Acta Cybernetica 13 (1997) 173-196.

Various Communications in PC Grammar Systems *

György Vaszil †‡

Abstract

A slightly modified communication protocol called immediate communication is introduced for PC grammar systems and the generative power of these systems is shown to be equal to what we call homogeneous systems, systems with queries of a special form. To acquire this result we also introduce a generalization of returning systems, called systems with returning languages.

1 Introduction

Parallel communicating grammar systems (PC grammar systems) were introduced in [6] as a grammatical description of the so-called classroom model of problem solving. The agents of the classroom are generative grammars, which all operate on their own sentential form, these represent the subsolutions of the overall solution which is the language generated by the whole system. During their operation the agents may communicate, they may exchange their strings with each other. The language generated by the system is the language generated by the classroom leader which is one of the component grammars, usually called the master grammar of the system.

Parallel communicating grammar systems have been the subject of detailed study over the past few years. See [3], [4], [5] for results on their generative power, and [2] on their size parameters. A summary of their properties can be found in the monograph [1].

The power of PC grammar systems is measured by their generative capacity, which may depend on a number of factors. The type of the component grammars and the number of the components are obviously very important among these factors, but there are many others to be considered.

In their paper [6], Gh. Păun and L. Santean considered variants with a universal clock and two basic methods for communication. The presence of the universal clock means that all components use their rules synchronized in time, one derivation step is taken by the system with all components using one of their rewriting rules.

*Presented at the workshop *Grammar Systems: Recent Results and Perspectives, July 26-27, 1996, Budapest.*

†Research supported by the Hungarian Scientific Research Fund, OTKA no. T017105.

‡Computer and Automation Institute Hungarian Academy of Sciences Kende utca 13-17, 1111 Budapest, Hungary, email: vaszil@luna.aszi.sztaki.hu

Communication in this construction is realized with the aid of special nonterminals, the so-called query symbols. Each of these symbols points to one of the component grammars of the system, and when one of them appears in a sentential form, it has to be replaced with the current sentential form of the component it refers to.

This is communication by request, which has two basic variants. One is called returning communication: after a component sends its string to an other component, it must return to its start symbol (or axiom) and begin to generate a new string. The other is called non-returning communication: the component which sends its string keeps a copy for itself and continues to process it after communication.

In the following we keep the basic features of the original model. We will consider synchronized systems with communication by request, but propose a slight change in the communication protocol introducing *immediate communications*, and investigate the impact of this modification on the generative power. To do this, we also generalize the notion of a returning communication by introducing *systems with returning languages.*

The results we obtain will show that the languages generated with immediate communication can be generated with a very much simplified form of query rules using the original protocol. This simple form of queries is what we call *homogeneous*.

2 Preliminaries

The reader is assumed to be familiar with the basics of formal language theory; further details can be found in [7].

The set of all words over an alphabet V and the empty word are denoted by V^* and ϵ respectively, the family of regular, linear and context-free grammars by REG, LIN and CF, respectively. $\mid w \mid$ and $\mid w \mid_X$ denotes the length of a word w and the number of occurences of symbols from set X in w, respectively.

Now we recall the notion of parallel communicating grammar systems from [6], for more material see the monograph [1].

Definition 2.1 A *parallel communicating grammar system* with n components, where $n \geq 1$, (a PC grammar system, for short), is an $(n+3)$-tuple $\Gamma = (N,\ K,\ T,\ G_1, \ldots, G_n)$, where N is a nonterminal alphabet, T is a terminal alphabet and $K = \{Q_1,\ Q_2, \ldots, Q_n\}$ is an alphabet of *query symbols*. N, T, and K are pairwise disjoint sets, $G_i = (N \cup K,\ T,\ P_i,\ S_i)$, $1 \leq i \leq n$, called a *component* of Γ, is a usual Chomsky grammar with nonterminal alphabet $N \cup K$, terminal alphabet T, a set of rewriting rules P_i and an axiom or (a start symbol) S_i. G_1 is said to be the *master* (grammar) of Γ.

