INFORMATIQUE THÉORIQUE ET APPLICATIONS

S. DUMITRESCU

G. PĂUN

On the power of parallel communicating grammar systems with right-linear components

Informatique théorique et applications, tome 31, nº 4 (1997), p. 331-354

<http://www.numdam.org/item?id=ITA_1997__31_4_331_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 31, n° 4, 1997, pp. 331-354)

ON THE POWER OF PARALLEL COMMUNICATING GRAMMAR SYSTEMS WITH RIGHT-LINEAR COMPONENTS (*)

by S. Dumitrescu (1) and G. Păun (1)

Abstract. – *We settle here two problems concerning the generative power of parallel communicating grammar systems with right-linear components: (1) each linear language can be generated by a non-centralized returning system, (2) the family of languages generated by centralized returning systems is incomparable with the family of languages generated by non-returning centralized systems. It is also proved that centralized returning systems with right-linear components are strictly more powerful than systems with regular rules in the restricted sense.*

1. INTRODUCTION

A parallel communicating (PC) grammar system is a construct consisting of several usual grammars, working synchronously, each on its own sentential form, and communicating by request; special (*query*) symbols are provided, Q_i, with the subscript identifying a component of the system; when a component j introduces a query symbol Q_i, the current sentential form of the component i is sent to the component j, where it replaces the occurrence(s) of Q_i in the sentential form of component j. The language generated by a specified component of the system (the *master*), after a series of such rewriting and communication steps (each component starts from its axiom), is the language generated by the system.

Many papers were devoted in the last years to the study of PC grammar systems. We refer to [2], [4] for details.

(*) Received November 1995, Accepted May 1997.
(1) Research supported by the Academy of Finland, project 11281.
Institute of Mathematics of the Romanian Academy, PO Box 1–764, 70700 Bucureşti, Romania, E-mail: gpaun@imar.ro

Many of these papers deal with PC grammar systems with regular components, where "regular" means in general "right-linear". However, a series of basic problems are still open in this area.

Two important classifications of PC grammar systems concern the *communication graph* and the *returning* feature: a system is called *centralized* when only the master may introduce query symbols and *non-centralized* in the non-restricted case; a system is called *returning* if after communicating a component resumes working from its axiom and *non-returning* when it has to continue processing the current string.

Centralized returning PC grammar systems with right-linear components are known to generate only semilinear languages, but all other types (non-centralized returning, centralized non-returning, and non-centralized non-returning) can generate non-semilinear languages. However, it is not known whether there are centralized returning systems which cannot be simulated by non-returning centralized systems. We show that this is the case, hence the two families are incomparable.

Another important open problem concerns the relationships between the families of linear and of context-free languages and those of languages generated, in the returning or non-returning way, by non-centralized systems with right-linear components. For instance, in [1] it is proved that there are context-free languages which cannot be generated by returning centralized regular PC grammar systems, unless $CF \subseteq NLOG$ (which is not at all expected). We solve here the problem for linear languages: returning non-centralized PC grammar systems with right-linear components can generate all linear languages. (We conjecture that this is not true for the non-returning mode.)

Finally, we show that at least in the returning centralized case, there is a difference between using right-linear rules and using regular rules in the strict sense. This shows that for PC grammar systems this distinction is important. In almost all cases in formal language theory, there is no difference from the generative capacity point of view between mechanisms using regular rules and those using right-linear rules. One of the basic features of a PC grammar system is the *synchronization* of the rewriting steps, hence the "speed" of producing strings on various components. This is the place where right-linear rules prove to be strictly more powerful than the regular ones. In fact, as we shall see, the chain rules are essential, not the rules of the form $A \rightarrow xB$ with A, B nonterminals and x a terminal string of the length greater than or equal to two: every right-linear system, centralized or not, returning or not, can be simulated (modulo λ) by a system of the same type and having only rules of the forms $A \to aB$, $A \to B$, $A \to a$, with A, B nonterminals and a terminal.

2. PARALLEL COMMUNICATING GRAMMAR SYSTEMS

For an alphabet V, we denote by V^* the free monoid generated by V; λ is the empty string, $|x|$ is the length of $x \in V^*$, $|x|_U$ is the number of occurrences of symbols in $U \subseteq V$ in $x \in V^*$. *REG, CF, CS, RE* denote the families in the Chomsky hierarchy. For further facts of formal language theory we shall use in the sequel, we refer to [12].

A *PC grammar system* of degree, n, $n \geq 1$, is a construct

$$\Gamma = (N, T, K, (P_1, S_1), \ldots, (P_n, S_n)),$$

where N, T, K are pairwise disjoint alphabets, with $K = \{Q_1, \ldots, Q_n\}$, $S_i \in N$, and P_i are finite sets of rewriting rules over $N \cup T \cup K$, $1 \leq i \leq n$; the elements of N are *nonterminal* symbols, those of T are *terminals*; the elements of K are called *query symbols*; the pairs (P_i, S_i) are the *components* of the system (often, we call the sets P_i components). Note that, by their indices, the query symbols are associated with the components. When discussing the type of the components in the Chomsky hierarchy, the query symbols are interpreted as nonterminals.

For $(x_1, \ldots, x_n)$, $(y_1, \ldots, y_n)$, with $x_i, y_i \in (N \cup T \cup K)^*$, $1 \leq i \leq n$ (we call *configuration* such an n-tuple) with $x_1 \notin T^*$, we write $(x_1, \ldots, x_n) \Rightarrow_r (y_1, \ldots, y_n)$ if one of the following two cases holds:

(i) $|x_i|_K = 0$ for all $1 \leq i \leq n$; then $x_i \Rightarrow_{P_i} y_i$ or $x_i = y_i \in T^*$, $1 \leq i \leq n$;

(ii) there is $i, 1 \leq i \leq n$, such that $|x_i|_K > 0$; we write such a string x_i as $x_i = z_1 Q_{i_1} z_2 Q_{i_2} \ldots z_t Q_{i_t} z_{t+1}$, for $t \geq 1, z_i \in (N \cup T)^*, 1 \leq i \leq t+1$; if $|x_{i_j}|_K = 0$ for all $1 \leq j \leq t$, then $y_i = z_1 x_{i_1} z_2 x_{i_2} \ldots z_t x_{i_t} z_{t+1}$, [and $y_{i_j} = S_{i_j}$, $1 \leq j \leq t$]; otherwise $y_i = x_i$. For all unspecified i we have $y_i = x_i$.

Point (i) defines a *rewriting* step (componentwise, on all components whose current strings are not terminal), (ii) defines a *communication* step: the query symbols Q_{i_j} introduced in some x_i are replaced by the associated strings x_{i_j}, providing that these strings do not contain further query symbols. The communication has priority over rewriting. The work of the system is blocked when circular queries appear, as well as when no query symbol is present but point (i) is not performed because a component cannot rewrite its sentential form, although it is a nonterminal string.

The above considered relation $\Rightarrow_r$ is said to be performed in the *returning* mode: after communicating, a component resumes working from its axiom. If the brackets, [and $y_{i_j} = S_{i_j}, 1 \leq i \leq t$], are removed, then we obtain the *non-returning* mode of derivation: after communicating, a component continues the processing of the current string. We denote by $\Rightarrow_{nr}$ the obtained relation.

The language generated by Γ is the language generated by its first component (G_1 above), when starting from $(S_1, \ldots, S_n)$, that is

$$L_f(\Gamma) = \{w \in T^* | (S_1, \ldots, S_n) \Rightarrow_f^* (w, \alpha_2, \ldots, \alpha_n),$$
$$\text{for} \quad \alpha_i \in (N \cup T \cup K)^*, \ 2 \leq i \leq n\}, \ f \in \{r, nr\}.$$

(No care is paid to strings in the components $2, \ldots, n$ in the last configuration of a derivation; clearly, the work of Γ stops when a terminal string is obtained in the first component.)

Two basic classes of PC grammar systems can be distinguished: *centralized* (only G_1, the *master* of the system, is allowed to introduce query symbols), and *non-centralized* (no restriction is imposed on the introduction of query symbols). Therefore, we get four basic families of languages: we denote by *PC(X)* the family of languages generated in the returning mode by non-centralized PC grammar systems with rules of type X (and of arbitrary degree); when centralized systems are used, we add the symbol C, when the non-returning mode of derivation is used, we add the symbol N, thus obtaining the families *CPC(X)*, *NPC(X)*, *NCPC(X)*. In what concerns *X*, we consider here regular (REG), right-linear (RL) and context-free (CF) rules.

