The Impact of the Number of Cooperating Grammars on the Generative Power

Lila Santean & Jarkko Kari
Mathematics Department
University of Turku
20500 Turku, Finland

June 7, 2010

Abstract

The parallel communicating grammar systems consist of grammars working synchronously and sending messages one to each other. In this paper, hierarchies of classes of languages generated by such devices are investigated.

1 Introduction

Many attempts have been made for finding a suitable model for parallel computing (see [9] for an algebraic and [1, 8] for an automata theoretical approach). **P**arallel **C**ommunicating **G**rammar **S**ystems (PCGS) have been introduced in [6] as a grammatical model in this aim, trying to involve as few as possible non-syntactic components.

A PCGS of degree n consists of n separate usual Chomsky grammars, working simultaneously, each of them starting from its own axiom; furthermore, each grammar i can ask from the grammar j the string generated so far. The result of this communication is that grammar i includes in its own string the string generated by grammar j, and that grammar j returns to its axiom and resumes working. One of the grammars is distinguished as a master grammar and the terminal strings generated by it constitute the language generated by the PCGS.

Many variants of PCGS can be defined, depending on the communication protocol (see [2]), on the type of the grammars involved (see [6], [3]), and so on. An important particular case is the centralized one, where only the master grammar is allowed to ask for strings generated by the others.

We investigate here infinite hierarchies of classes of languages generated by centralized or noncentralized PCGS with regular or context-free components, determined by the degree of the PCGS, that is, the number of grammars involved.

2 Definitions and notations

We assume the reader familiar with basic definitions and notations in formal language theory (see [7]) and we specify only some notions related to PCGS.

For a vocabulary V, we denote by V^* the free monoid generated by V under the operation of concatenation, and by λ the null element. For $x \in V^*$, $|x|$ is the length of x and if K is a set, $|x|_K$ denotes the number of occurrences of letters of K in x. We denote by REG, LIN, CF, CS, the classes of regular, linear, context-free and context-sensitive grammars.

Definition A PCGS of degree n, $n \geq 1$, is a system

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

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

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

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

(ii) if $|x_i|_K > 0$ for some $i, 1 \leq i \leq n$, then for each such i we write $x_i = z_1 Q_{i_1} z_2 Q_{i_2} \ldots z_t Q_{i_t} z_{t+1}, t \geq 1$, $|z_j|_K = 0, 1 \leq j \leq t+1$; if $|x_{i_j}|_K = 0, 1 \leq j \leq t$, then $y_i = z_1 x_{i_1} z_2 x_{i_2} \ldots z_t x_{i_t} z_{t+1}$ 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 remaining indexes i, that is, for those i, $1 \leq i \leq n$, for which x_i does not contain communication symbols, we put $y_i = x_i$.

Informally, an n-tuple $(x_1, x_2, \ldots, x_n)$ directly yields $(y_1, y_2, \ldots, y_n)$ if either no communication symbol appears in $x_1, \ldots, x_n$ and we have a componentwise derivation, $x_i \Longrightarrow y_i$ in G_i, for each $i, 1 \leq i \leq n$, or communication symbols appear and we perform a *communication step*, as these symbols impose: each occurrence of Q_{i_j} in x_i is replaced by x_{i_j}, provided x_{i_j} does not contain further communication symbols.

A derivation consists of *rewriting steps* and *communication steps*.

If no communication symbol appears in any of the components, we perform a *rewriting step* which consists of a rewriting step performed synchronously in each of the grammars. If one of the components is a terminal string, it is left unchanged while the others are performing the rewriting step. If in one of the components none of the nonterminals can be rewritten any more, the derivation is blocked.

If in any of the components a communication symbol is present, a *communication step* is performed. It consists of replacing all the occurrences of communication symbols with the components they refer to, providing these components do not contain further communication symbols. If some communication symbols are not satisfied in this step, they may be satisfied in one of the next ones. Communication steps are performed until no more communication symbols are present. No rewriting is allowed if any communication symbol occurs in one of the components. Therefore, if circular queries emerge, the derivation is blocked.

The language generated by the system consists of the terminal strings generated on the first position, regardless the other components (terminal or not):

$$L(\pi) = \{\alpha \in V_{T,1}^* | (S_1, S_2, \ldots, S_n) \Longrightarrow^* (\alpha, \beta_2, \beta_3, \ldots, \beta_n)\}$$

If we impose the restriction that only the first grammar may ask for strings generated by the others, that is $K \bigcap (\bigcup_{i=2}^n V_{N,i}) = \emptyset$, we obtain the *centralized case*.

We denote by $\mathrm{PC}_n(X)$ (respectively $\mathrm{CPC}_n(X)$) the family of noncentralized (centralized) PCGS of degree n with all the components being type-X grammars, $X \in \{\mathrm{REG, LIN, CF, CS}\}$ and by $\mathcal{L}(\mathrm{PC}_n(X))$ ($\mathcal{L}(\mathrm{CPC}_n(X))$) the families of languages generated by these types of PCGS. Furthermore, $\mathrm{PC}(X)$ denotes $\bigcup_{n=1}^\infty \mathrm{PC}_n(X)$ and $\mathrm{CPC}(X)$ denotes $\bigcup_{n=1}^\infty \mathrm{CPC}_n(X)$.

Let us give now a simple example that shows the generative power of PCGS.