Definition 2.2 Let $\Gamma = (N,\ K,\ T, G_1, \ldots, G_n)$, $n \geq 1$, be a PC grammar system as above. An n-tuple $(x_1, \ldots, x_n)$, where $x_i \in (N \cup T \cup K)^*$, $1 \leq i \leq n$, is called a *configuration* of Γ. $(S_1, \ldots, S_n)$ is said to be the *initial configuration.*

PC grammar systems change their configurations by performing direct derivation steps.

Definition 2.3 Let $\Gamma = (N, K, T, G_1, \ldots, G_n)$, $n \geq 1$, be a PC grammar system and let $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$ be two configurations of Γ. We say that $(x_1, \ldots, x_n)$ *directly derives* $(y_1, \ldots, y_n)$, denoted by $(x_1, \ldots, x_n) \Rightarrow (y_1, \ldots, y_n)$, if one of the next two cases hold:

1. There is no x_i which contains any query symbol, that is, $x_i \in (N \cup T)^*$ for $1 \leq i \leq n$. In this case $x_i \Rightarrow_{G_i} y_i$. For $x_i \in T^*$ we have $x_i = y_i$. The system is blocked, if there is an x_j with $| x_j |_N \neq 0$ and none of the rules of P_j can be applied to x_j.

2. There is some x_i, $1 \leq i \leq n$, which contains at least one occurrence of query symbols. Let x_i be of the form $x_i = z_1 Q_{i_1} z_2 Q_{i_2}, \ldots, z_t Q_{i_t} z_{t+1}$, where $z_j \in (N \cup T)^*$, $1 \leq j \leq t+1$ and $Q_{i_l} \in K$, $1 \leq l \leq t$. In this case $y_i = z_1 x_{i_1} z_2 x_{i_2} \ldots z_t x_{i_t} z_{t+1}$, where x_{i_l}, $1 \leq l \leq t$ does not contain any query symbol. In *returning* systems $y_{i_l} = S_{i_l}$, $1 \leq l \leq t$, in *non-returning* systems $y_{i_l} = x_{i_l}$, $1 \leq l \leq t$. If some x_{i_l} contains at least one occurrence of query symbols, then $y_i = x_i$ and also $y_{i_l} = x_{i_l}$, $1 \leq l \leq t$.

If for all x_i with $| x_i |_K \neq 0, x_i = z_1 Q_{i_1} z_2 Q_{i_2}, \ldots, z_t Q_{i_t} z_{t+1}$ there is at least one Q_{i_j}, $1 \leq j \leq t$ that x_{i_j} also contains a query symbol, then the system is blocked due to a circular query.

For all i, $1 \leq i \leq n$, for which y_i is not specified above, $y_i = x_i$.

The first case is the description of a rewriting step: If no query symbols are present in any of the sentential forms, then each component grammar uses one of its rewriting rules except those which have already produced a terminal string. The derivation is blocked if a sentential form is not a terminal string, but no rule can be applied to it.

The second case describes a communication: If some query symbol, say Q_i, appears in a sentential form, then the rewriting stops and a communication step must be performed. The symbol Q_i must be replaced by the current sentential form of component G_i, say x_i, supposing that x_i does not contain any query symbol. If this sentential form also contains some query symbols, then first these symbols must be replaced with the requested sentential forms. If this condition cannot be fulfilled (a circular query appeared), then the derivation is blocked.

Let $\Rightarrow_{rew}$ and $\Rightarrow_{com}$ a denote a rewriting and a communication step respectively.

If the sentential form of a component was communicated to another, this component can continue its own work in two ways: In so-called *returning* systems, the component must return to its axiom and begin to generate a new string. In *non-returning* systems the components do not return to their axiom, but continue to process the current string.

A system is *centralized* if only the component G_1 is allowed to introduce query symbols, otherwise it is *non-centralized.*

By the word *query* we refer to a sentential form containing at least one query symbol. A query is satisfied by a *communication* replacing the query symbols with the requested sentential forms. This may be done in one or more *communication* *steps*. The phrase *communication step* is used to denote the process of satisfying the query symbols, which can be satisfied in "parallel". For example the returning communication prescribed by $(Q_2,\ Q_3,\ \alpha,\ Q_3)$ takes two communication steps to realise: first we get $(Q_2,\ \alpha,\ S_3,\ \alpha)$, and then $(\alpha,\ S_2,\ S_3,\ \alpha)$. The two consecutive steps together will be referred to as a *communication sequence*.