By *regular* rules we mean rewriting rules of the forms $A \rightarrow aB$, $A \rightarrow a$, for A, B nonterminals and a terminal. By *right-linear* rules we mean rules of the forms $A \rightarrow xB$, $A \rightarrow x$, with A, B nonterminals and x terminal strings.

In all cases, we allow only λ-free rules. If the language we consider contains the empty string, then a rule $S \rightarrow \lambda$ is allowed in the master grammar. (Note that, because the derivation stops when using such a rule, λ cannot be communicated to another component.)

Here are two simple **examples**. Consider the system

$$\Gamma_1 = (\{S_1, S_2, S_3\}, K, \{a, b, c\}, (P_1, S_1), (P_2, S_2), (P_3, S_3)),$$
$$P_1 = \{S_1 \rightarrow aS_1, \ S_1 \rightarrow aQ_2, \ S_2 \rightarrow bQ_3, \ S_3 \rightarrow c\},$$
$$P_2 = \{S_2 \rightarrow bS_2\},$$
$$P_3 = \{S_3 \rightarrow cS_3\}.$$

The reader can easily see that

$$L_r(\Gamma_1) = L_{nr}(\Gamma_2) = \{a^n b^{n+1} c^{n+2} | n \geq 1\}.$$

Note that Γ_1 contains regular rules only, but $L_f(\Gamma)$ is not context-free, $f \in \{r, nr\}$. Moreover, consider

$$\begin{aligned} \Gamma_2 &= (\{S\}, K, \{a\}, (P_1, S), (P_2, S)), \\ P_1 &= \{S \to aQ_2,\ S \to a\}, \\ P_2 &= \{S \to aS\}. \end{aligned}$$

We obtain

$$\begin{aligned} L_r(\Gamma_2) &= \{a^{2n+1} | n \geq 0\}, \\ L_{nr}(\Gamma_2) &= \{a^{\frac{(n+1)(n+2)}{2}} | n \geq 0\}. \end{aligned}$$

The language $L_r(\Gamma_2)$ is regular, but $L_{nr}(\Gamma_2)$ is not regular (it is not even a matrix language, [8]). Again the system contains only regular rules.

The diagram in Figure 1 indicates the relations between the eight basic families of languages discussed in this paper, as well as their relationships with families in the Chomsky hierarchy (*MAT* denotes the family of languages generated by matrix grammars with λ-free context-free rules and without appearance checking). The arrows indicate inclusions, not necessarily proper; the families which are not connected by a path in this diagram are not necessarily incomparable.

Proofs of these relations can be found in [2], [5], [9], [10].

We shall add to this diagram two important relations: $LIN \subset PC(RL)$ and $CPC(REG) - NCPC(RL) \neq \emptyset$; in this way we obtain that $CPC(RL)$ and $NCPC(RL)$ are incomparable, the only incomparability result known in this moment in the PC grammar systems area.

3. COMPARING THE PREVIOUS FAMILIES

At the first sight, because systems with right-linear components generate the strings "from the left to the right", linear languages of the form

$$L = \{wc\, mi(w) | w \in \{a, b\}^*\}$$

cannot be generated by such systems. This is true for centralized systems (see the proof for point (iv) of Theorem 4.1, in [2], using the linear language $\{a^n b^m c b^m a^n | n, m \geq 1\}$, which is shown not to be in *CPC(RL)*, and this

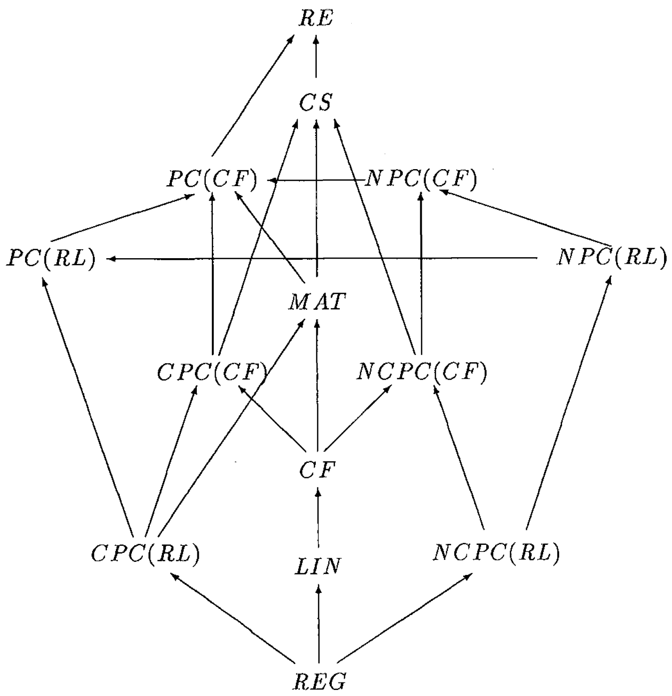

Figure 1.

is probably true also for non-centralized non-returning systems. However, the returning centralized framework provides tools for simulating linear grammars by systems with right-linear components.

THEOREM 1: $LIN \subset PC(RL)$.

Proof: Clearly, we have to prove only the inclusion (*PC*(*RL*) contains non-context-free languages).

Take a linear language $L \subseteq T^*$. We can write

$$L = (L \cap \{\lambda\}) \cup \bigcup_{a \in T} \partial_a^r(L)\{a\},$$

(∂_a^r denotes the right derivative with respect to the symbol a). The family *PC*(*RL*) is closed under union ([2], Theorem 7.56 and the remark after it), hence it is enough to prove that each language $\partial_a^r(L)\{a\}$ is in *PC*(*RL*).

The family *LIN* is closed under right derivative, hence $\partial_a^r(L) \in LIN$ for each $a \in T$. Take a linear grammar $G_a = (N_a, T, P_a, S_a)$ such that

$L(G_a) = \partial_a^r(L)$, $a \in T$. Assume that

$$P_a = P_{a,N} \cup P_{a,T},$$
$$P_{a,N} = \{r_i : A_i \to u_i B_i v_i | 1 \le i \le n,\ A_i,\ B_i \in N_a,\ u_i,\ v_i \in T^*\},$$
$$P_{a,T} = \{q_i : C_i \to x_i | 1 \le i \le m,\ C_i \in N_a,\ x_i \in T^*\}.$$

We construct the system

$$\Gamma_a = (N, K, T, (P_0, S_0), (P_1, S_1), (P_1', S_1'), \ldots, (P_n, S_n), (P_n', S_n')),$$

where

$$\begin{aligned} N = {} & \{S_0, S_{0,1}, S_{0,2}\} \cup \{S_i, S_{i,1}, S_i', S_{i,1}', S_i''\} \cup \\ & \cup \{A, A', A'', \overline{A} | A \in N_a\}, \\ P_0 = {} & \{S_0 \to x_i \overline{C}_i | C_i \to x_i \in P_{a,T}, 1 \le i \le m\} \cup \\ & \cup \{S_0 \to S_{0,1},\ S_{0,1} \to S_{0,2}\} \cup \\ & \cup \{S_{0,2} \to Q_i' | 1 \le i \le n\} \cup \\ & \cup \{A'' \to \overline{A} | A \in N_a\} \cup \\ & \cup \{S_a'' \to a\}, \\ P_i = {} & \{S_i \to S_{i,1},\ S_{i,1} \to S_i,\ S_i \to Q_0,\ \overline{B}_i \to v_i A_i\}, \\ P_i' = {} & \{S_i' \to S_{i,1}',\ S_{i,1}' \to S_i',\ S_i' \to S_i'',\ S_i'' \to u_i Q_i, \\ & A_i \to A_i',\ A_i' \to A_i''\}, \quad \text{for} \quad r_i : A_i \to u_i B_i v_i \in P_{a,N}, 1 \le i \le n. \end{aligned}$$

The query symbols Q_i are associated with components P_i, $0 \le i \le n$, and Q_i', with P_i', $1 \le i \le n$. All the symbols in N not in N_a are new and distinct from each other and from the symbols of $N_a \cup T$.

The string to be generated circulates among components as suggested in Figure 2.

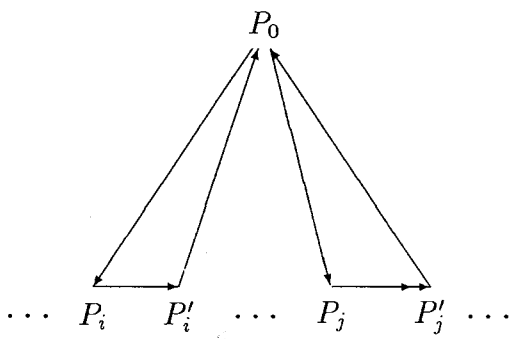

Figure 2.