Example 1 Let π be the PCGS $\pi = (G_1, G_3, G_3)$ where

$$\begin{aligned}
G_1 &= (\{S_1, S_1', S_2, S_3, Q_2, Q_3\}, \{a, b, c\}, S_1, \{S_1 \longrightarrow abc, \\
&\quad S_1 \longrightarrow a^2b^2c^2, S_1 \longrightarrow a^3b^3c^3, S_1 \longrightarrow aS_1', S_1' \longrightarrow aS_1', \\
&\quad S_1' \longrightarrow a^3Q_2, S_2 \longrightarrow b^2Q_3, S_3 \longrightarrow c\}) \\
G_2 &= (\{S_2\}, \{b\}, \{S_2 \longrightarrow bS_2\}) \\
G_3 &= (\{S_3\}, \{c\}, \{S_3 \longrightarrow cS_3\}) .
\end{aligned}$$

This is a regular centralized PCGS of degree 3 and it is easy to see that we have

$$L(\pi) = \{a^nb^nc^n | n \geq 1\}$$

which is a non-context-free language. □

3 Infinite hierarchies of the language classes $\mathcal{L}(\mathbf{CPC}_n(\mathbf{REG}))$ and $\mathcal{L}(\mathbf{PC}_n(\mathbf{REG}))$

In [6], [3], [5], [4] and [2] various properties of PCGS have been investigated, including the generative power, closure under basic operations, complexity, and efficiency.

As concerning hierarchies of classes of languages generated by PCGS, it is obvious that

$$\begin{array}{rcll} \mathrm{CPC}_n(X) & \subseteq & \mathrm{CPC}_{n+1}(X), & \text{for all } n \geq 1 \text{ and } X \in \{\text{REG,LIN,CF,CS}\}, \\ \mathrm{PC}_n(X) & \subseteq & \mathrm{PC}_{n+1}(X), & \text{for all } n \geq 1 \text{ and } X \in \{\text{REG,LIN,CF,CS}\}. \end{array}$$

We shall prove in the following that for $X = \mathrm{REG}$, the inclusions are proper. Moreover, for the centralized case, a more general result, namely a pumping lemma, is obtained, but such a lemma cannot be proved for the noncentralized case.

Lemma 1 (Pumping lemma) *Let $L \in CPC_n(REG)$. There exists a natural number N such that every word α in L satisfying $|\alpha| > N$ can be decomposed as*

$$\alpha = \alpha_1\beta_1\alpha_2\beta_2 \ldots \alpha_m\beta_m\alpha_{m+1}$$

where $1 \leq m \leq n, \beta_i \neq \lambda$ for $1 \leq i \leq m$ and the word

$$\alpha_1\beta_1^k\alpha_2\beta_2^k \ldots \alpha_m\beta_m^k\alpha_{m+1}$$

is in L for all $k \geq 0$.

Proof. Let $\pi = (G_1, G_2, \ldots, G_n)$ be a centralized PCGS of degree n, where G_i are regular grammars, $G_i = (V_{N,i}, V_{T,i}, S_i, P_i), 1 \leq i \leq n$. In order to be able to iterate portions of the derivation, for obtaining the pumping effect, "similar" configurations have to be found. Therefore, we first proceed by clarifying the notion of similarity.

In every configuration of π, each component has at most one nonterminal. Let $c_1 = (x_1A_1, x_2A_2, \ldots, x_nA_n)$ and $c_2 = (y_1B_1, y_2B_2, \ldots, y_nB_n)$ be two configurations where x_i, y_i are terminal strings and A_i, B_i are nonterminals or λ, for $1 \leq i \leq n$.

The configurations c_1 and c_2 are called *equivalent* (denoted by $c_1 \equiv c_2$) if $A_i = B_i$ for each i, $1 \leq i \leq n$. Clearly, $\equiv$ is an equivalence relation and the number of equivalence classes is

$$A = \prod_{i=1}^{n}(|V_{N,i}| + 1).$$

However, the condition that two configuration are equivalent is not sufficient for iterating the subderivation between them, because communication steps may possibly occur. Therefore, a stronger condition has to be imposed on the two configurations in this aim, namely

(i) $c_1 \equiv c_2$,

(ii) if the communication symbol $Q_i, 2 \leq i \leq n$ is used in the derivation between c_1 and c_2, then $x_i = y_i$.

It will be shown in the following that in any derivation of length M_n there exist two configurations satisfying the conditions (i) and (ii), where M_n is defined recursively below:

$$
\begin{aligned}
M_1 &= A, \\
M_{j+1} &= A \cdot (P+1)^{j \cdot M_j}, \text{ for } 1 \le j \le n-1.
\end{aligned}
$$

P denotes the maximum number of productions that exist in any of the grammars $G_2, G_3, \ldots, G_n$, for any nonterminal. (We notice that starting from any nonterminal there are no more than $(P+1)^n$ different derivations of length at most n in any of the grammars.)

Claim: For every $j, 1 \le j \le n$, in any derivation of π of length M_j where less than j different communication symbols are used, there are two configurations satisfying both conditions (i) and (ii).

The claim is proved using induction on j.

If $j = 1$ then no communication symbols are used in the derivation and (ii) is trivially true. Since the length of the derivation is $M_1 = A$, there are $A+1$ configurations in it. The number of equivalence classes of $\equiv$ is A, so the *pigeon hole principle* says that (i) holds true for some configurations.

