq p, 1 \leqq j \leqq t\}$$

$$P_1 = P \cup \{A_1 \to A_1\} \cup \{A_i \to Q_{ij+1} \mid 1 \leqq i \leqq p, 1 \leqq j \leqq t\}$$

and, for each $1 \leqq i \leqq p$, $1 \leqq j \leqq t$,

$$G_{ij+1} = (\{A_1, \ldots, A_p\}, V_T, A_i, P \cup \{A_r \to A_r \mid 1 \leqq r \leqq p\}).$$

The inclusion $L(\gamma) \subseteq L(G)$ is obvious (each grammar G_{ij+1} has A_i as axiom, $1 \leqq i \leqq p$, $1 \leqq j \leqq t$; when G_1 introduces a query symbol Q_{ij+1}, this means it used a rule $A_i \to Q_{ij+1}$ and the string which will replace Q_{ij+1} is obtained by a derivation of the form $A_i \overset{*}{\Rightarrow} w$, in the grammar G_{ij+1}; this implies the derivations on various components of γ which will be communicated to G_1 will complete a derivation in G; note that the rules $A_i \to A_i$ introduced in each G_{ij+1} in order to synchronize the derivations do not lead to derivations not in G.)

Conversely, take a string $x \in L(G)$ with $Time_G(x) = m$ and take a derivation D of x with exactly m steps. If $m \leqq 3t$, then D can be considered a derivation in γ as it can be reproduced in G_1. If $m > 3t$, then we write D in the form

$$D: A_1 \overset{*}{\Rightarrow} u_1 A_{i_1} v_1 \overset{*}{\Rightarrow} u_1 u_2 A_{i_2} v_2 v_1 \overset{*}{\Rightarrow} \cdots$$

$$\cdots \overset{*}{\Rightarrow} u_1 \ldots u_{t-1} A_{i_{t-1}} v_{t-1} \ldots v_1 \overset{*}{\Rightarrow} u_1 \ldots u_{t-1} w v_{t-1} \ldots v_1$$

such that

$$Time_G(A_1 \overset{*}{\Rightarrow} u_1 A_{i_1} v_1) = \left[\frac{m}{t}\right]$$

$$Time_G(A_{i_j} \overset{*}{\Rightarrow} u_{j+1} A_{i_{j+1}} v_{j+1}) = \left[\frac{m}{j}\right], \quad 1 \leqq j \leqq t-2$$

($[q]$ is the greatest integer smaller than x). If m is a multiple of t, then

$$Time_G(A_{i_{t-1}} \overset{*}{\Rightarrow} w) = \left[\frac{m}{t}\right].$$

If m is not a multiple of t, then we further write $A_{i_{t-1}} \overset{*}{\Rightarrow} w$ as

$$A_{i_{t-1}} \overset{*}{\Rightarrow} u_t A_{i_t} v_t \overset{*}{\Rightarrow} u_t w' v_t$$

with

$$Time_G(A_{i_{t-1}} \overset{*}{\Rightarrow} u_t A_{i_t} v_t) = \left[\frac{m}{t}\right]$$

$$Time_G(A_{i_t} \overset{*}{\Rightarrow} w') = m - \left[\frac{m}{t}\right] t.$$

Clearly, $(m - [m/t]t) < t$. We construct the following derivation in γ:

$$(A_1, A_{i_1}, A_{i_2}, \ldots, A_{i_{t-1}}) \overset{*}{\Rightarrow} (u_1 A_{i_1} v_1, u_2 A_{i_2} v_2, \ldots, u_t A_{i_t} v_t)$$

$$\Rightarrow (u_1 Q_{i_1} v_1, u_2 A_{i_2} v_2, \ldots, u_t A_{i_t} v_t)$$

$$\Rightarrow (u_1 u_2 A_{i_2} v_2 v_1, A_{i_2}, u_3 A_{i_3} v_3, \ldots, u_t A_{i_t} v_t)$$

$$\Rightarrow (u_1 u_2 Q_{i_2} v_2 v_1, A_{i_2}, u_3 A_{i_3} v_3, \ldots, u_t A_{i_t} v_t) \Rightarrow \cdots$$

$$\Rightarrow (u_1 u_2 \ldots u_t A_{i_t} v_t \ldots v_2 v_1, A_{i_2}, \ldots, A_{i_t})$$

$$\overset{*}{\Rightarrow} (u_1 u_2 \ldots u_t w' v_t \ldots v_2 v_1, A_{i_2}, \ldots, A_{i_t}).$$