The component P_0 starts and ends the derivation; at the first step, it simulates the terminal rules of P_a, at the last one, it introduces the symbol a (the obtained language is $L_r(\Gamma_a) = L(G_a)\{a\}$). In-between these steps, P_0 only prepares the current string for the components P_i, $1 \leq i \leq n$. Each pair P_i, P_i', $1 \leq i \leq n$, simulates a rule in P_a, namely that with the index i in $P_{a,N}$, r_i : $A_i \rightarrow u_i B_i v_i$. The component P_i introduces the right "context" v_i, whereas P_i' introduces the left "context" u_i. The derivation in Γ_a goes from the center of the string to its ends, on the path

$$w \longrightarrow u_{i_1} w v_{i_1} \longrightarrow u_{i_2} u_{i_1} w v_{i_1} v_{i_2} \longrightarrow \ldots \longrightarrow u_{i_k} \ldots u_{i_1} w v_{i_1} \ldots v_{i_k},$$

that is in the reversed way of producing strings in G_a:

$$u_{i_k} v_{i_k} \longrightarrow u_{i_k} u_{i_{k-1}} v_{i_{k-1}} v_{i_k} \longrightarrow \ldots$$
$$\ldots \longrightarrow u_{i_k} \ldots u_{i_1} v_{i_1} \ldots v_{i_k} \longrightarrow u_{i_k} \ldots u_{i_1} w v_{i_1} \ldots v_{i_k}.$$

Having these explanations in mind, let us examine in some details the possible derivations in Γ_a. When starting from its axiom, each component P_i, P_i', $1 \leq i \leq n$, can either choose an "active way" or an "inactive way". The first way leads to a query symbol, hence to the communication of a string, the second one means to use the "waiting rules" $S_i \rightarrow S_{i,1}$, $S_{i,1} \rightarrow S_i$ in P_i and $S_i' \rightarrow S_{i,1}'$, $S_{i,1}' \rightarrow S_i'$ in P_i'. In this way, the components P_i, P_i' can do nothing an even number of steps, being prepared after that for an "active way" again.

We start from the configuration $(S_0, S_1, S_1', \ldots, S_n, S_n')$. For the first step, we can distinguish several cases:

Case 1: The component P_0 introduces $x_s \overline{C}_s$, for some $C_s \rightarrow x_s \in P_{a,T}$, $1 \leq s \leq m$, and no component P_i, $1 \leq i \leq n$, introduces the query symbol Q_0. The derivation is immediately blocked, because P_0 cannot rewrite its (nonterminal) string.

Case 2: P_0 introduces the symbol $S_{0,1}$ and no component P_i, $1 \leq i \leq n$, introduces Q_0. We get

$$\Rightarrow_r (S_{0,1}, \ldots, S_{i,1},\ S_{i,1}' | S_i'', \ldots).$$

(We have indicated the alternative possibilities of using rules in P_i', for a generic i, $1 \leq g \leq n$, by separating the possible strings obtained in different variants by vertical bars.) We continue with

$$\Rightarrow_r (S_{0,2}, \ldots, S_i,\ S_i' | u_i Q_i, \ldots).$$

If some P_i' has introduced u_iQ_i, then the derivation is blocked after the communication, because P_i' cannot rewrite the string u_iS_i. Therefore, in order to continue, we must have obtained the configuration $(S_{0,2}, S_1, S_1', \dots, S_n, S_n')$. Now, P_0 will introduce Q_j' for some j, $1 \le j \le n$. If no P_i, $1 \le i \le n$, introduces at the same time the symbol Q_0, then the derivation is blocked: we communicate either $S_{j,1}'$ or S_j'' to P_0, and P_0 cannot rewrite these symbols. If some P_i has introduced Q_0, then first we communicate $S_{j,1}'$ or S_j'' to P_0, then to P_i and now the derivation is blocked because P_i cannot rewrite these symbols.

Case 3: P_0 introduces $S_{0,1}$ and some component P_i, $1 \le i \le n$, introduces Q_0. The symbol $S_{0,1}$ is communicated to P_i and the derivation is blocked, because P_i cannot rewrite $S_{0,1}$.

Therefore, at the first step of the derivation at least one component P_i must introduce Q_0, whereas P_0 must introduce some $x_s\overline{C}_s$, hence the obtained configuration and the next step must be

$$(x_s\overline{C}_s, \dots, Q_0, S_{i,1}'|S_i'', \dots) \Rightarrow_r (S_0, \dots, x_s\overline{C}_s, S_{i,1}'|S_i'', \dots).$$

Now, if $B_i \neq C_s$, then the derivation is blocked. Therefore, exactly those components P_i must introduce Q_0, that are identified by $x_s\overline{C}_s$; assume that such a component is P_i. Thus, the beginning of the derivation must be

$$(S_0, \dots, S_i, S_i', \dots) \Rightarrow_r (x_s\overline{C}_s, \dots, Q_0, S_{i,1}'|S_i'', \dots, S_{j,1}, S_{j,1}', \dots),$$

with all other components, like indicated for P_j, P_j' above, using the "waiting rules". We continue with

$$\begin{aligned}&\Rightarrow_r (S_0, \dots, x_s\overline{C}_s, S_{i,1}'|S_i'', \dots, S_{j,1}, S_{j,1}'), \dots\\ &\Rightarrow_r (x_t\overline{C}_t|S_{0,1}, \dots, x_sv_iA_i, S_i'|u_iQ_i, \dots, S_j, S_j', \dots).\end{aligned}$$

The derivation is blocked when P_i' has introduced S_i' (no further derivation step is possible in P_i), and similarly when P_0 has introduced a new string $x_t\overline{C}_t$. Therefore we have

$$\begin{aligned}&(S_{0,1}, \dots, x_sv_iA_i, u_iQ_i, \dots, S_j, S_j', \dots)\\ &\Rightarrow_r (S_{0,1}, \dots, S_i, u_ix_sv_iA_i, \dots, S_j, S_j', \dots)\\ &\Rightarrow_r (S_{0,2}, \dots, S_{1,i}|Q_0, u_ix_sv_iA_i', \dots, S_{j,1}|Q_0, S_{j,1}'|S_j'', \dots).\end{aligned}$$

The derivation is blocked when any Q_0 appears ($S_{0,2}$ cannot be rewritten in other components than P_0), as well as when some P_j' has introduced S_j'' : at the next step it will introduce Q_j, and the received string cannot be rewritten.

Therefore, when P_i' works, all components P_i, P_j, S_j', $1 \leq j \leq n$, $j \neq i$, must use "waiting rules". Thus, we have

$$\begin{aligned}&\Rightarrow_r (S_{0,2}, \ldots, S_{i,1}, u_i x_s v_i A_i', \ldots, S_{j,1}, S_{j,1}', \ldots)\\ &\Rightarrow_r (Q_k', \ldots, S_i, u_i x_s v_i A_i'', \ldots, S_j, S_j', \ldots).\end{aligned}$$

If $k \neq i$, then the derivation will be blocked. If $k = i$, then we get

$$\begin{aligned}&\Rightarrow_r (u_i x_s v_i A_i'', \ldots, S_i, S_i', \ldots, S_j, S_j', \ldots)\\ &\Rightarrow_r (u_i x_s v_i \overline{A}_i, \ldots, S_{g,1}|Q_0, S_{g,1}'|S_g'', \ldots), \quad (*)\end{aligned}$$

where g is a generic index, $1 \leq g \leq n$.

We are in a situation similar to that after the first step of the derivation, but having on the first component the string $u_i x_s v_i \overline{A}_i$ corresponding to two rules in P_a,

$$A_i \to u_i B_i v_i,\ C_s \to x_s, \quad \text{for} \quad B_i = C_s.$$

They can produce the derivation in G

$$A_i \Rightarrow u_i B_i v_i \Rightarrow u_i x_s v_i,$$

In order not to block the derivation, we must have exactly one occurrence of Q_0 in the configuration $(*)$, namely on a position g such that $A_i = B_g$; correspondingly, P_g' must have S_g'' as a current string.

Consequently, we can continue the walk in the graph in Figure 2, at each cycle (P_0, P_i, P_i') simulating the rule r_i in $P_{a,N}$. When the string on the first component is of the form wS_a'', hence a rule $S_a \to u_t X v_t$ has been simulated, P_0 can finish the derivation using $S_a'' \to a$.