Suppose then that the claim has been proved for j. Consider a derivation of length M_{j+1} where at most j different communication symbols are present. If it contains a subderivation of length M_j where less than j different communication symbols are used, then, according to the induction hypothesis, the two configurations of the claim can be found inside this subderivation.

On the other hand, suppose that all the different communication symbols that are used in the derivation of length M_{j+1} are also used in each of its subderivations of length M_j. In the derivation there are $M_{j+1}+1$ configurations. More than $(P+1)^{j \cdot M_j}$ of them must be in the same equivalence class of $\equiv$, thus satisfying (i).

Suppose that Q_i is a communication symbol that is used in the derivation. The nearest occurrence of Q_i preceding any configuration must appear in one of the M_j predecessor configurations. Considering that after communicating, the sending grammar returns to its axiom, it follows that there may exist at most $(P+1)^{M_j}$ different i-th components in the configurations. If one counts the possibilities for all the components that correspond to all communication symbols that appear in the derivation, one gets $(P+1)^{j \cdot M_j}$ different cases. This means that we have at most $(P+1)^{j \cdot M_j}$ configurations in the derivation which differ by at least one component whose corresponding communication symbol has been used in the derivation. An application of the pigeon hole principle tells that there are two configurations in the same equivalence class which also satisfy (ii).

So, the claim has been proved. Let us return now to the pumping lemma.

Let α be a word in the language generated by π, whose length is at least $n \cdot max \cdot M_n$, where max is the maximum length of the right sides of all productions.

Then the length of the minimal derivation of α is at least M_n.

We have already shown that during this derivation there exist two configurations $c_1 = (x_1R_1, x_2R_2, \ldots, x_nR_n)$ and $c_2 = (y_1R_1, y_2R_2, \ldots, y_nR_n)$ satisfying the conditions (i) and (ii) :

$$\begin{aligned}(S_1, S_2, \ldots, S_n) &\Longrightarrow^* (x_1R_1, x_2R_2, \ldots, x_nR_n)\\ &\Longrightarrow^* (x_1z_1R_1, x_2z_2R_2, \ldots, x_nz_nR_n)\\ &\Longrightarrow^* (\alpha, \ldots)\end{aligned}$$

If Q_i is used between c_1 and c_2 then, according to property (ii), $z_i = \lambda$.

As concerning the remaining components, one of the following cases holds:

(i). z_1 is a nonempty terminal word, $z_1 \in V_{T,1}^+$,

(ii). There exists one index j such that Q_j is not used in the derivation between c_1 and c_2, Q_j is used in the derivation of α which starts with c_2, and z_j is a nonempty terminal word over $V_{T,j}^+$.

Indeed, if neither of these cases holds, this implies that the components of c_1 and c_2 are identical on the positions which are actually used in the construction of α. This would however imply that we can remove the subderivation $c_1 \Longrightarrow^* c_2$ obtaining a shorter legal derivation of α — contradiction with the assumption of minimality of the derivation.

The derivation steps between c_1 and c_2 may be repeated k times for any k. After this iteration, the components j for which z_j is a nonempty terminal word will be of the form $x_jz_j^kR_j$ and the other ones will remain unchanged.

If, after k iterations the derivation is continued by adding the steps of the subderivation $c_2 \Longrightarrow^* (\alpha, \ldots)$, a legal derivation of a terminal word generated by π is obtained. The word differs from α slightly: z_j is replaced by z_j^k if $j = 1$ or Q_j is used in the derivation steps after iteration, but not within it.

As the number of the subwords which can be thus pumped is at most n, the lemma is proved. □

We are now in position to prove the following

Theorem 1 *For all $n > 1$*

$$\mathcal{L}(CPC_n(REG)) \setminus \mathcal{L}(CPC_{n-1}(REG)) \neq \emptyset.$$

Proof. For every $n > 1$ let L_n be the language

$$L_n = \left\{a_1^{k+1}a_2^{k+2}\ldots a_n^{k+n} | k \geq 0\right\}.$$

L_n is contained in the family $\mathcal{L}(\mathrm{CPC}_n(\mathrm{REG}))$ as it is generated by the PCGS $\pi_n = (G_1, G_2, \ldots, G_n)$ where

$$G_1 = (\{S_1, S_2, \ldots, S_n, Q_2, Q_3, \ldots, Q_n\}, \{a_1, a_2, \ldots, a_n\}, S_1, P_1),$$

$$\begin{aligned} P_1 &= \{S_1 \longrightarrow a_1 S_1, \\ &\quad S_i \longrightarrow a_i Q_{i+1} \text{ for } 1 \le i \le n-1, \\ &\quad S_n \longrightarrow a_n\}, \end{aligned}$$

and when $2 \le i \le n$

$$G_i = (\{S_i\}, \{a_i\}, S_i, \{S_i \longrightarrow a_i S_i\}) .$$

However, the language L_n is not contained in $\mathcal{L}(\text{CPC}_{n-1}(\text{REG}))$. Indeed, let us assume that L_n is generated by $\pi_{n-1} \in \text{CPC}_{n-1}(\text{REG})$. Let N be the number whose existence is stated by the pumping lemma, and α the word

$$\alpha = a_1^{N+1} a_2^{N+2} \dots a_n^{N+n}.$$

Following the lemma, the words α_i obtained from α by pumping at most $n-1$ subwords of it are in L_n — contradiction with the form of the words of L_n.