Let $\Rightarrow^+$ and $\Rightarrow^*$ denote the transitive, and the reflexive, transitive closure of $\Rightarrow$ respectively.

Definition 2.4 Let k be a natural number, $k \geq 1$ and let the k step derivations of a PC grammar system be denoted by $(S_1, \ldots, S_n) = (\alpha_1^0, \ldots, \alpha_n^0) \Rightarrow^k (\alpha_1^k, \ldots, \alpha_n^k)$ where $(\alpha_1^k, \ldots, \alpha_n^k)$ is the configuration reached by the system in k steps. The *language* generated by a PC grammar system Γ is

$$L(\Gamma) = \{\alpha_1^k \in T^* \mid (S_1, \ldots, S_n) \Rightarrow^k (\alpha_1^k, \ldots, \alpha_n^k), \alpha_1^j \notin T^*, 1 \leq j < k\}.$$

Thus, the generated language consists of the terminal strings first appearing as sentential forms of the master grammar, G_1.

Let the classes of returning and non-returning PC grammar systems with at most n components of type X, $X \in \{REG,\ LIN,\ CF\}$ and $n \geq 1$ and the corresponding language classes be denoted by PC_nX, NPC_nX and $\mathcal{L}(PC_nX)$, $\mathcal{L}(NPC_nX)$ for non-centralized systems and CPC_nX, $NCPC_nX$, $\mathcal{L}(CPC_nX)$, $\mathcal{L}(NCPC_nX)$ for centralized systems, respectively . When an arbitrary number of components is considered we use $*$ instead of n

3 PC grammar systems with immediate communications

In the communication protocol of [6] the query symbols occurring in one string can only be replaced in one communication step. If it is not possible, the system has to wait until all the query symbols of a sentential form can be replaced. For example the queries (Q_2Q_3, Q_3, a) are satisfied in the returning mode with the following two steps:

$(Q_2Q_3, Q_3, a) \Rightarrow_{com} (Q_2Q_3, a, S_3) \Rightarrow_{com} (aS_3, S_2, S_3)$. Observe that Q_3 of the query Q_2Q_3 did not get replaced in the first step.

In the *immediate communication mode* we allow the replacement of all query symbols that request sentential forms not containing other query symbols. The query above will be satisfied with:

$(Q_2Q_3, Q_3, a) \Rightarrow_{com} (Q_2a, a, S_3) \Rightarrow_{com} (aa, S_2, S_3)$.

Definition 3.1 Let $\Gamma = (N,\ K,\ T, G_1, \ldots, G_n)$, $n \geq 1$, be a usual PC grammar system and let $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$ be two configurations of Γ. We say that $(x_1, \ldots, x_n)$ *directly derives* $(y_1, \ldots, y_n)$, *with immediate communications* if one of the next two cases holds:

1. There is no x_i which contains query symbols, $x_i \in (N \cup T)^*$ for $1 \leq i \leq n$. In this case the system performs a *rewriting step* denoted by $(x_1, \ldots, x_n) \Rightarrow (y_1, \ldots, y_n)$, where $x_i \Rightarrow y_i$ in G_i. For $x_i \in T^*$ we have $x_i = y_i$ and the system is blocked if there is an x_j with $|x_j|_N \neq 0$ and no rule of P_j can be applied to x_j.