From the previous explanations, it should be clear that $L_r(\Gamma_a) = L(G_a)\{a\}$, which completes the proof. □

From [2], Theorem 7.11, we know that each language in *CPC(RL)* is semi-linear. The second example in the previous section proves that there are non-semilinear languages in *NCPC(REG)*. Consequently, *NCPC(REG)* – *CPC(RL)* $\neq \emptyset$. Also the reverse difference is non-empty, hence the two families are incomparable.

THEOREM 2: *CPC(REG)* – *NCPC(RL)* $\neq \emptyset$.

Proof: Let

$$L = \{a^n b^n | n \geq 1\}^+ \{a\}.$$

This language is in *CPC(REG)*, because it can be generated by

$$\begin{aligned}\Gamma &= (\{S\}, K, \{a, b\}, (P_1, S), (P_2, S)),\\ P_1 &= \{S \to aS,\ S \to aQ_2,\ S_2 \to a\},\\ P_2 &= \{S \to bS\}.\end{aligned}$$

Any returning derivation in Γ has the following form:

$$\begin{aligned}&(S, S) \Rightarrow_r^* (a^{n_1}Q_2, b^{n_1}S) \Rightarrow_r (a^{n_1}b^{n_1}S, S)\\ &\Rightarrow_r^* (a^{n_1}b^{n_1}a^{n_2}Q_2, b^{n_2}S) \Rightarrow_r (a^{n_1}b^{n_1}a^{n_2}b^{n_2}S, S)\\ &\Rightarrow_r^* (a^{n_1}b^{n_1} \dots a^{n_k}b^{n_k}S, S) \Rightarrow_r (a^{n_1}b^{n_1} \dots a^{n_t}b^{n_t}a, bS),\end{aligned}$$

for some $t \geq 1$, $n_i \geq 1$, $1 \leq i \leq t$. Consequently, $L_r(\Gamma) = L$.

We shall now prove that $L \notin NCPC(RL)$. The intuitive idea of the proof is that in a PC grammar system generating L we have a component i communicating its string arbitrarily many times to the master. Due to the non-returning mode of derivation, the string of this component i will grow from a communication to another one (at least it remains the same), hence the string produced by the system must have as substrings a non-decreasing sequence of strings. However, L contains strings which do not fulfil such a restriction, a contradiction.

Let us formalize the previous idea. Suppose that $L = L_{nr}(\Gamma)$, for some centralized PC grammar system $\Gamma = (N, K, \{a, b\}, (P_1, S_1), \dots, (P_r, S_r))$, with right-linear components. Because $L \notin REG$, we must have $r \geq 2$. Since the system is centralized, only communications from P_j to P_1 are performed, for $2 \leq j \leq r$. Due to the non-returning mode of working, after communicating, each P_j, $2 \leq j \leq r$, keeps a copy of its sentential form and continues to rewrite it.

For each word $w \in L$, $w = a^{n_1}b^{n_1} \dots a^{n_t}b^{n_t}a$, with $t \geq 1, n_1, \dots, n_t \geq 1$, we call the i-th *block* of w the subword $a^{n_i}b^{n_i}$, $1 \leq i \leq t$.

Let $w \in L$, $w = a^{n_1}b^{n_1} \dots a^{n_t}b^{n_t}a, t \geq 1, n_1, \dots, n_t \geq 1$, and D be a derivation of w,

$$D\ : (S_1, \dots, S_r) \Rightarrow_{nr}^* (w, \alpha_2, \dots, \alpha_r),$$

for $\alpha_j \in (N \cup \{a, b\})^*$, $2 \leq j \leq r$. For i, $1 \leq i \leq t$, let D_i be the subderivation of D which produces the i-th block of w in the string of the component D_1. Consequently, D_i has the form

$$D_i\ : (w_1A_1, w_2A_2, \dots, w_rA_r) \Rightarrow_{nr}^* (w_1ua^{n_i}b^{n_i}u'B_1, w_2'B_2, \dots, w_r', B_r),$$

where $w_1 u = a^{n_1}b^{n_1}\dots a^{n_{i-1}}b^{n_{i-1}}$, $u' \in Pref(a^{n_{i+1}}b^{n_{i+1}}\dots a^{n_t}b^{n_t}a)$, $w_j, w'_j \in \{a,b\}^*$, $A_j, B_j \in N\cup\{\lambda\}$, $2 \le j \le r$, $A_1 \in N\cup K$, $B_1 \in N\cup\{\lambda\}$, and, moreover, D_i is minimal, in the sense that both the first and the last derivation step of D_i introduce at least one terminal symbol in the i-th block of w. For a triple (w, D, i) as above, we denote by k the number of communication steps in D_i where the transmitted string contains terminal symbols, by m the maximum number of terminal symbols introduced in the component P_1 at a communication step, considering only symbols which contribute to the i-th block, and by p the number of all terminal symbols generated by the component P_1 and which become a part of $a^{n_i}b^{n_i}$.

ASSERTION 1: There is a natural number $k_0 \ge 1$ such that, for each triple (w, D, i) as above, we have

$$p \le k_0(k+1)(km+1).$$

Proof of Assertion 1: Assume that this assertion is not true. Then there is a triple (w, D, i) for which

$$p > q_0(6q_1^k + 1)(k+1)(km+1),$$

where

$$q_0 = \max\{|x| \mid A \to xB \in P_1,\ x \in \{a,b\}^*,\ A \in N,\ B \in N \cup K \cup \{\lambda\}\},$$
$$q_1 = card(N \cup K \cup \{\lambda\}).$$

Between the communication steps that introduce at least one terminal symbol and possibly before the first communication and after the last one of D_i with this property, the component P_1 produces at most $k+1$ strings of terminal symbols which participate to obtaining $a^{n_i}b^{n_i}$. The sum of the lengths of these strings is p, therefore there is at least one such string x with $|x| \ge \frac{p}{k+1}$.

Let D' be the subderivation of D_i which produces x. It follows that D' has the form

$$D' : (z_1X_1, z_2X_2, \dots, z_rX_r) \Rightarrow^*_{nr} (z_1v_1xv_2Y_1, z_2x_2Y_2, \dots, z_rx_rY_r),$$

where $X_j, Y_j \in N \cup \{\lambda\}, z_j, x_j \in \{a,b\}^*, 2 \le j \le r$, $z_1, v_1, v_2 \in \{a,b\}^*, X_1 \in N, Y_1 \in N \cup K$, and v_1 maybe different from λ if D' is the subderivation of D_i which starts with the same configuration as D_i and stops before the first communication step, v_2 may be different from λ if D' is the subderivation of D_i which follows after the last communication step, and $v_1 v_2 = \lambda$ if D' is a subderivation between two communication steps. Note that, D' contains rewriting steps and communications when only nonterminals are transmitted. Consequently, x is a subword of $a^{n_i} b^{n_i}$. Moreover, $|x_j| \leq m$ for those j, $2 \leq j \leq r$, for which communications from P_j to P_1 will follow in D_i (this follows from the definition of m).

We divide the subderivation D' in subderivations such that at least $km+1$ terminal symbols are produced in each of them in the string of P_1. Since q_0 is the maximal number of symbols that can be introduced in P_1 at a rewriting step ($q_0 \geq 1$ because $p > 0$), it follows that we can impose, in addition, the condition that any of the generated strings of these subderivations does not have more than $q_0(km+1)$ symbols, without losing the first condition. Then, the total number of these subderivations is at least $\frac{|x|}{q_0(km+1)}$. As $|x| \geq \frac{p}{k+1}$ and $p > q_0(6q_1^r + 1)(k+1)(km+1)$, it follows that we have at least $6q_1^r + 1$ such subderivations. Since q_1^r is the maximum number of different r-tuples $(Y_1, \ldots, Y_r)$, $Y_i \in N \cup K \cup \{\lambda\}$, $1 \leq i \leq r$, it follows that there are $(Z_1, \ldots, Z_r)$, $Z_i \in N \cup K \cup \{\lambda\}$, $1 \leq i \leq r$, and seven different configurations $C_1, \ldots, C_7$, not two of them in the same subderivation as defined above, such that C_s has the nonterminal Z_j in the component j (and maybe terminal symbols), $1 \leq j \leq r$, $1 \leq s \leq 7$. Assume that these seven configurations occur in D' in the order of their indices. Then in at least one of the derivations $C_2 \Rightarrow_{nr}^* C_4$ and $C_4 \Rightarrow_{nr}^* C_6$, occurrences of only one terminal symbol are introduced (because in $C_2 \Rightarrow_{nr}^* C_6$ only terminal symbols which contribute to $a^{n_i} b^{n_i}$ are introduced; in order to be sure of this we have left apart the configurations C_1 and C_7— they could be the first and the last ones of D_i). Let us assume, without loss of the generality, that $C_2 \Rightarrow_{nr}^* C_4$ is this derivation. Then $C_2 = (y_1 Z_1, y_2 Z_2, \ldots, y_r Z_r)$, $C_4 = (y_1 u_1 Z_1, y_2 u_2 Z_2, \ldots, y_r u_r Z_r)$, $u_1 \in \{\alpha\}^*$, $\alpha \in \{a, b\}$. Clearly, $|u_1| \geq km + 1$. Replacing in D the subderivation $C_2 \Rightarrow_{nr}^* C_4$ by the subderivation obtained by repeating $C_2 \Rightarrow_{nr}^* C_4$ for q times, $q \geq 2$, we obtain a terminal derivation D'', which generates a word $w' \in L$ having the first $i-1$ blocks identical with those in w. The i-th block of w' has in addition to the i-th block of $w(q-1)|u_1|$ occurrences of α generated by P_1 and at most $(q-1)km$ occurrences of symbols introduced by the communication steps (that follow after the iteration of the subderivation $C_2 \Rightarrow_{nr}^* C_4$).