We can conclude that the inclusions $\mathcal{L}(\text{CPC}_{n-1}(\text{REG})) \subset \mathcal{L}(\text{CPC}_n(\text{REG}))$ are proper for every $n > 1$. □

Corollary *The hierarchy $\mathcal{L}(CPC_n(REG)), n \ge 1$, is infinite.* □

Note. A similar pumping lemma does not hold for languages generated by noncentralized PCGS of degree n, and that is proven by the following example.

Example 2 Let $\pi = (G_1, G_2, G_3)$ where

$$\begin{aligned} G_1 &= (\{S_1, B, B_1, Q_2\}, \{a\}, S_1, \{S_1 \longrightarrow aB, \\ &\quad S_1 \longrightarrow Q_2, B_1 \longrightarrow B, B \longrightarrow \lambda, B_1 \longrightarrow \lambda\}) \\ G_2 &= (\{S_2, B, Q_1, Q_3\}, \{a\}, S_2, \{S_2 \longrightarrow Q_1, B \longrightarrow Q_3\}) \\ G_3 &= (\{S_3, Q_1, B, B_1\}, \{a\}, S_3, \{S_3 \longrightarrow Q_1, B \longrightarrow B_1\}) . \end{aligned}$$

A derivation according to π will have the following form :

$$\begin{aligned} (S_1, S_2, S_3) &\Longrightarrow (aB, Q_1, Q_1) \Longrightarrow (S_1, aB, aB) \Longrightarrow (Q_2, aQ_3, aB_1) \\ &\Longrightarrow (Q_2, a^2 B_1, S_3) \Longrightarrow (a^2 B_1, S_2, S_3) \\ &\Longrightarrow (a^2 B, Q_1, Q_1) \Longrightarrow (S_1, a^2 B, a^2 B) \\ &\Longrightarrow^* (a^{2^{n-1}} B, Q_1, Q_1) \Longrightarrow (S_1, a^{2^{n-1}} B, a^{2^{n-1}} B) \\ &\Longrightarrow (Q_2, a^{2^{n-1}} Q_3, a^{2^{n-1}} B_1) \Longrightarrow (Q_2, a^{2^n} B_1, S_3) \\ &\Longrightarrow (a^{2^n} B_1, S_2, S_3) \Longrightarrow (a^{2^n}, Q_1, Q_1), \text{ for any } n \ge 1. \end{aligned}$$

We notice that in all the cases where the production $S_1 \longrightarrow aB$ was applicable instead of $S_1 \longrightarrow Q_2$, its application would have inevitably led to the word a. Therefore we conclude that

$$L(\pi) = \left\{ a^{2^n} | n \ge 0 \right\} .$$

If a pumping lemma would hold for languages in $\mathcal{L}(\mathrm{PC}(\mathrm{REG}))$, then the set of lengths of words of any infinite language in $\mathcal{L}(\mathrm{PC}(\mathrm{REG}))$ would contain an infinite arithmetical progression. The lengths of words in $L(\pi) \in \mathcal{L}(\mathrm{PC}_3(\mathrm{REG}))$ grow exponentially, therefore such an infinite arithmetical progression cannot be found. So, a pumping lemma for languages generated by noncentralized PCGS of degree n does not hold. □

However, even if such a lemma is not true, the infinity of the hierarchy $\mathcal{L}(\mathrm{PC}(\mathrm{REG}))$ can be directly proven by finding a language that can be generated by a noncentralized PCGS of degree $m+1$ but not by a noncentralized PCGS of degree m.

Theorem 2 *For all $m \geq 1$*

$$\mathcal{L}(PC_{m+1}(REG)) \setminus \mathcal{L}(PC_m(REG)) \neq \emptyset.$$

Proof. Let L be the language

$$L = \{a_1^n a_2^n \dots a_{2m}^n | n \in \mathbf{N}\}$$

We shall prove the theorem by showing that L belongs to $\mathcal{L}(\mathrm{PC}_{m+1}(\mathrm{REG}))$ but not to $\mathcal{L}(\mathrm{PC}_m(\mathrm{REG}))$. In the following we show that L is equal to the language generated by the PCGS

$$\pi = (G_1, G_2, \dots, G_{m+1}) \in \mathrm{PC}_{m+1}(\mathrm{REG})$$

where $G_i = (V_{N,i}, V_{T,i}, S_i, P_i)$ are regular grammars for $1 \leq i \leq m+1$ and

$$\begin{aligned}
V_{T,i} &= \{a_1, a_2, \dots, a_{2m}\}, 1 \leq i \leq m+1 \\
V_{N,1} &= \{S_1\} \cup \{Q_i | 2 \leq i \leq m+1\} \cup \{X_2^k | 1 \leq k \leq 2m+1\} \\
&\quad \cup \{X_j^{2m+1} | 2 \leq j \leq m+1\} \\
V_{N,i} &= \{S_i, \alpha_i\} \cup \{Q_{i-1}, Q_{i+1}\} \cup \{X_i^k | 1 \leq k \leq 2m+1\} \\
&\quad \cup \{X_{i+1}^k | i \leq k \leq 2m-i+1\}, 2 \leq i \leq m \\
V_{N,m+1} &= \{S_{m+1}, \alpha_{m+1}\} \cup \{Q_m\} \cup \{X_{m+1}^k | 1 \leq k \leq 2m+1\} \\
P_1 &= \{S_1 \longrightarrow a_1 Q_2, X_2^1 \longrightarrow a_2 X_2^2, S_1 \longrightarrow a_1 a_2 \dots a_{2m}, \\
&\quad X_{m+1}^{2m+1} \longrightarrow a_{2m}\} \cup \{X_2^k \longrightarrow X_2^{k+1} | 2 \leq k < 2m\} \\
&\quad \cup \{X_j^{2m+1} \longrightarrow a_{2j-2} a_{2j-1} Q_{j+1} | 2 \leq j \leq m\}, \\
P_j &= \{S_j \longrightarrow X_j^1, S_j \longrightarrow a_{2j-1} Q_{j+1}, S_j \longrightarrow Q_{j-1}, S_j \longrightarrow \alpha_j\} \\
&\quad \cup \{X_j^k \longrightarrow X_j^{k+1} | 1 \leq k < j-1\} \\
&\quad \cup \{X_{j+1}^j \longrightarrow a_{2j} X_{j+1}^{j+1}\} \\
&\quad \cup \{X_{j+1}^k \longrightarrow X_{j+1}^{k+1} | j < k \leq 2m-j\}
\end{aligned}$$