When m is a multiple of t, then on the last component one generates $u_t w' v_t = w$ from $A_{i_{t-1}}$ in $[m/t]$ steps.

Besides the $[m/t]$ initial rewriting steps, in the previous derivation we have $t-1$ steps of using rules of the form $A_{i_j} \to Q_{i_j}$ on the first component and of the form $A_{i_r} \to A_{i_r}$ on the others, $t-1$ steps of the communication and, possibly, further at most $t-1$ steps in the final stage (for $A_{i_t} \overset{*}{\Rightarrow} w'$). In conclusion, we have less than $[m/t]+3t$ steps, therefore $Time_\gamma(x) < (1/t)Time_G(x)+3t$.

It is easy to see that the relations between $Var(G)$, $Prod(G)$, $Symb(G)$, $Length(G)$ and $Var(\gamma)$, $Prod(\gamma)$, $Symb(\gamma)$, $Length(\gamma)$ are as specified in theorem.

5. CLOSURE PROPERTIES

In general, it seems to be hard to say something about closure properties of families of languages generated by *PCGS*, because, on the one hand, it is not easy to prove positive results and, on the other hand, there are no known languages not in $\mathscr{L}(CPC(CF))$, $\mathscr{L}(PC(CF))$ and other such families. Here we shall investigate only the families $\mathscr{L}(PC(CF))$, $\mathscr{L}(PC(CF_\lambda))$ (more closure properties can be proved to hold for them).

First, let us note that by standard proofs, all families involving arbitrary rules are closed under arbitrary homomorphisms and all families involving λ-free rules are closed under λ-free homomorphisms.

THEOREM 4 *The families $\mathscr{L}(PC(CF))$, $\mathscr{L}(PC(CF_\lambda))$ are closed under union.*

Proof Consider two *PCGS*, $\gamma_1 = (G'_1, \ldots, G'_n)$, $\gamma_2 = (G''_1, \ldots, G''_m)$, $G'_i = (V'_{N,i}, V'_{T,i}, S'_i, P'_i)$, $1 \leqq i \leqq n$, $G''_i = (V''_{N,i}, V''_{T,i}, S''_i, P''_i)$, $1 \leqq i \leqq m$. Without loss of generality, we may assume that $V'_{T,1} \cap (\bigcup_{i=1}^{m} V''_{N,i}) = \varnothing$ and $V''_{T,1} \cap (\bigcup_{i=1}^{n} V'_{N,i}) = \varnothing$.

If both of $L(\gamma_1)$, $L(\gamma_2)$ are finite, then the assertion is trivial. If one of $L(\gamma_1)$, $L(\gamma_2)$ is finite—say $L(\gamma_2)$—then we construct the *PCGS* $\gamma = (G_1, G'_2, \ldots, G'_n)$ with

$$G_1 = (V'_{N,1} \cup \{S_1\}, V'_{T,1} \cup V''_{T,1}, S_1, P_1)$$

$$P_1 = P'_1 \cup \{S_1 \to x \mid \text{for } S'_1 \to x \in P'_1\}$$

$$\cup \{S_1 \to w \mid w \in L(\gamma_2)\}.$$

The equality $L(\gamma)=L(\gamma_1)\cup L(\gamma_2)$ is obvious.
Assume now that both $L(\gamma_1)$ and $L(\gamma_2)$ are infinite. We construct the *PCGS*

$$\gamma=(G_1, G_2, \ldots, G_{n+1}, G_{n+2}, \ldots, G_{n+m+1})$$

where

$$G_1=(\{S_1, Q_2, Q_{n+2}\}, V'_{T,1}\cup V''_{T,2}, S_1,$$
$$\{S_1\to S_1, S_1\to Q_2, S_1\to Q_{n+2}\})$$