Since $(q-1)|u_1| \geq (q-1)(km+1) > (q-1)km$, it follows that the number of occurrences of a in the i-th block of w' is not equal to the number of occurrences of b in that block. This contradicts the relation $w' \in L$, hence concludes the proof of Assertion 1.

ASSERTION 2: There is a natural number $k_1 \geq 1$ such that

$$k \leq k_1(rm + 1),$$

for each triple (w, D, i) as above.

Proof of Assertion 2: Assume that this is not true, hence there is a triple (w, D, i) for which

$$k > 5(q_1^r + 1)(rm + 1).$$

Consider the subderivation D_i of D

$$D_i \ : (w_1 A_1, w_2 A_2, \ldots, w_r, A_r) \Rightarrow_{nr}^{*} (w_1 u a^{n_i} b^{n_i} u' B_1, w_2' B_2, \ldots, w_r' B_r).$$

We do not take into account the first and the last communication steps (when strings from $b^+ a^+ (N \cup \{\lambda\})$ can be communicated) and one step when a string from $a^+ b^+ N$ can be communicated. Then $k - 3$ communication steps remain, when strings in $\{a, b\}^+ N$ are communicated. It is clear that there is $\alpha \in \{a, b\}$ such that the number k', of communication steps when strings in $\alpha^+ N$ are transmitted, is at least $\frac{k-3}{2}$. Consequently,

$$k' \geq \frac{2(q_1^r + 1)(rm + 1)}{2} + \frac{3(q_1^r + 1)(rm + 1)}{2} \geq (q_1^r + 1)(rm + 1).$$

These communications are consecutive (all possible intermediate communications are transmitting only nonterminal symbols). Denote by D' the subderivation of D_i which starts with the first and finishes with the last of these communication steps. Since D' has at least $(q_1^r + 1)(rm + 1)$ communication steps when strings from $\alpha^+ N$ are transmitted and, because between two communications at least one rewriting step is performed, it follows that there is a subderivation D_1' of D' with $q_1^r + 1$ communications; moreover

$$D_1' \ : (z_1 X_1, \ldots, z_r X_r) \Rightarrow_{nr}^{*} (z_1 x_1 Y_1, z_2' Y_2, \ldots, z_r' Y_r),$$

$z_1 \in \{a, b\}^*$, $x_1 \in \alpha^*, X_1, Y_1 \in N \cup K, z_j, z_j' \in \{a, b\}^*, X_j, Y_j \in N \cup \{\lambda\}, 2 \leq j \leq r$, and for the components $P_j, 2 \leq j \leq r$, which communicate strings to P_1 in D' we have $z_j' = z_j$. Indeed, the number of derivations when in at least one component P_j which communicates to P_1 in D', at least one terminal symbol is produced, is less than or equal to $(r - 1)m$, because m is the maximal number of terminal symbols that can appear in such a component P_j in D' and $(q_1^r + 1)(rm + 1) \geq (q_1^r + 1)(r - 1)m$.

The derivation D_1' has $q_1^r + 1$ communication steps, hence there are two configurations C_1 and C_2 which have the same nonterminals in the corresponding components and the derivation $C_1 \Rightarrow_{nr}^* C_2$ contains at least one communication step of those mentioned above. Then $C_1 = (y_1 Z_1, y_2 Z_2, \ldots, y_r Z_r)$, $C_2 = (y_1 u_1 Z_1, y_2' Z_2, \ldots, y_r' Z_r)$, for some $y_1 \in \{a, b\}*, u_1 \in a^*, Z_1 \in N \cup K, Z_j \in N \cup \{\lambda\}, y_j, y_j' \in \{a, b\}^*, 2 \leq j \leq r$, and for each $j, 2 \leq j \leq r$, such that there is a communication from P_j to P_1 in D', we have $y_j' = y_j$. It follows that $|u_1| > 0$ because at the communication step in D' some terminal symbols have been introduced in P_1. If we replace in D the subderivation $C_1 \Rightarrow_{nr}^* C_2$ by the derivation obtained by iterating it q times, $q \geq 2$, then we obtain a terminal derivation D'', generating a string w'. This string must be in L and has the first $i-1$ blocks identical with those of w. In the i-th block, w' has in addition to w the substring u_1^{q-1} (due to the form of C_1 and C_2, the substring communicated in D_i are not modified after iterating $C_1 \Rightarrow_{nr}^* C_2$). As u_1^{q-1} is non-empty and contains occurrences of only one symbol, it follows that w' does not have the same number of occurrences of a and b in the i-th block, a contradiction which completes the proof of Assertion 2.

ASSERTION 3: Let D be a terminal derivation in Γ in which a communication of a string $xa^s b^n ayA$ is performed, for some $x, y \in \{a, b\}^*, A \in N \cup \{\lambda\}, n \geq 1, s \geq 0$. Then

$$s < q_2(q_1^r + 3),$$

where

$$q_2 = \max\{|u| | B \to uC \in P_j, u \in \{a, b\}^*, B \in N, C \in N \cup \{\lambda\}, 1 \leq j \leq n\}.$$

Proof of Assertion 3: Assume that the assertion is not true, hence there is a terminal derivation D' in Γ of the form

$$\begin{aligned} D' &: (S_1, \ldots, S_r) \Rightarrow_{nr}^* (zQ_j, \ldots, xa^s b^n ayA, \ldots) \\ &\Rightarrow_{nr} (zxa^s b^n ayA, \ldots, xa^s b^n ayA, \ldots) \Rightarrow_{nr}^* (w, \delta_2, \ldots, \delta_r), \end{aligned}$$

where $x, y, z \in \{a, b\}^*, A \in N \cup \{\lambda\}, \delta_i \in (N \cup \{a, b\})^*, 2 \leq i \leq r$, for some j, $2 \leq j \leq r, n \geq 1$, and $s \geq q_2(q_1^r + 3)$. Then in the derivation $S_j \Rightarrow_{nr}^* xa^s b^n ayA$ in the component P_j at least $q_1^r + 1$ rewriting steps using rules of the form $B \to uC, u \in a^+, B, C \in N$, were necessary for obtaining the string a^s. At least two configurations of D from the beginning of these rewriting steps have the same nonterminals in the corresponding components. Assume that $C_1 = (z_1X_1, z_2X_2, \ldots, xa^tX_j, \ldots, z_rX_r)$ and $C_2 = (z'_1X_1, z'_2X_2, \ldots, xa^ta^vX_j, \ldots, z'_rX_r)$, are such configurations, z_i, $z'_i \in \{a,b\}^*$, $1 \leq i \leq r$, $i \neq j$, $X_i \in N \cup \{\lambda\}$, $1 \leq i \leq r$, X_1, $X_j \neq \lambda$, $t \geq 0$, $v \geq 1$. We replace in D' the subderivation $C_1 \Rightarrow^*_{nr} C_2$ by an iteration of it for q times. In this way a terminal derivation D'_q is obtained,

$$D' : (S_1, \ldots, S_r) \Rightarrow^*_{nr} (z'Q_j, \ldots, xa^{s+qv}b^n ayA, \ldots)$$
$$\Rightarrow_{nr} (z'xa^{s+qv}b^n ayA, \ldots, xa^{s+qv}b^n ayA, \ldots) \Rightarrow^*_{nr} (w'_q, \delta'_2, \ldots, \delta'_r),$$

where $z', w'_q \in \{a,b\}^*, \delta'_i \in (N \cup \{a,b\})^*, 2 \leq i \leq r$. Consequently, $w'_q \in L, q \geq 1$. For $q = n, w'_n$ contains the subword $a^{s+nv}b^n a$, which is not in L, a contradiction completing the proof of Assertion 3.