2. There is some x_i, $1 \leq i \leq n$, which contains at least one occurrence of query symbols. In this case, the system performs an *immediate communication step* denoted by $(x_1, \ldots, x_n) \Rightarrow (y_1, \ldots, y_n)$, in the following way:

Let x_i be of the form $x_i = z_1 Q_{i_1} z_2 Q_{i_2}, \ldots, z_t Q_{i_t} z_{t+1}$, where $z_j \in (N \cup T)^*$, $1 \leq j \leq t+1$ and $Q_{i_l} \in K$, $1 \leq l \leq t$. Now $y_i = z_1 \delta_{i_1} z_2 \delta_{i_2} \ldots z_t \delta_{i_t} z_{t+1}$, where δ_{i_l}, $1 \leq l \leq t$ is x_{i_l} if x_{i_l} does not contain any query symbol, or δ_{i_l} is Q_{i_l} if x_{i_l} contains at least one query symbol. If $\delta_{i_l} = x_{i_l}$, then in *returning* systems $y_{i_l} = S_{i_l}$, in *non-returning* systems $y_{i_l} = x_{i_l}$, $1 \leq l \leq t$. If $\delta_{i_l} = Q_{i_l}$, then $y_{i_l} = x_{i_l}$, $1 \leq l \leq t$ in both type of systems. The derivation is blocked by a circular query if for all i with $|x_i|_K \neq 0$, $x_i = z_1 Q_{i_1} z_2 Q_{i_2}, \ldots, z_t Q_{i_t} z_{t+1}$ and $y_i = z_1 \delta_{i_1} z_2 \delta_{i_2} \ldots z_t \delta_{i_t} z_{t+1}$, there is a $\delta_{i_l} = Q_{i_l}$, for some l, $1 \leq l \leq t$.

The first case is the description of a usual rewriting step, the second case describes an immediate communication: if more query symbols, say Q_i, Q_j, appear in a sentential form and x_i, the current sentential form of component G_i, does not contain query symbols, then Q_i must be replaced by x_i, even if Q_j can not be replaced by x_j, the current sentential form of G_j in the same step, because it contains further queries. In short, strings without query symbols must be communicated immediately.

Let the class of PC grammar systems of type X with immediate communications and n components of type Y and the corresponding language classes be denoted by fX_nY and $\mathcal{L}(fX_nY)$ respectively, $X \in \{PC, NPC, CPC, CNPC\}$, $Y \in \{REG, LIN, CF\}$. If an arbitrary number of components is considered we put $*$ instead of n.

In a communication sequence with immediate communication, the strings requested by other components are always sent to their destination without any delay if they do not contain further queries. Using the usual communication protocol, it is possible that a sentential form is requested by two other components, but sent to only one of them. For example, if x_i is requested by $x_k = Q_i Q_j$ and $x_l = Q_i$, but $x_j = Q_m$ also contains a query symbol, then x_i can not be sent to x_k, until the query symbol of x_j, the other requested sentential form is replaced. This makes it possible in returning systems, that a query symbol is replaced by the axiom of the queried component instead of the string present at the appearance of the query. In the example above the result of the communication sequence is the following: $y_k = S_i x_m$, $y_l = x_i$, while using immediate communication it would be $y_k = x_i x_m$, $y_l = x_i$.

In a number of cases this difference can not influence the results of a communication sequence. For example, nonreturning systems do not return to their axiom during the communication sequence, centralized systems never request strings containing query symbols and regular or linear components have sentential forms containing at most one query symbol. In these cases the generative capacity of immediate communications coincides with the usual communication modes.

Observation 1

1. $\mathcal{L}(fNPC_nX) = \mathcal{L}(NPC_nX)$, $X \in \{REG, LIN, CF\}$
2. $\mathcal{L}(fPC_nX) = \mathcal{L}(PC_nX)$, $X \in \{REG, LIN\}$
3. $\mathcal{L}(fCPC_nX) = \mathcal{L}(CPC_nX)$, $X \in \{REG, LIN, CF\}$

In the next section we will investigate the generative power of the remaining case, the case of *non-centralized context-free returning* systems.

4 The power of returning systems with immediate communications

In this section we study the generative capacity of *context-free non-centralized returning systems with immeditae communications*, but first we introduce the notion of PC grammar systems with returning languages which will be of help in our investigations.

A PC grammar system with returning languages is a natural extension of a returning system. Each component has an associated language, the so-called returning language. After communication they are allowed to start a new derivation with any word of this language instead of starting with their axiom again.