$G_2, \ldots, G_{n+1}$ are exactly $G'_1, \ldots, G'_n$, with each occurrence of

$$Q_i, \quad 1\leqq i\leqq n, \text{ replaced by } Q_{i+1},$$

$G_{n+2}, \ldots, G_{n+m+1}$ are exactly $G''_1, \ldots, G''_m$, with each occurrence of

$$Q_i, \quad 1\leqq i\leqq m, \text{ replaced by } Q_{i+n+1}.$$

Each derivation in G_1 is of the form $S_1 \overset{*}{\Rightarrow} S_1 \Rightarrow Q_2$ or of the form $S_1 \overset{*}{\Rightarrow} S_1 \Rightarrow Q_{n+2}$. The components $G_2, \ldots, G_{n+1}$ work exactly as γ_1, the components $G_{n+2}, \ldots, G_{n+m+1}$ work as γ_2. The string in G_2 (G_{n+2}, respectively) communicated to G_1 must be in $V'^{*}_{T,1}$ (in $V''^{*}_{T,1}$, respectively). When a string in $L(\gamma_1)$ is to be prepared on the components $G_2, \ldots, G_{n+1}$ of γ, the components $G_{n+2}, \ldots, G_{n+m+1}$ work on a long enough string, in order to not block the derivation, and similarly $G_2, \ldots, G_{n+1}$ during preparing a string in $L(\gamma_2)$ (the languages $L(\gamma_1)$, $L(\gamma_2)$ are infinite). In conclusion, $L(\gamma)=L(\gamma_1)\cup L(\gamma_2)$.

THEOREM 5 *The families $\mathscr{L}(PC(CF))$, $\mathscr{L}(PC(CF_\lambda))$ are closed under concatenation.*

Proof Consider two *PCGS*, $\gamma_1=(G'_1, \ldots, G'_n)$, $\gamma_2=(G''_1, \ldots, G''_m)$ as in the previous proof. Again, if both $L(\gamma_1)$, $L(\gamma_2)$ are finite, then the question is trivial, and if exactly one of them is finite, the question can be easily settled. If both $L(\gamma_1)$, $L(\gamma_2)$ are infinite, then we construct the *PCGS*

$$\gamma=(G_1, G_2, \ldots, G_{n+1}, G_{n+2}, \ldots, G_{n+m+1}, G_{n+m+2})$$

where

$$G_1=(\{S_1, R, Q_2, Q_{n+2}\}, V'_{T,1}\cup V''_{T,1}, S_1,$$
$$\{S_1\to S_1, S_1\to Q_2R, R\to R, R\to Q_{n+2}\}),$$

$G_2, \ldots, G_{n+1}$ are exactly $G'_1, \ldots, G'_n$, with each occurrence of

$$Q_i, \quad 1\leqq i\leqq n, \text{ replaced by } Q_{i+1},$$

$G_{n+2}, \ldots, G_{n+m+1}$ are exactly $G''_1, \ldots, G''_m$, with each occurrence of

Q_i, $1 \leqq i \leqq m$, replaced by $Q_i + n + 1$,

$$G_{n+m+2} = (\{S_{n+m+2}\} \cup \{Q_i \mid 2 \leqq i \leqq n+m+1\}, \{a\}, S_{n+m+2},$$
$$\{S_{n+m+2} \to S_{n+m+2}, S_{n+m+2} \to Q_2 \ldots Q_{n+1} Q_{n+2}$$
$$\ldots Q_{n+m+1} S_{n+m+2}\}).$$

Each derivation in G_1 must use the rules $S_1 \to Q_2 R$, $R \to Q_{n+2}$. As no symbol in $V'_{N,1}$, $V''_{N,1}$ can be rewritten in G_1, the strings communicated to G_1 after introducing Q_2, Q_{n+2} must be terminal, hence in $L(\gamma_1)$, $L(\gamma_2)$. This implies $L(\gamma) \subseteq L(\gamma_1)L(\gamma_2)$.