Let now w be an arbitrary word in L, $w = a^{n_1}b^{n_1} \ldots a^{n_t}b^{n_t}a, t \geq 1$, D a derivation of w and D_i a subderivation which produces the i-th block of w, for some i, $1 \leq i \leq t$. According to Assertions 1 and 2, we have

$$p \leq k_0(k+1)(km+1), \tag{1}$$

$$k \leq k_1(rm+1), \tag{2}$$

where p, k, m are the numbers associated to the triple (w, D, i) and k_0, k_1 are constants.

The length of the i-th block of w is $2n_i$; p of the symbols appearing in this block are introduced by P_1 during the subderivation D_i, and the other symbols by the communication steps when strings xX, $x \in \{a,b\}^+, X \in N \cup \{\lambda\}$, are transmitted. As the number of symbols in x which contribute to $a^{n_i}b^{n_i}$ is at most m, it follows that

$$2n_i \leq p + km.$$

Using relations (1) and (2), we obtain

$$\begin{aligned} 2n_i &\leq k_0(k+1)(km+1) + km \leq k_0(k+2)(km+1) \leq \\ &\leq k_0(k_1(rm+1)+2)(k_1(rm+1)m+1) \leq \\ &\leq k_0(k_1(rm+1)+2)(k_1(rm+1)^2+1) \leq \\ &\leq k_0(k_1(rm+1)+2)^3 = k_0(k_1rm+k_1+2)^3. \end{aligned}$$

Consequently, $2n_i \leq k_0(k_1rm + k_1 + 2)^3$. It follows that

$$k_1rm + k_1 + 2 \geq \sqrt[3]{\frac{2n_i}{k_0}}, \tag{3}$$

$$m \geq \frac{\sqrt[3]{\frac{2n_i}{k_0}} - k_1 - 2}{k_1 r}. \tag{4}$$

Consider the mapping $f : \mathbf{N} \to \mathbf{R}$ defined by

$$f(n) = \frac{\sqrt[3]{\frac{2n}{k_0}} - k_1 - 2}{k_1 r}.$$

This is an increasing mapping and $\lim_{n\to\infty} f(n) = \infty$. Denote by n_0 the smallest natural number such that $f(n_0) \geq 1$.

From (4) we have $m \geq f(n_i)$ and $f(n_i)$ does not depend on w or on D_i, but only on the length of the i-th block of w. Consequently, for any word $w \in L$ and any derivation D of a derivation producing it, for obtaining a block $a^n b^n$ of w, $n \geq n_0$, at least one communication step is performed, when a string of the form $xX, x \in \{a,b\}^*, X \in N \cup \{\lambda\}$, is transmitted, such that the number of symbols appearing in x which contribute to $a^n b^n$ is at least equal to $f(n)$. Note that $f(n) < 2n$.

Denote $k_2 = q_2(q_1^r + 3)$. As $\lim_{n\to\infty} f(n) = \infty$, there are natural numbers $n_1, n_2, \ldots, n_{2r-1}$ such that $n_1 \geq \max(k_2, n_0)$ and for each i, $1 \leq i \leq 2r-2$, $f(n_{i+1}) > 2n_i$. Since $f(n) < 2n$, for all natural numbers n, it follows that $n_{2r-1} > n_{2r-2} > \ldots > n_1$, hence $n_i \geq k_2, 1 \leq i \leq 2r-1$. Let $w = a^{n_{2r-1}} b^{n_{2r-1}} \ldots a^{n_1} b^{n_1} a$ in L, and let D be a derivation of w. For each i, $1 \leq i \leq 2r-1$, there is a component $P_{j_i}, 2 \leq j_i \leq r$, which communicates to P_1 a string of the form $z_i = x_i a^{m'_i} b^{m_i - m'_i} y_i X_i$ with $x_i, y_i \in \{a,b\}^*, X_i \in N \cup \{\lambda\}, m'_i \geq 0, m_i \geq f(n_i)$, and $a^{m'_i} b^{m_i - m'_i}$ contributes to the i-th block. Denote by p_i this communication step, $1 \leq i \leq 2r-1$. It follows that $2n_i \geq m_i > 2n_{i-1}$, $2 \leq i \leq 2r-1$.

Assume that $j_i = j_s$ for some $i, s, 3 \leq i \leq 2r-1, 1 \leq s \leq i-2$. If $p_i = p_s$, then at this communication step also the $(i-1)$-th block is introduced in the string of P_1, hence $z_{j_i} = xba^{n_{i-1}} b^{n_{i-1}} ayX_{j_i}, x, y \in \{a,b\}^*$. But $n_{i-1} \geq k_2$ and according to Assertion 3, a string like z_{j_i} cannot be communicated, a contradiction. It follows that $p_i \neq p_s$. Because the step p_s is performed after p_i (and the system is non-returning), the string $x_i a^{m'_i} b^{m_i - m'_i} y_i$ is a subword of z_{j_s}. Hence, when z_{j_s} is communicated, also the string $a^{m'_i} b^{m_i - m'_i}$ is introduced, but this is not a subword of any block $s, \ldots, i-1$, because $m_i \geq 2n_{i-1} > \ldots > 2n_s$. Consequently, the communication step p_s contributes both to the s-th and to the i-th blocks of w and we obtain the same conclusion as in the previous case.

In conclusion, $j_i \neq j_s, 3 \leq i \leq 2r-1, 1 \leq s \leq i-2$. It follows that $j_{2r-1}, j_{2r-3}, \ldots, j_1$ are r different numbers. On the other hand, all of them are in the set $\{2, \dots, r\}$, which contains only $r - 1$ elements. Contradiction. □

4. REGULAR VERSUS RIGHT-LINEAR RULES

In many papers (this is true, for instance, for [2]), for "regular" PC grammar systems one works with right-linear rules, but the proofs are given (when possible) for the stronger variant: using strictly regular rules in examples and right-regular rules in proofs which can work in a general set-up. However, up to now no comparison of the two types of systems is made. We will show that such a comparison is necessary, there are cases when the right-linear rules are strictly more powerful than the regular ones, a situation which is quite unfrequent in formal language theory.

THEOREM 3: *CPC(REG)* $\subset$ *CPC(RL).*

Proof: The inclusion is trivial, we have to prove only its strictness. To this aim, we consider the language

$$L = \left\{a^m w d^m | m \geq 1, w \in \{b^2, c^3\}^*, \frac{|w|_b}{2} + \frac{|w|_c}{3} = m\right\}.$$

Consider the centralized PC grammar system

$$\Gamma = (\{S_1, S_2, S_3, A, B, C, D\}, K, \{a, b, c, d\}, (P_1, S_1), (P_2, S_2), (P_3, S_3)),$$

with

$$\begin{aligned} P_1 &= \{S_1 \to aA, S_1 \to aQ_2, A \to aA, A \to aQ_2, B \to Q_3, D \to d\}, \\ P_2 &= \{S_2 \to b^2B, S_2 \to c^3B, B \to b^2B, B \to c^3B\}, \\ P_3 &= \{S_3 \to C, C \to D, D \to dD\}. \end{aligned}$$

Each returning derivation in Γ is of one of the following forms:

$$\begin{aligned} (1) \quad (S_1, S_2, S_3) &\Rightarrow_r^* (a^m Q_2, wB, d^{m-2}D) \Rightarrow_r^* (a^m wB, S_2, d^{m-2}D) \\ &\Rightarrow_r (a^m wQ_3, xB, d^{m-1}D) \Rightarrow_r (a^m wd^{m-1}D, xB, S_3) \\ &\Rightarrow_r (a^m wd^m, xx'B, C), \end{aligned}$$

where $m \geq 2$, $w \in \{b^2, c^3\}^+$, $\frac{|w|_b}{2} + \frac{|w|_c}{3} = m$, and x, $x' \in \{b^2, c^3\}$,

$$\begin{aligned} (2) \quad (S_1, S_2, S_3) &\Rightarrow_r^* (aQ_2, wB, C) \Rightarrow_r (awB, S_2, C) \\ &\Rightarrow_r (awQ_3, xB, D) \Rightarrow_r (awD, xB, S_3) \Rightarrow_r (awd, xx'B, C), \end{aligned}$$

where $w, x, x' \in \{b^2, c^3\}$.

Consequently, $L_r(\Gamma) = L$, hence $L \in CPC(RL)$.