Definition 4.1 A *PC grammar system with returning languages* is a $(2n+3)$-tuple $\Gamma = (N, K, T, R_1, \ldots, R_n, G_1, \ldots, G_n)$, where N, K, T and $G_1, \ldots, G_n$ are the same as usual, and $R_1, \ldots, R_n$ are non-empty sets of words over $(N \cup T)$, the so-called *returning languages*. ($R_i \subseteq (N \cup T)^*, R_i \neq \{\epsilon\}, R_i \neq \emptyset, 1 \leq i \leq n$).

The system works like a usual returning system, but after communication components may start a new derivation with any word of their returning language.

Let the class of context-free PC grammar systems with returning languages of n components of type X, $X \in \{PC, fPC\}$ and the corresponding language classes be denoted by rX_nCF and $\mathcal{L}(rX_nCF)$, respectively.

With the aid of systems with returning languages we will be able to prove our theorem about the power of immediately communicating systems, which will turn out to be the same as that of usual returning systems with a certain form of queries, which we will call homogeneous queries.

Definition 4.2 Let us call a query *homogeneous*, if all query symbols contained in the corresponding sentential form request the same string, that is, the sentential form is of type $\alpha_1 Q_i \alpha_2 Q_i \ldots \alpha_{t-1} Q_i \alpha_t$, where $1 \leq i \leq n$, $2 \leq t$ and $\alpha_j \in (N \cup T)^*$, $1 \leq j \leq t$.

A *component with homogeneous queries* is a component grammar G_i, $1 \leq i \leq n$, which is allowed to introduce only homogeneous queries, it has no rule of the form $X \to \alpha Q_i \beta Q_j \gamma$, with $i \neq j$, $\alpha, \beta, \gamma \in (N \cup T \cup K)^*$.

A PC grammar system is called *homogeneous*, if it has components with homogeneous queries only.

Let the class of homogeneous PC grammar systems of type X with n context-free components and the corresponding language classes be denoted by hX_nCF and $\mathcal{L}(hX_nCF)$ respectively, where $X \in \{PC, NPC\}$.

The following inclusion is obvious because communication sequences with homogeneous queries produce the same result in the usual and in the immediate communication modes.

Observation 2 $\mathcal{L}(hPC_nCF) \subseteq \mathcal{L}(fPC_nCF)$

Our aim is to prove also the converse inclusion. First we present a lemma about systems with returning languages.

Lemma 4.1 *Let Γ be a returning PC grammar system with immediate communications, having n context-free components and finite returning languages R_i consisting of only nonterminal symbols, $R_i \subseteq N$, $1 \leq i \leq n$.*

If these conditions hold, then there exists Γ', a returning system with immediate communications and $4n$ components which generates the same language as Γ.

Proof: Let $\Gamma = (N,\ K,\ T,\ G_1, R_1, \ldots, G_n, R_n) \in rfPC_nCF$ with nonterminal alphabet N, set of query symbols K, terminal alphabet T, n context-free components $G_1, \ldots, G_n$ and returning languages $R_1, \ldots, R_n$, $R_i \subseteq N$, $1 \leq i \leq n$. Now let $\Gamma' \in fPC_{4n}CF$ be the following:

$$\Gamma' = (N', K', T, G_1^1, .., G_n^1, G_1^2, .., G_n^2, G_1^a, .., G_n^a, G_1^t, .., G_n^t)$$