Conversely, each pair of derivations $(S'_1, \ldots, S'_n) \stackrel{*}{\Rightarrow} (x, \alpha_2, \ldots, \alpha_n)$, $(S''_1, \ldots, S''_m) \stackrel{*}{\Rightarrow} (y, \beta_2, \ldots, \beta_m)$ can be simulated in γ as follows:

$$(S_1, \ldots, S_{n+m+2}) \stackrel{*}{\Rightarrow} (Q_2 R, x, \alpha_2, \ldots, \alpha_n, \delta''_1, \ldots, \delta''_m, Q_2 \ldots Q_{n+1} Q_{n+2} \ldots Q_{n+m+1} S_{n+m+2})$$
$$\Rightarrow (xR, S'_1, \ldots, S'_n, S''_1, \ldots, S''_m, x\alpha_2 \ldots \alpha_n \delta''_1 \ldots \delta''_m S_{n+m+2})$$
$$\stackrel{*}{\Rightarrow} (xQ_{n+2}, \delta'_1, \ldots, \delta'_n, y, \beta_2, \ldots, \beta_m, x\alpha_2 \ldots \alpha_n \delta''_1 \ldots \delta''_m S_{n+m+2})$$
$$\Rightarrow (xy, \delta'_1, \ldots, \delta'_n, S''_1, \beta_2, \ldots, \beta_m, x\alpha_2 \ldots \alpha_n \delta''_1 \ldots \delta''_m S_{n+m+2}).$$

(Remember that $L(\gamma_1)$, $L(\gamma_2)$ are infinite, hence we can find long enough derivations

$$(S'_1, \ldots, S'_n) \stackrel{*}{\Rightarrow} (\delta'_1, \ldots, \delta'_n), (S''_1, \ldots, S''_m) \stackrel{*}{\Rightarrow} (\delta''_1, \ldots, \delta''_m)$$

in $G_2, \ldots, G_n$ and in $G_{n+1}, \ldots, G_{n+m+1}$, respectively.) In conclusion, $L(\gamma_1)L(\gamma_2) \subseteq L(\gamma)$, hence $L(\gamma) = L(\gamma_1)L(\gamma_2)$.

THEOREM 6 *The families $\mathscr{L}(PC(CF))$, $\mathscr{L}(PC(CF_\lambda))$ are closed under Kleene + and *.*

Proof Consider a $PCGS$ $\gamma_0 = (G'_1, \ldots, G'_n)$, with $G'_i = (V'_{N,i}, V'_{T,i}, S'_i, P'_i)$, $1 \leqq i \leqq n$, and construct the $PCGS$

$$\gamma = (G_1, G_2, \ldots, G_{n+1}, G_{n+2}, G_{n+3})$$

where

$$G_1 = (\{S_1, S'_{n+2}, Q_2, Q_{n+2}\}, V_{T,1}, S_1,$$
$$\{S_1 \to S_1, S_1 \to Q_2 Q_{n+2}, S'_{n+2} \to S_1, S_1 \to Q_2\}),$$
$$G_2 = (V'_{N,1} \cup \{S''_2, Q_{n+1}\}, V'_{T,1}, S''_2,$$
$$\{S''_2 \to h(x) \mid S_1 \to x \in P'_1\} \cup \{A \to h(x) \mid A \to x \in P'_1\}),$$

$$G_{i+1}=(V'_{N,i}\cup\{S''_{i+1},Q_{n+1}\},V'_{T,i},S''_{i+1},$$
$$\{S''_{i+1}\to S'_i\}\cup\{A\to h(x)\mid A\to x\in P'_i\}),\quad 2\leqq i\leqq n,$$