Let us now assume that $L = L_r(\Gamma)$ for some centralized PC grammar system with regular components, $\Gamma = (N, K, T, (P_1, S_1), \ldots, (P_n, S_n))$, $T = \{a, b, c, d\}$.

ASSERTION 1: There is a natural number k such that for any derivation $(S_1, \ldots, S_n) \Rightarrow_r^* (a^m w d^m, \alpha_2, \ldots, \alpha_n)$ in Γ, where $m \geq 1, w \in \{b^2, c^3\}^+$, the number of the communication steps which contribute to w (hence the steps when strings of the form $xX, x \in a^*\{b, c\}^+ d^*, X \in N$, are communicated) is less than or equal to k.

Proof of Assertion 1: Assume that there is no natural numbers k satisfying the required condition. Then there is a derivation $D : (S_1, \ldots, S_n) \Rightarrow_r^* (a^m w d^m, \alpha_2, \ldots, \alpha_n)$, $w \in \{b^2, c^3\}^+$, for which the number of the communication steps that participate to obtaining the substring w is at least $(p+1)^n + 2$, where $p = card(N \cup K)$. Let C_1 be the configuration obtained after the first such communication step and C_2 that obtained before the last such communication step. Then the string in the first component, both in C_1 and in C_2, is of the form $a^m x A, A \in N \cup K, x \in \{b, c\}^+$, and the strings communicated during the subderivation $C_1 \Rightarrow_r^* C_2$ are of the form $yX, y \in \{b, c\}^+, X \in N$. Also, the subderivation $C_1 \Rightarrow_r^* C_2$ contains at least $(p+1)^n$ steps (communication or rewriting steps), hence it has at least $(p+1)^n + 1$ configurations. As the maximal number of different n-tuples of the form $(A_1, \ldots, A_n), A_i \in N \cup K \cup \{\lambda\}, 1 \leq i \leq n$, is $(p+1)^n$, it follows that there are two configurations $C_3 = (x_1 A_1, \ldots, x_n A_n)$ and $C_4 = (y_1 A_1, \ldots, y_n A_n)$, such that $C_1 \Rightarrow_r^* C_3 \Rightarrow_r^* C_4 \Rightarrow_r^* C_2$. Since the grammar system is centralized and has regular productions, it follows that $y_1 = x_1 z_1, z_1 \neq \lambda$. From the definition of C_1 and C_2, we obtain $z_1 \in \{b, c\}^+$.

The derivation steps between C_3 and C_4 can be repeated s times, for any natural number s. At each iteration a symbol b or c is introduced in the first component. If, after these s iterations of the subderivation $C_3 \Rightarrow_r^* C_4$, the derivation is continued using the same steps as in $C_4 \Rightarrow_r^* C_2 \Rightarrow_r^* (a^m w d^m, \alpha_2, \ldots, \alpha_n)$, then we obtain a derivation of a terminal word of the form $a^m w' u, u \in T^*, w' \in \{b, c\}^+, |w'| \geq s$. Take $s = 4m$. Then $|w'| \geq 4m$ and

$$\frac{|w'|_b}{2} + \frac{|w'|_c}{3} \geq \frac{|w'|_b}{3} + \frac{|w'|_c}{3} = \frac{|w'|}{3} \geq \frac{4}{3}m > m.$$

Hence $a^m w' u \notin L$, a contradiction finishing the proof of Assertion 1.

For $s \geq 1, 2 \leq i \leq n, X, Y \in N$, we denote

$$A(s,i,X,Y) = \{t \geq 0 \mid \text{ there is } y \in \{b^2,c^3\}^+, |y| = s, |y|_b = t, \text{ and there is a derivation } X \Rightarrow_r^* yY \text{ in } P_i \text{ which will contribute to the generation of a string which will be communicated to } P_1 \text{ during a terminal derivation}\},$$

and for $s \geq 1, X \in N, Y \in N \cup K$, we denote

$$A(s,1,X,Y) = \{t \geq 0 \mid \text{ there is } y \in \{b^2,c^3\}^+, |y| = s, |y|_b = t, \text{ and there is a derivation } X \Rightarrow_r^* yY \text{ in } P_1 \text{ which is used in a terminal derivation in } \Gamma\}.$$

ASSERTION 2: $card(A(s,i,X,Y)) \leq 1$, $1 \leq i \leq n$, $X \in N$, $Y \in N \cup K$, $s \geq 1$.

Proof of Assertion 2: We consider first the case $i = 1$. Assume that there is $s_0 \geq 1$ and $X, Y \in N$ such that $card(A(s_0,1,X,Y)) \geq 2$. Then there is a derivation

$$D : (S_1, \ldots) \Rightarrow_r^* (xX, \ldots) \Rightarrow_r^* (xyY, \ldots) \Rightarrow^* (z, \ldots),$$

where $z \in L, x \in T^*, y \in \{b^2,c^3\}^+$, $|y| = s_0$ and between the configurations $(xX, \ldots)$ and $(xyY, \ldots)$ there is no communication step and, moreover, there is $y' \in \{b^2,c^3\}^+$ such that $X \Rightarrow_r^* y'Y$ in G_1, $|y'| = s_0$ and $|y'|_b \neq |y|_b$.

Then in the derivation D we can replace the subderivation $(xX, \ldots) \Rightarrow_r^* (xyY, \ldots)$ by $(xX, \ldots) \Rightarrow_r^* (xy'Y, \ldots)$ (only the rewriting steps from the first component are changed, the others remain the same). Thus we obtain a terminal derivation

$$D' : (S_1, \ldots) \Rightarrow_r^* (xX, \ldots) \Rightarrow_r^* (xy'Y, \ldots) \Rightarrow_r^* (z', \ldots),$$

where, if $z = xyu$, then $z' = xy'u, u \in T^*$. Since $z \in L_r(\Gamma) = L$ and $y \in \{b^2,c^3\}^+$, it follows that $z = a^m v_1 y v_2 d^m, v_1, v_2 \in \{b^2,c^3\}^+$, hence $z' = a^m v_1 y' v_2 d^m$. From the definition of L, since $z, z' \in L$, we obtain

$$\frac{|v_1yv_2|_b}{2} + \frac{|v_1yv_2|_c}{3} = m = \frac{|v_1y'v_2|_b}{2} + \frac{|v_1y'v_2|_c}{3},$$

hence,

$$\frac{|y|_b}{2} + \frac{|y|_c}{3} = \frac{|y'|_b}{2} + \frac{|y'|_c}{3}.$$

We know that $|y|_c = |y| - |y|_b = s_0 - |y|_b$ and $|y'|_c = |y'| - |y'|_b = s_0 - |y'|_b$. Replacing the corresponding terms in the equality above, we obtain $|y|_b = |y'|_b$, which contradicts our assumption. Hence Assertion 2 is proved for $i = 1$.

Assume now that there are i, $1 \leq i \leq n$, $s_0 \geq 1$, and $X, Y \in N$ such that $card(A(s_0, i, X, Y)) \geq 2$. Then there is a derivation in Γ

$$D : (S_1, \ldots, S_i, \ldots) \Rightarrow_r^* (x_1 B, \ldots, y_1 X, \ldots) \Rightarrow_r^* (x_1 x_2 C, \ldots, y_1 y Y, \ldots)$$
$$\Rightarrow_r^* (x_1 x_2 x_3 Q_i, \ldots, y_1 y y_2 Z, \ldots) \Rightarrow_r^* (x_1 x_2 x_3 y_1 y y_2 Z, \ldots, S_i) \Rightarrow_r^* (z, \ldots),$$

where $z \in L$, $x_1, x_2, x_3, y_1, y_2 \in T^*$, $y \in \{b^2, c^3\}^+$, $|y| = s_0$, $B \in N \cup K$, C, $Z \in N$, and between the configurations $(x_1 B, \ldots, y_1 X, \ldots)$ and $(x_1 x_2 x_3 Q_i, \ldots, y_1 y y_2 Z, \ldots)$ there is no communication step, and, moreover, there is $y' \in \{b^2, c^3\}^+$ such that $X \Rightarrow_r^* y' Y$ in P_i and $|y'| = s_0, |y'|_b \neq |y|_b$. Therefore, we can replace in D the subderivation $(x_1 B, \ldots, y_1 X, \ldots) \Rightarrow_r^* (x_1 x_2 C, \ldots, y_1 y Y, \ldots)$ by $(x_1 B, \ldots, y_1 Y, \ldots) \Rightarrow_r^* (x_1 x_2 C, \ldots, y_1 y' Y, \ldots)$ (the change is performed only in the component P_i). In this way we obtain a new derivation, which generates the terminal string $z' = x_1 x_2 x_3 y_1 y' y_2 u$, where $u \in T^*$, $z = x_1 x_2 x_3 y_1 y y_2 u$. Continuing as in the case $i = 1$ we obtain a contradiction; this completes the proof of Assertion 2.