$$
\begin{aligned}
&\cup\{X_j^k \longrightarrow X_j^{k+1} | 2m-j+1 < k \leq 2m-1\} \\
&\cup\{X_j^{2m} \longrightarrow X_j^1, X_j^{2m} \longrightarrow X_j^{2m+1}\} \\
&\cup\{X_j^{2m+1} \longrightarrow X_j^{2m+1}\} \\
&\cup\{\alpha_j \longrightarrow \alpha_j\},\ \text{for } 2 \leq j \leq m \\
P_{m+1} \quad = \quad &\{S_{m+1} \longrightarrow X_{m+1}^1, S_{m+1} \longrightarrow Q_m, S_{m+1} \longrightarrow \alpha_{m+1}, \\
&X_{m+1}^{2m} \longrightarrow X_{m+1}^1, X_{m+1}^{2m+1} \longrightarrow X_{m+1}^{2m+1}, \alpha_{m+1} \longrightarrow \alpha_{m+1}\} \\
&\cup\{X_{m+1}^k \longrightarrow X_{m+1}^{k+1} | 1 \leq k \leq 2m, k \neq m\}
\end{aligned}
$$

For proving that $L \subseteq L(\pi)$ we shall show that, for every n, the word $a_1^n a_2^n \dots a_{2m}^n$ can be generated by π.

Claim: For all $n \in \mathbf{N}$, there exists a derivation $D : (S_1, S_2, \dots, S_{m+1}) \Longrightarrow^* \left(a_1 Q_2, a_1^n a_2^n X_2^1, \dots, a_{2m-1}^n a_{2m}^n X_{m+1}^1\right)$, according to π.

The claim shall be proved by induction on n. For $n = 0$, we can construct the derivation

$$\left(S_1, S_2, \dots, S_{m+1}\right) \Longrightarrow \left(a_1 Q_2, X_2^1, \dots, X_{m+1}^1\right).$$

Let us suppose now that there exists a derivation D

$$(S_1, S_2, \dots, S_{m+1}) \Longrightarrow^* \left(a_1 Q_2, a_1^n a_2^n X_2^1, \dots, a_{2m-1}^n a_{2m}^n X_{m+1}^1\right).$$

We shall construct a valid derivation D' for the configuration

$$\left(a_1 Q_2, a_1^{n+1} a_2^{n+1} X_2^1, \dots, a_{2m-1}^{n+1} a_{2m}^{n+1} X_{m+1}^1\right).$$

The idea of the construction is the following. We shall add a subderivation to the derivation D, such that every component, excepting the first one, shall have in the end the exponent increased by one. The increasing of the exponent implies the catenation of one letter to the left side of the terminal word, and one to the right. This wouldn't be possible in an ordinary regular grammar, where the letters are only added to one end. Using the communication, letters can be added here to both ends of the terminal word of some component. This is done first by communicating the word to the left component. Together with the communication symbol, a letter is produced, that means it is catenated to the left end of the word. Afterwards, working in this auxiliary component another letter is produced, that means it is catenated to the right. Finally, (after the change has been made in all components) the new word is communicated back to the original component where it belonged.

This procedure can be applied in a chain, from left to right, using the fact that we have one grammar for which we do not need to change the word, that is we have an auxiliary place. After all the needed letters are produced, the new strings are in components situated to the left of their original ones. Then, beginning with the m'th component, the strings are moved one position to the right, and the requested configuration is obtained. Special attention has to be paid to the components in the "waiting status", because the changing of the string is only done for one component at a time. Therefore, until the turn of a particular component to be communicated comes, only renamings are performed in it, the upper index of the nonterminals $X_j^k, 1 \leq j \leq m+1, 1 \leq k \leq 2m+1$ counting the "waiting" steps.