where

$$h:\left(\bigcup_{i=1}^{n}(V'_{N,i}\cup V'_{T,i})\right)^*\to\left(\bigcup_{i=1}^{n}(V'_{N,i}\cup V'_{T,i})\cup\{Q_{n+1}\}\right)^*$$

is the homomorphism defined by $h(\alpha)=\alpha$ for $\alpha\notin K$, and $h(Q_j)=Q_{j+1}$, $1\leqq j\leqq n$;

$$G_{n+2}=(\{S_{n+2},S'_{n+2},T\},\{a\},S_{n+2},$$
$$\{S_{n+2}\to S'_{n+2},S'_{n+2}\to T,T\to T\}),$$
$$G_{n+3}=(\{S_{n+3},S'_{n+3},Q_2,\ldots,Q_{n+2}\},\{a\},S_{n+3},$$
$$\{S_{n+3}\to Q_2S'_{n+3},S'_{n+3}\to Q_3\ldots Q_{n+2}S_{n+3},$$
$$S'_{n+3}\to S'_{n+3}\}).$$

Let us examine a derivation in γ.

$$(S_1,S''_2,S''_3,\ldots,S''_{n+1},S_{n+2},S_{n+3})$$
$$\Rightarrow(\alpha_1,\alpha_2,S'_3,\ldots,S'_{n+1},S'_{n+2},Q_2S'_{n+3})$$
$$\Rightarrow(\alpha'_1,S''_2,S'_3,\ldots,S'_{n+1},\alpha_{n+2},\alpha_2S'_{n+3}).$$

If $\alpha_1=S_1$, then $\alpha'_1=S_1$, $\alpha_{n+2}=S'_{n+2}$. If $\alpha_1=Q_2$, then $\alpha'_1=\alpha_2$ and the derivation ends (correctly when $\alpha_2\in V'^*_{T,1}$). If $\alpha_1=Q_2Q_{n+2}$, then $\alpha'_1=\alpha_2S'_{n+2}$, $\alpha_{n+2}=S_{n+2}$ and we continue by

$$(\alpha_2S'_{n+2},S''_2,S'_3,\ldots,S'_{n+1},S_{n+2},\alpha_2S'_{n+3})$$
$$\Rightarrow(\alpha_2S_1,\beta_2,\beta_3,\ldots,\beta_{n+1},S'_{n+2},\alpha_2S'_{n+3}).$$

Therefore either the derivation ends, or we have a current n-tuple of the form $(xS_1,\beta_2,\beta_3,\ldots,\beta_{n+1},S'_{n+2},yS'_{n+2})$, with $(\beta_2,\ldots,\beta_{n+1})$ correctly generated in γ_0. Now, G_{n+2} introduces the symbol T, which will be repeated; it cannot be rewritten in G_1. If the next query in G_1 will be Q_2, then the string generated by G_2 (that is by G'_1 in γ_0) must be terminal and the derivation ends. If the next query in G_1 will be Q_2Q_{n+2}, then the string in G_2 must be terminal and that in G_{n+2} must be S'_{n+2} (the only nonterminal of G_{n+2} which can be rewritten in G_1). This means, we have the next steps of derivation:

$$(x'S_1,\delta_2,\delta_3,\ldots,\delta_{n+1},T,y'S'_{n+3})$$

$$\Rightarrow (x'S_1, \delta'_2, \delta'_3, \ldots, \delta'_{n+1}, T, y'Q_3 \ldots Q_{n+2}S_{n+3})$$

$$\Rightarrow (x'S_1, \delta'_2, S''_3, \ldots, S''_{n+1}, S_{n+2}, y'\delta'_3 \ldots \delta'_{n+1} T S_{n+3})$$

$$\Rightarrow (x'Q_2Q_{n+2}, \delta''_2, S'_3, \ldots, S'_{n+1}, S'_{n+2}, y'\delta'_3 \ldots \delta'_{n+1} TQ_2S'_{n+3})$$

$$\Rightarrow (x'\delta''_2 S'_{n+2}, S''_2, S'_3, \ldots, S'_{n+1}, S_{n+2}, y'\delta'_3 \ldots \delta'_{n+1} T\delta''_2 S'_{n+3}).$$