For every $s \geq 1$, denote

$$A(s) = \bigcup_{i=1}^{n} \bigcup_{X, Y \in N} A(s, i, X, Y)$$

and

$$A'(s) = \{t \geq 0 | \text{ there is } y \in \{b^2, c^3\}^+, |y| = s, |y|_b = t\}.$$

Moreover, denote $k_0 = n \cdot p^2$. According to Assertion 2, $card(A(s)) \leq k_0$ for all $s \geq 1$. Let $s_0 = 6k_0$. Then $card(A'(s_0)) \geq k_0 + 1$. Indeed, let $x_j = b^{6j} c^{6(k_0 - j)}, 0 \leq j \leq k_0$. Then $x_j \in \{b^2, c^3\}^+$ and $|x|_j = 6k_0, 0 \leq j \leq k_0$. Since $card(A'(s_0)) > card(A(s))$, it follows that there is $t_0 \in A'(s_0) - A(s)$. Let $y \in \{b^2, c^3\}^+$ such that $|y| = s_0$ and $|y|_b = t_0$. Denote $m = \frac{|y|_b}{2} + \frac{|y|_c}{3}$ (m is a natural number, because $y \in \{b^2, c^3\}^+$). Clearly

$$2(k+1)m = \frac{|y^{2(k+1)}|_b}{2} + \frac{|y^{2(k+1)}|_c}{3}$$

and $y^{2(k+1)} \in \{b^2, c^3\}^+$, where k is the number in Assertion 1. Let $z = a^{2(k+1)m}y^{2(k+1)}d^{2(k+1)m}$. Obviously, $z \in L = L_r(\Gamma)$. According to Assertion 1, any derivation of z has at most k communication steps which contribute to the string $y^{2(k+1)}$. It follows that there is a substring y' of $y^{2(k+1)}$, $|y'| \geq |y^2|$, such that either y' is generated entirely using rules in P_1, or $y'Y'$ is generated in a component $P_i, 2 \leq i \leq n$, and then communicated to P_1, for $Y' \in N$. As $|y'| \geq |y^2| = 2s_0$ and y' is a substring of $y^{2(k+1)}$, it follows that y is a subword of y' and, according to the definition of the sets $A(s, i, X, Y), s \geq 1, 1 \leq i \leq n, X \in N, Y \in N \cup K$, it follows that there are $X \in N, Y \in N \cup K$, such that y satisfies the condition in the definition of the set $A(s_0, i, X, Y)$. Consequently, $|y|_b \in A(s_0)$; however $|y|_b = t_0 \notin A(s_0)$, a contradiction. $\square$

We dot not know whether the previous result is true also for non-centralized systems. Somehow surprisingly, the following counterpart of it is true, proving that in right-linear systems the chain rules are important, not the rules $A \to x, A \to xB, A \to xQ_i$ with $|x| \geq 2$.

THEOREM 4: *For every PC grammar system Γ with right-linear rules, centralized or not, there is a PC grammar system Γ', of the same type as Γ as concerns centralization, with rules of the forms $A \to cB, A \to B, A \to c$, with A nonterminal, B nonterminal or query symbol, and c terminal, such that $L_f(\Gamma) - \{\lambda\} = L_f(\Gamma'), f \in \{r, nr\}$.*

Proof: Take $\Gamma = (N, K, T, (P_1, S_1), \ldots, (P_n, S_n))$. Denote

$$q = \max\{|x| | A \to x \in P_i, 1 \leq i \leq n, A \in N, x \in T^*(N \cup K \cup \{\lambda\})\}.$$

If $q \leq 1$, then Γ is already of the desired form. Assume that $q \geq 2$. We construct the system

$$\Gamma' = (N', K, T, (P_1', S_1), \ldots, (P_n', S_n)),$$

with

$$\begin{aligned} N' = {} & N \cup \{[r, j] | r : A \to \alpha_1 \ldots \alpha_s \in P_i, A \in N, \\ & \alpha_t \in T, 1 \leq t \leq s-1, \alpha_s \in N \cup K \cup T, s \geq 1, 1 \leq j \leq q\}, \\ P_i' = {} & \{A \to [r, 1], [r, 1] \to [r, 2], \ldots, [r, q-s] \to [r, q-s+1], \\ & [r, q-s+1] \to \alpha_1[r, q-s+2], [r, q-s+2] \to \alpha_2[r, q-s+3], \\ & \ldots, [r, q-1] \to \alpha_{s-1}[r, q], [r, q] \to \alpha_s | r : A \to \alpha_1 \ldots \alpha_s \in P, \\ & A \in N, \alpha_t \in T, 1 \leq t \leq s-1, \alpha_s \in N \cup K \cup T, s \geq 1\}, \end{aligned}$$

for each i, $1 \leq i \leq n$.

The equality $L_f(\Gamma) - \{\lambda\} = L_f(\Gamma')$ is obvious, for each $f \in \{r, nr\}$: each rule $r : A \rightarrow \alpha_1 \ldots \alpha_s$ of Γ is simulated in Γ' by exactly $q+1$ rules, starting with $A \rightarrow [r, 1]$, and ending with $[r, q] \rightarrow \alpha_s$. Starting from a configuration of Γ (initially we have $(S_1, \ldots, S_n)$), Γ' produces in this way another configuration of Γ. The query symbols can be introduced only by rules $[q, r] \rightarrow \alpha_s$, hence the communication steps are performed as in Γ, without involving symbols $[r, j]$. The type of derivation – returning or non-returning – plays no role in this argument. Clearly, Γ' is centralized when Γ is centralized, which completes the proof. $\square$

The previous theorem corresponds to the obvious fact that each right-linear grammar is equivalent (modulo λ) with a grammar having rules of the forms $A \rightarrow cB$, $A \rightarrow B$, $A \rightarrow c$. In the case of context-free derivations in a Chomsky grammar, also the rules of the form $A \rightarrow B$ can be eliminated. In the case of PC grammar systems the difference is due to the synchronization of rewriting steps.

Note that the previous construction does not work for PC grammar system with context-free components, because of multiple queries: it is necessary that in each rule $A \rightarrow x$ with $|x| \geq 2$ we have $x = x'\alpha$ with $x' \in (N \cup T)^*$.

REFERENCES

1. L. Cai, The computational complexity of PCGS with regular components, *Proc. of Developments in Language Theory Conf.*, Magdeburg, 1995.
2. E. Csuhaj-Varju, J. Dassow, J. Kelemen and Gh. Păun, *Grammar Systems. A Grammatical Approach to Distribution and Cooperation,* Gordon and Breach, London, 1994.
3. J. Dassow and Gh. Păun, *Regulated Rewriting in Formal Language Theory*, Springer, Berlin, Heidelberg, 1989.
4. J. Dassow, Gh. Păun and G. Rozenberg, Generating languages in a distributed way: Grammar systems, in *Handbook of Formal Languages* (G. Rozenberg, A. Salomaa, eds.), Springer-Verlag, Berlin, Heidelberg, 1997.
5. S. Dumitrescu, Non-returning PC grammar systems can be simulated by returning systems, *Theoretical Computer Sci.*, 1996, *165*, pp. 463-474.
6. S. Dumitrescu, Gh. Păun and A. Salomaa, Pattern languages *versus* parallel communicating grammar systems, *Intern. J. Found. Computer Sci.,* to appear.
7. S. Ginsburg, *The Mathematical Theory of Context-free Languages*, McGraw Hill Book Comp., New York, 1996.
8. D. Hauschild and M. Jantzen, Petri nets algorithms in the theory of matrix grammars, *Acta Informatica*, 1994, *31*, pp. 719-728.

9. V. MIHALACHE, Matrix grammars *versus* parallel communicating grammar systems, in vol. *Mathematical Aspects of Natural and Formal Languages* (Gh. PĂUN, ed.), World Sci. Publ., Singapore, 1994, pp. 293-318.
10. V. MIHALACHE, On the generative capacity of parallel communicating grammar systems with regular components, *Computers and AI*, 1996, *15*, pp. 155-172.
11. Gh. PĂUN and L. SÂNTEAN, Parallel communicating grammar systems: the regular case, *Ann. Univ. Buc., Series Matem.-Inform.*, 1989, *38*, pp. 55-63.
12. A. SALOMAA, *Formal Languages*, Academic Press, New York, 1973.