where

$$\begin{aligned}
N' &= \{S_i^1,\ S_i^2,\ S_i^a, S^{a\prime}, S_i^t, S_i^{t\prime}, S_i^{t\prime\prime} \mid 1 \leq i \leq n\} \cup \\
&\quad\ \{X,\ [X] \mid X \in N\}, \\
P_i^1 &= \{S_i^1 \to Q_i^2,\ S_i^1 \to Q_i^a\} \cup \{X \to [X] \mid X \to \alpha \in P_i\}, \\
P_i^2 &= \{S_i^2 \to Q_i^1\} \cup \{[X] \to \alpha \mid X \to \alpha \in P_i, \alpha \in (N \cup T)^*\} \cup \\
&\quad\ \{[X] \to \alpha_1 Q_{j_1}^1 \alpha_2 \ldots Q_{j_t}^1 \alpha_{t+1} \mid X \to \alpha_1 Q_{j_1} \alpha_2 \ldots Q_{j_t} \alpha_{t+1} \in P_i, \\
&\quad\ \alpha_l \in (N \cup T)^*, 1 \leq l \leq t+1\}, \\
P_i^a &= \{S_i^a \to [S_i], S_i^a \to S_i^{a\prime}, S_i^{a\prime} \to [Y] \mid Y \in R_i\} \text{ and} \\
P_i^t &= \{S_i^t \to S_i^{t\prime}, S_i^{t\prime} \to S_i^{t\prime\prime}, S_i^{t\prime\prime} \to Q_i^a\} \cup \{[Y] \to [Y]', [Y]' \to Q_i^a \mid Y \in R_i\}
\end{aligned}$$

for $1 \leq i \leq n$.

The system has four n-tuples of component grammars, and the rules $X \to \alpha \in P_i$, $1 \leq i \leq n$ of Γ are broken into two parts $X \to [X]$ and $[X] \to \alpha$. $G_1^1, \ldots, G_n^1$ contain the first parts $X \to [X]$ and $G_1^2, \ldots, G_n^2$ the second parts $[X] \to \alpha$.

They work in the following way: all G_i^1 use the first part of some rules while G_i^2 introduce the queries Q_i^1. Now the sentential forms of G_i^1 replace the query symbols in G_i^2, where the application of the rules is finished using their second parts. Next the sentential forms are communicated to G_i^1 and the process starts all over again. The assistant components G_i^a and G_i^t are used to simulate the return of a component to some symbol of the returning language, $1 \leq i \leq n$.

First we show how the initial derivation step of Γ is simulated by Γ'. We start from a configuration

$$(S_1^1, \ldots, S_n^1,\ S_1^2, \ldots, S_n^2,\ S_1^a, \ldots, S_n^a,\ S_1^t, \ldots, S_n^t)$$

and get

$$(\delta_1^1, \ldots, \delta_n^1,\ Q_1^1, \ldots, Q_n^1,\ \delta_1^a, \ldots, \delta_n^a,\ S_1^{t\prime}, \ldots, S_n^{t\prime}),$$

where δ_i^1 is either Q_i^2 or Q_i^a. If some $\delta_j^1 = Q_j^2$ then the derivation is blocked by a circular query. δ_i^a is either $[S_i]$ or $S_i^{a\prime}$. If some $\delta_i^a = S_i^{a\prime}$ then the derivation is also blocked since δ_i^a is passed to G_i^2 and P_i^2 does not contain rules to rewrite $S_i^{a\prime}$. So we must have

$$(Q_1^a, \ldots, Q_n^a,\ Q_1^1, \ldots, Q_n^1,\ [S_1], \ldots, [S_n],\ S_1^{t\prime}, \ldots, S_n^{t\prime}).$$

After one communication step we get

$$(S_1^1, \ldots, S_n^1,\ [S_1], \ldots, [S_n],\ S_1^a, \ldots, S_n^a,\ S_1^{t\prime}, \ldots, S_n^{t\prime}),$$

and then

$$(\delta_1^1, \ldots, \delta_n^1,\ \alpha_1^2, \ldots, \alpha_n^2,\ \delta_1^a, \ldots, \delta_n^a,\ S_1^{t\prime\prime}, \ldots, S_n^{t\prime\prime}).$$