The derivation D' has therefore the following form:

$$\big(a_1Q_2, \ldots, a_{2j-3}^n a_{2j-2}^n X_j^1, a_{2j-1}^n a_{2j}^n X_{j+1}^1, \ldots, a_{2m-1}^n a_{2m}^n X_{m+1}^1\big)$$

$$\Downarrow \begin{array}{l} j-1 \text{ rewriting steps and} \\ j-1 \text{ communication steps} \end{array}$$

$$\Big(a_1^{n+1} a_2^{n+1} X_2^j, \ldots, a_{2j-1} Q_{j+1}, a_{2j-1}^n a_{2j}^n X_{j+1}^j, \ldots, a_{2m-1}^n a_{2m}^n X_{m+1}^j\Big)$$

$$\Downarrow \text{ communication step}$$

$$(a_1^{n+1} a_2^{n+1} X_2^j, \ldots, a_{2j-1}^{n+1} a_{2j}^n X_{j+1}^j, S_{j+1}, \ldots, a_{2m-1}^n a_{2m}^n X_{m+1}^j)$$

$$\Downarrow \text{ rewriting step}$$

$$(a_1^{n+1} a_2^{n+1} X_2^{j+1}, \ldots, a_{2j-1}^{n+1} a_{2j}^{n+1} X_{j+1}^{j+1}, a_{2j+1} Q_{j+2}, \ldots, a_{2m-1}^n a_{2m}^n X_{m+1}^{j+1})$$

$$\Downarrow_* \begin{array}{l} m-j \text{ communication steps and} \\ m-j-1 \text{ rewriting steps} \end{array}$$

$$(a_1^{n+1} a_2^{n+1} X_2^m, \ldots, a_{2j-1}^{n+1} a_{2j}^{n+1} X_{j+1}^m, a_{2j+1}^{n+1} a_{2j+2}^{n+1} X_{j+2}^m, \ldots, S_{m+1})$$

$$\Downarrow \text{ rewriting step}$$

$$(a_1^{n+1} a_2^{n+1} X_2^{m+1}, \ldots, a_{2j-1}^{n+1} a_{2j}^{n+1} X_{j+1}^{m+1}, a_{2j+1}^{n+1} a_{2j+2}^{n+1} X_{j+2}^{m+1}, \ldots, Q_m)$$

$$\Downarrow_* \begin{array}{l} m \text{ communication steps and} \\ m-1 \text{ rewriting steps} \end{array}$$

$$(S_1, \ldots, a_{2j-3}^{n+1} a_{2j-2}^{n+1} X_j^{2m}, a_{2j-1}^{n+1} a_{2j}^{n+1} X_{j+1}^{2m}, \ldots, a_{2m-1}^{n+1} a_{2m}^{n+1} X_{m+1}^{2m})$$

$$\Downarrow \text{ rewriting step}$$

$$(a_1 Q_2, \ldots, a_{2j-3}^{n+1} a_{2j-2}^{n+1} X_j^1, a_{2j-1}^{n+1} a_{2j}^{n+1} X_{j+1}^1, \ldots, a_{2m-1}^{n+1} a_{2m}^{n+1} X_{m+1}^1).$$

We have found a derivation according to π for the configuration requested by the induction step, therefore the claim is proved.

The membership of the word $a_1^n a_2^n \ldots a_{2m}^n$ in $L(\pi)$ for every $n \geq 1$ follows now from the claim. Indeed, we replace the last step of the derivation D (in which a new round is started) with a subderivation which plays the role of collecting all the strings in the first component, in the correct order.

Therefore we have:

$$\begin{aligned}
(S_1, S_2, \ldots, S_{m+1}) &\Longrightarrow^* (S_1, a_1^n a_2^n X_2^{2m}, \ldots, a_{2m-1}^n a_{2m}^n X_{m+1}^{2m}) \\
&\Longrightarrow (a_1 Q_2, a_1^n a_2^n X_2^{2m+1}, \ldots, a_{2m-1}^n a_{2m}^n X_{m+1}^{2m+1}) \\
&\Longrightarrow (a_1^{n+1} a_2^n X_2^{2m+1}, S_2, \ldots, a_{2m-1}^n a_{2m}^n X_{m+1}^{2m+1}) \\
&\Longrightarrow (a_1^{n+1} a_2^{n+1} a_3 Q_3, \alpha_2, \ldots a_{2m-1}^n a_{2m}^n X_{m+1}^{2m+1}) \\
&\Longrightarrow^* (a_1^{n+1} a_2^{n+1} \ldots a_{2m}^n X_{m+1}^{2m+1}, \alpha_2, \ldots, \alpha_m, S_{m+1}) \\
&\Longrightarrow (a_1^{n+1} a_2^{n+1} \ldots a_{2m}^{n+1}, \alpha_2, \ldots, \alpha_m, \alpha_{m+1}).
\end{aligned}$$

The converse inclusion follows because, except the alternative of stopping the derivation, the use of other productions than the ones we have actually used leads to the blocking of the derivation (either by introducing nonterminals which cannot be further rewritten, or by introducing circular communication requests). This implies that the only words that can be generated by the PCGS π are the ones of the form $a_1^n a_2^n \ldots a_{2m}^n$.

We have therefore proven that $L(\pi) = L$, which shows that L belongs to $\mathcal{L}(\mathrm{PC}_{m+1}(\mathrm{REG}))$.

Next we prove that $L \notin \mathcal{L}(\mathrm{PC}_m(\mathrm{REG}))$. Let us assume, on the contrary, that there exists a PCGS $\pi' \in \mathrm{PC}_m(\mathrm{REG})$ such that $L = L(\pi')$.