If S_1 is not replaced by Q_2 or by Q_2Q_{n+1} in the previous step of derivation, then the string δ''_2 is "lost" as being requested by G_{n+3}. Now we obtain

$$(x'\delta''_2 S_1, \pi_2, \ldots, \pi_{n+1}, S'_{n+2}, y'\delta'_3 \ldots \delta'_{n+1} T\delta''_2 S'_{n+3})$$

and the process can be iterated. When the rule $S_1 \rightarrow Q_2$ is used in G_1, the derivation ends by producing a string in $L(\gamma_0)^+$.

In order to obtain the closure under $*$, a rule $S_1 \rightarrow \lambda$ must be added to G_1.

THEOREM 7 *The families* $\mathscr{L}(PC(CF))$, $\mathscr{L}(PC(CF_\lambda))$ *are closed under substitution by* λ*-free languages.*

Proof Consider a *PCGS* $\gamma_0 = (G'_1, \ldots, G'_n)$ with $G'_i = (V'_{N,i}, V'_{T,i}, S'_i, P'_i)$, $1 \leqq i \leqq n$, $V'_{T,1} = \{a_1, \ldots, a_r\}$, and take the context-free grammars $G''_i = (V''_{N,i}, V''_{T,i}, S''_i, P''_i)$, $1 \leqq i \leqq r$. If $L(\gamma_0)$ is finite, then each context-free substitution maps it to a context-free language. Thus, we have to discuss only the case when $L(\gamma_0)$ is infinite. We construct the *PCGS*

$$\gamma = (G_1, G_2, \ldots, G_{n+1}, G_{n+2}, \ldots, G_{n+r+1}, G_{n+r+2})$$

where

$$G_1 = (\{S_1, Q_2\} \cup \{Q_{n+i+1} \mid 1 \leqq i \leqq r\} \cup \{a'_i \mid 1 \leqq i \leqq r\}, \bigcup_{i=1}^{r} V''_{T,i},$$

$$S_1, \{S_1 \rightarrow S_1, S_1 \rightarrow Q_2\} \cup \{a'_i \rightarrow a'_i, a'_i \rightarrow Q_{n+i+1} \mid 1 \leqq i \leqq r\}),$$

$$G_{i+1} = (V'_{N,i} \cup \{a' \mid a \in V'_{T,i}\}, \{a\}, S'_i,$$

$$\{A \rightarrow h(x) \mid A \rightarrow x \in P'_i\}), \quad 1 \leqq i \leqq n,$$

with

$$h: \left(\bigcup_{i=1}^{n} (V'_{N,i} \cup V'_{T,i})\right)^* \rightarrow \left(\bigcup_{i=1}^{n} V'_{N,i} \cup \{Q_{n+1}\} \cup \left\{a' \mid a \in \bigcup_{i=1}^{n} V'_{T,i}\right\}\right)^*$$

the homomorphism defined by

$$h(A)=A, A\in \bigcup_{i=1}^{n} V'_{N,i}-K,$$

$$h(a)=a', a\in \bigcup_{i=1}^{n} V'_{T,i}, h(Q_j)=Q_{j+1}, \quad 1\leqq j\leqq n,$$

$$G_{n+i+1}=(V''_{N,i}\cup\{S_{n+i+1}\}, V''_{T,i}, S_{n+i+1},$$

$$P''_i\cup\{S_{n+i+1}\to S''_i, S_{n+i+1}\to S_{n+i+1}\}), \quad 1\leqq i\leqq r,$$