Here α_i^2 differ only in the indices of the query symbols from the strings produced by G_i of Γ, $1 \leq i \leq n$, through the first rewriting step. More precisely if $(S_1, ..., S_n) \Rightarrow_{rew} (\alpha_1, ..., \alpha_n)$, then $\alpha_i^2 = \alpha_i$ if $|\alpha_i|_K = 0$, $\alpha_i^2 = \alpha_{i_1} Q_{j_1}^1 \alpha_{i_2} ... Q_{j_t}^1 \alpha_{i_t}$ if $\alpha_i = \alpha_{i_1} Q_{j_1} \alpha_{i_2} ... Q_{j_t} \alpha_{i_t}$. The δ_i^1 are either Q_i^a or Q_i^2 and δ_i^a are either $S_i^{a\prime}$ or $[S_i]$, $1 \leq j \leq n$. If $\delta_j^1 = Q_j^a$ for some j, $1 \leq j \leq n$, then the system is going to be blocked after the next rewriting step, when G_j^t introduces Q_j^a, because P_j^t does not contain rules to rewrite $[S_j]$ or $S_j^{a\prime}$. If $\delta_i^1 = Q_i^2$ for all i, $1 \leq i \leq n$, then $\delta_i^a = S_i^{a\prime}$ for all i, $1 \leq i \leq n$, because $[S_i]$ can not be rewritten with the rules of P_i^a. So we must have

$$(Q_1^2, \ldots, Q_n^2,\ \alpha_1^2, \ldots, \alpha_n^2,\ S_1^{a\prime}, \ldots, S_n^{a\prime},\ S_1^{t\prime\prime}, \ldots, S_n^{t\prime\prime}),$$

and then after a number of communication steps we get

$$(\beta_1, \ldots, \beta_n,\ S_1^2, \ldots, S_n^2,\ S_1^{a\prime}, \ldots, S_n^{a\prime},\ S_1^{t\prime\prime}, \ldots, S_n^{t\prime\prime}),$$

where the $\beta_i = \gamma_i$ if $(\gamma_1, \ldots, \gamma_n)$ are the sentential forms of Γ produced by the initial rewriting step and the possibly following communication sequence and $\gamma_i \notin R_i$. If $\gamma_j \in R_j$ for some j, $1 \leq j \leq n$, then $\beta_j = S_j^1$. If no α_i^2 contains query symbols (there are no communication steps following the initial rewriting step in Γ), then $\beta_i = \alpha_i^2$. If β_1 is terminal then the system can stop here, if it is not, then the simulation must go on. We start with

$$(\alpha_1, \dots, \alpha_n,\ S_1^2, \dots, S_n^2,\ S_1^{a\prime}, \dots, S_n^{a\prime},\ S_1^{t\prime\prime}, \dots, S_n^{t\prime\prime}),$$

where $\alpha_i \in (N \cup T)^* \cup \{S_i^1\}$ and get

$$(\delta_1^1, \dots, \delta_n^1,\ Q_1^1 \dots, Q_n^1,\ [Y_1], \dots, [Y_n],\ Q_1^a, \dots, Q_n^a),$$

with $\delta_i^1 = [\alpha_i]$ where $[\alpha_i]$ is α_i with one of its nonterminals X rewritten to $[X]$ or if $\alpha_i \in T^*$ then $[\alpha_i] = \alpha_i$. If some communication occured in the previous step and the j-th sentential form was sent to an other component, then $\alpha_j = S_j^1$ and δ_j^1 is either Q_j^a or Q_j^2. If $\delta_j^1 = Q_j^2$ for some j, $1 \leq j \leq n$, then the system is blocked by a circular query, so if $\alpha_j = S_j^1$ for some j, then we must have

$$([\alpha_1^1], .., Q_j^a, .., [\alpha_n],\ Q_1^1 \dots, Q_n^1,\ [Y_1], \dots, [Y_n],\ Q_1^a, \dots, Q_n^a),$$

with $Y_i \in R_i$, $1 \leq i \leq n$. After a communication step we get

$$(S_1^1, \dots, S_n^1,\ [\alpha_1], .., [Y_j], .., [\alpha_n],\ S_1^a, \dots, S_n^a,\ [Y_1], \dots, [Y_n]).$$