There exists a function $f : \mathbf{N} \to \mathbf{N}$ such that, every configuration obtainable from the initial one after n steps (we count the rewriting as well as the communication steps) possesses only components of length less than $f(n)$. In fact it is easy to see that we can choose $f(n) = max \cdot 2^n$, where max is the maximum length of the right sides of all productions. Let p be the number of equivalence classes determined by the equivalence relation $\equiv$ defined in the *Pumping lemma*. Let now w be the word $w = a_1^{f(2p)} \ldots a_{2m}^{f(2p)}$ and D a minimal derivation of it. The length of the derivation is greater than $2p$, therefore, during the first p steps we find two equivalent configurations:

$$\begin{aligned}
(S_1, \ldots, S_{2m}) &\Longrightarrow^* c_1 = (x_1 A_1, \ldots, x_m A_m) \\
&\Longrightarrow^* c_2 = (y_1 A_1, \ldots, y_m A_m) \\
&\Longrightarrow^* (w, \ldots),
\end{aligned}$$

where $|x_i|, |y_i| < f(p)$ for every $1 \le i \le m$.

We first notice that no word $y_i, 1 \le i \le m$ which contains more than two different letters can become a subword of w. This follows because, if some "useful" y_i would contain at least three different letters, the exponent of the middle letter would remain less than $f(2p)$ — contradiction. We further notice that all terminal letters must appear in some "useful" $y_i, 1 \le i \le n$. Indeed, let's suppose that some letter would be only generated after the appeareance of c_2. Then we could construct a derivation obtained from D by continuing the subderivation which ends with c_1 with the steps of the subderivation $c_2 \Longrightarrow^* (w, \ldots)$.

The word obtained in this way is a terminal one, different from w (recall that D is a minimal derivation) but still the exponent of the letter generated in the last mentioned subderivation is $f(2p)$ — contradiction. Combining these two observations, we conclude that every word $y_i, 1 \leq i \leq m$ is of the form $y_i = a_j^{q_j} a_k^{q_k}, 1 \leq j, k \leq 2m, j \neq k, q_j + q_k < f(p)$, and all terminal letters appear in some y_i.

As the derivation D has the length greater than $2p$, we shall find among the configurations that follow c_2 two more equivalent configurations:

$$\begin{aligned} D : (S_1, \ldots, S_m) &\Longrightarrow^* \quad c_2 = (y_1 A_1, \ldots, y_m A_m) \\ &\Longrightarrow^* \quad c_3 = (z_1 B_1, \ldots, z_m B_m) \\ &\Longrightarrow^* \quad c_4 = (t_1 B_1, \ldots, t_m B_m) \\ &\Longrightarrow^* \quad (w, \ldots). \end{aligned}$$

Using a similar reasoning as above and the fact that all the letters of w appear already in the components of c_2, we conclude that c_3 and c_4 have the same properties as c_2 regarding their form and contribution to w. No communication step is involved in the derivation between them (if that would be the case we would find in c_4 a component t_i containing more than 2 different letters).

If we construct now a derivation obtained from D by continuing from c_3 with the steps of the subderivation $c_4 \Longrightarrow^* (w, \ldots)$ we obtain a word in $L(\pi)$ in which some letters have $f(2p)$ occurrences (regular rewriting can add letters only to the right, so that the number of some terminal letters does not change in the subderivation $c_3 \Longrightarrow^* c_4$). However, the word obtained cannot be w, because D was a minimal derivation of w — contradiction.

It follows that our assumption that L can be generated by a PCGS of degree m with regular components is false. We have proved that L can be generated by a regular PCGS of degree $m+1$ but not by a regular PCGS of degree m. □

Corollary *The hierarchy $\mathcal{L}(PC_n(REG)), n \geq 1$ is infinite.* □

Corollary *For all $m \geq 1$*

$$\mathcal{L}(CPC_{2m}(REG)) \setminus \mathcal{L}(PC_m(REG)) \neq \emptyset.$$

It follows from the proofs of theorems 1 and 2.

Example 3 As an example of the PCGS's constructed in the proof of the previous theorem we present the PCGS $\pi_3 = (G_1, G_2, G_3)$ of degree 3 generating the language $\{a_1^n a_2^n a_3^n a_4^n | n \geq 1\}$. The components are defined as follows:

$$\begin{aligned} G_1 \quad = \quad & \big(\{S_1, Q_2, Q_3, X_2^1, X_2^2, X_2^3, X_2^4, X_2^5, X_3^5\}, \{a_1, a_2, a_3, a_4\}, S_1, \\ & \{S_1 \longrightarrow a_1 Q_2, X_2^1 \longrightarrow a_2 X_2^2, X_2^2 \longrightarrow X_2^3, X_2^3 \longrightarrow X_2^4, \\ & S_1 \longrightarrow a_1 a_2 a_3 a_4, X_2^5 \longrightarrow a_2 a_3 Q_3, X_3^5 \longrightarrow a_4\}\big) \end{aligned}$$

$$
\begin{aligned}
G_2 \;=\;& \big(\{S_2, Q_1, Q_3, X_2^1, X_2^2, X_2^3, X_2^4, X_2^5, X_3^2, X_3^3, \alpha_2\}, \{a_1, a_2, a_3, a_4\}, S_2,\\
& \{S_2 \longrightarrow X_2^1, S_2 \longrightarrow a_3 Q_3, S_2 \longrightarrow Q_1, X_3^2 \longrightarrow a_4 X_3^3, X_2^4 \longrightarrow X_2^1,\\
& X_2^4 \longrightarrow X_2^5, X_2^5 \longrightarrow X_2^5, S_2 \longrightarrow \alpha_2, \alpha_2 \longrightarrow \alpha_2\}\big)\\
G_3 \;=\;& \big(\{S_3, Q_2, X_3^1, X_3^2, X_3^3, X_3^4, X_3^5, \alpha_3\}, \{a_1, a_2, a_3, a_4\}, S_3,\\
& \{S_3 \longrightarrow X_3^1, S_3 \longrightarrow Q_2, X_3^1 \longrightarrow X_3^2, X_3^3 \longrightarrow X_3^4, X_3^4 \longrightarrow X_3^1,\\
& X_3^4 \longrightarrow X_3^5, X_3^5 \longrightarrow X_3^5, S_3 \longrightarrow \alpha_3, \alpha_3 \longrightarrow \alpha_3\}\big)
\end{aligned}
$$