$$G_{n+r+2}=(\{S_{n+r+2}, Q_{n+2},\dots, Q_{n+r+1}\}, \{a\}, S_{n+r+2},$$

$$\{S_{n+r+2}\to S_{n+r+2}, S_{n+r+2}\to Q_{n+2}\dots Q_{n+r+1}S_{n+r+2}\}).$$

It is easy to see that (1) each derivation in G_1 must use a rule $S_1\to Q_2$, (2) the components $G_2,\dots,G_{n+1}$ work exactly as γ_0, but the produced string has each symbol $a\in V'_{T,1}$ replaced by a', (3) the components G_{n+r+1}, $1\leqq i\leqq r$ produce strings in $L(G_i'')$ and (4) the last component, G_{n+r+2}, only "cleans" the strings generated by G_{n+i+1}, $1\leqq i\leqq r$, returning to axioms these components. After introducing Q_2 in G_1 and communicating the string x generated in that moment by $G_2,\dots,G_{n+1}$ (hence by γ_0, modulo the homomorphism h), the only rewritings in G_1 are done by rules $a_i'\to a_i'$ and $a_i'\to Q_{n+i+1}$, $1\leqq i\leqq r$. In this way, each a_i' is replaced by a string generated by G_i''. If this string is not terminal, the derivation is blocked. When generating the string x in $G_2,\dots,G_{n+1}$, the components $G_{n+2},\dots,G_{n+r+1}$ can be returned to axioms as many times as necessary (by queries using the rule $S_{n+r+2}\to Q_{n+2}\dots Q_{n+r+1}S_{n+r+2}$ in G_{n+r+2}), thus not limiting the length of the derivation in $G_2,\dots,G_{n+1}$. Similarly, when generating strings in $G_{n+2},\dots,G_{n+r+1}$, the components $G_2,\dots,G_{n+1}$ work to a long enough string in $L(\gamma_0)$ (this language is infinite). Moreover, after using a rule $a_i'\to Q_{n+i+1}$ in G_1 and substituting Q_{n+i+1} by the corresponding string in G_{n+i+1}, the components $G_{n+2},\dots,G_{n+r+1}$ can be again returned to axioms, in order to obtain a new string in some G_{n+j+1}, maybe also in G_{n+i+1}, which will be communicated to G_1 for substituting a symbol a_j'.

In conclusion, $L(\gamma)=s(L(\gamma_0))$, for s the substitution defined by $s(a_i)=L(G_i'')$, $1\leqq i\leqq r$; and the proof is ended.

Remark Note that the proof of Theorem 4 holds also for regular and for linear $PCGS$; the proofs of Theorems 5, 6, 7 do not hold for these cases.

Open problem Are the families $\mathscr{L}(PC(CF))$, $\mathscr{L}(PC(CF_\lambda))$ closed under intersection by regular sets? If the answer would be affirmative, this will imply that $\mathscr{L}(PC(CF_\lambda))$ is a full AFL, and, as probably $\mathscr{L}(PC(CF))$ is closed under restricted homomorphisms, this family will be an AFL.

Acknowledgement

Useful discussions with Dr. Marian Gheorghe are gratefully acknowledged.

References

[1] E. Csuhaj-Varju and J. Dassow, On cooperating/distributed grammar systems, *Journal of Inform. Processing* (E.I.K.), (1990).

[2] J. Dassow and Gh. Păun, On some variants of cooperating/distributed grammar systems, *Stud. Cerc. Matem.* **42,** 2 (1990).

[3] J. Gruska, Descriptional complexity of context-free languages, *Proc. of Symp. Math. Found. of Computer Science*, High Tatras (1973), 71–84.

[4] Gh. Păun, On the power of synchronization in parallel communicating grammar systems, *Stud. Cerc. Matem.* **41,** 3 (1989), 191–197.

[5] Gh. Păun, Parallel communicating grammar systems: the context-free case, *Found. Control Engineering* **14,** 1 (1989), 39–50.

[6] Gh. Păun, On the syntactic complexity of parallel communicating grammar systems, *RAIRO/Th. Informatics* (submitted).

[7] Gh. Păun and L. Sântean, Parallel communicating grammar systems: the regular case, *Ann. Univ. Buc., Ser. Mat.-Inform.* **37,** 2 (1989), 55–63.

[8] A. Salomaa, *Formal Languages*, Academic Press, New York, London, 1973.