Derivation according to π_3 has the following form:

$$
\begin{aligned}
(S_1, S_2, S_3) \;&\Longrightarrow\; (a_1 Q_2, X_2^1, X_3^1) \Longrightarrow (a_1 X_2^1, S_2, X_3^1)\\
&\Longrightarrow\; (a_1 a_2 X_2^2, a_3 Q_3, X_3^2) \Longrightarrow (a_1 a_2 X_2^2, a_3 X_3^2, S_3)\\
&\Longrightarrow\; (a_1 a_2 X_2^3, a_3 a_4 X_3^3, Q_2) \Longrightarrow (a_1 a_2 X_2^3, S_2, a_3 a_4 X_3^3)\\
&\Longrightarrow\; (a_1 a_2 X_2^4, Q_1, a_3 a_4 X_3^4) \Longrightarrow (S_1, a_1 a_2 X_2^4, a_3 a_4, X_3^4)\\
&\Longrightarrow^*\; (S_1, a_1^n a_2^n X_2^4, a_3^n a_4^n X_3^4) \Longrightarrow (a_1 Q_2, a_1^n a_2^n X_2^5, a_3^n a_4^n X_3^5)\\
&\Longrightarrow\; (a_1^{n+1} a_2^n X_2^5, S_2, a_3^n a_4^n X_3^5)\\
&\Longrightarrow\; (a_1^{n+1} a_2^{n+1} a_3 Q_3, \alpha_2, a_3^n a_4^n X_3^5)\\
&\Longrightarrow\; (a_1^{n+1} a_2^{n+1} a_3^{n+1} a_4^n X_3^5, \alpha_2, S_3)\\
&\Longrightarrow\; (a_1^{n+1} a_2^{n+1} a_3^{n+1} a_4^{n+1}, \alpha_2, \alpha_3).
\end{aligned}
$$

□

The problem of the infinity of the hierarchies $\mathcal{L}(\mathrm{PC}_n(X))$ and $\mathcal{L}(\mathrm{CPC}_n(X))$, $n \geq 1$ remains open for $X \in \{CF, CS\}$. The conjecture is that $\mathcal{L}(\mathrm{PC}_n(CF))$, $\mathcal{L}(\mathrm{CPC}_n(CF))$, $n \geq 1$ are infinite. Still, this cannot be proved using a similar pumping lemma that we have used above, because such a lemma does not hold.

Indeed, let us consider the following PCGS: $\pi = (G_1, G_2)$, where

$$
\begin{aligned}
G_1 \;=\;& (\{S_1, Q_2, S_2\}, \{a_1, a_2, a_3, a_4\}, S_1,\\
& \{S_1 \longrightarrow a_1 S_1 a_2, S_1 \longrightarrow a_1 Q_2^k a_2, S_2 \longrightarrow \lambda\}\big)\\
G_2 \;=\;& (\{S_2\}, \{a_3, a_4\}, S_2, \{S_2 \longrightarrow a_3 S_2 a_4\}).
\end{aligned}
$$

It is easy to see that the the language generated by π is

$$
L(\pi) = \{a_1^n \left(a_3^n a_4^n\right)^k a_2^n | n \geq 1\}
$$

Suppose that there is a pumping lemma analogous to the one presented earlier for the regular case, which would say that every long enough word in every language $L \in \mathcal{L}(\mathrm{CPC}_2(\mathrm{CF}))$ can be pumped in at most C positions, for some constant C (C depending only on the number of the components). However, the language $L(\pi)$ with $k > C$ would not satisfy the pumping lemma. The problem arises because of the possibility of simultaneous communication symbols occurring on the left side of productions, that implies that the number of positions that could be pumped does not depend only on the number of the components.

References

[1] J.Kari. Decision Problems concerning Cellular Automata. *University of Turku, PhD Thesis* (1990).

[2] Gh.Paun. On the power of synchronization in parallel communicating grammar systems. *Stud. Cerc. Matem.* **41** vol.3 (1989).

[3] Gh.Paun. Parallel communicating grammar systems: the context-free case. *Found. Control Engineering* **14** vol.1 (1989).

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

[5] Gh.Paun, L.Santean. Further remarks on parallel communicating grammar systems. *International Journal of Computer Mathematics* **35** (1990).

[6] Gh.Paun, L.Santean. Parallel communicating grammar systems: the regular case. *Ann. Univ. Buc. Ser. Mat.-Inform.* **37** vol.2 (1989).

[7] A.Salomaa. *Formal languages.* Academic Press New York London (1973).

[8] K.Culik, J.Gruska, A.Salomaa. Systolic trellis automata. *International Journal of Computer Mathematics* **15** and **16**. (1984).

[9] C.A.R.Hoare. Communicating sequential processes. *Comm. ACM* **21** vol. 8 (1978).