Now the system continues the derivation as if G_j^1 has have returned to Y_j instead of its start symbol. We get

$$(\delta_1^1, \dots, \delta_n^1,\ \beta_1^2, \dots, \beta_n^2,\ \delta_1^a, \dots, \delta_n^a,\ [Y_1]', \dots, [Y_n]'),$$

where δ_i^1 and δ_i^a are the same as above and β_i^2 differ only in the indices of the query symbols from the strings produced by G_i of Γ, $1 \leq i \leq n$, as described above. The δ_i^1 are either Q_i^a or Q_i^2 and δ_i^a are either $S_i^{a\prime}$ or $[S_i]$, $1 \leq j \leq n$. If $\delta_j^1 = Q_j^a$ for some j, $1 \leq j \leq n$, then previously described situation arises, the system is going to be blocked after the next rewriting step, when G_j^t introduces Q_j^a, because P_j^t does not contain rules to rewrite $[S_j]$ or $S_j^{a\prime}$. If $\delta_i^1 = Q_i^2$ for all i, $1 \leq i \leq n$, then $\delta_i^a = S_i^{a\prime}$ for all i, $1 \leq i \leq n$ again, because $[S_i]$ can not be rewritten with the rules of P_i^a. So we must have

$$(Q_1^2, \dots, Q_n^2,\ \beta_1^2, \dots, \beta_n^2,\ S_1^{a\prime}, \dots, S_n^{a\prime},\ [Y_1]', \dots, [Y_n]').$$

After the communication sequence we get

$$(\gamma_1, \dots, \gamma_n,\ S_1^2, \dots, S_n^2,\ S_1^{a\prime}, \dots, S_n^{a\prime},\ [Y_1]', \dots, [Y_n]'),$$

where γ_i are the results of the communication sequence prescribed by the β_i sentential forms with $\gamma_j = S_j^1$ if the sentential form of G_j^1 has been sent to an other component during the communication sequence. If $\gamma_1 \in T^*$ the system stops or else it can continue to simulate Γ in the same manner. □

Let the systems satisfying the conditions of the lemma and the corresponding language classes be denoted by $\overline{r}X_nCF$ and $\mathcal{L}(\overline{r}X_nCF)$, $X \in \{PC, fPC\}$, respectively. Note that this proof is based on the fact that using immediate communications each component sends its string only once during a communication sequence, in other words the strings a component has returned to after a communication step are never communicated in the same communication sequence. Since homogeneous systems also have this property and since the simulating system constructed according to the previous theorem is homogeneous if we simulate a homogeneous system, we have the following:

Corollary 4.2 $\mathcal{L}(\bar{r}hPC_*CF) = \mathcal{L}(hPC_*CF)$.

Before we proceed, we need some further observations about the nature of derivations in PC grammar systems. In the proof of our main theorem we would like to treat the communication sequences of a derivation as "units" together. This means that we will assume that terminal words of the master appear only as a result of a rewriting step or as a result of a whole communication sequence, so we need to prove that all languages of PC grammar systems can also be generated this way, where the details of communication sequences are "hidden".

Definition 4.3 Let Γ be a PC grammar system. The language generated by Γ with *hidden communications* is

$$L_h(\Gamma) = \{\omega \in T^* \mid (S_1, S_2, \dots, S_n) \Rightarrow^* (\omega, \alpha_2, \dots, \alpha_n)\},$$

where $|\alpha_i|_K = 0$, $2 \leq i \leq n$ or $\alpha_2, \dots, \alpha_n$ contain a circular query. In other words, the generated language consists of terminal strings present as sentential forms of the first component either after a rewriting step which does not introduce queries, or at the end of a communication sequence, or in a final blocking configuration.

Let the class of languages generated with hidden communications by X type PC grammar systems with n context-free components be denoted by $\mathcal{L}_h(X_nCF)$, $X \in \{fPC, \bar{r}fPC\}$.

Lemma 4.3 *If L is a language generated by a context-free PC grammar system $\Gamma \in X_nCF$, $X \in \{PC, fPC\}$, $L = L(\Gamma)$, then L can also be generated by a system with returning languages $\Gamma' \in \bar{r}X_{2n+2}CF$ with hidden communications, $L = L_h(\Gamma')$.*

Proof: Let $\Gamma = (N, K, T, G_1, \dots, G_n)$ with N, K, T and G_i, $1 \leq i \leq n$ as usual and let $\Gamma' = (N', K', T, G_0, R_0, G_1^1, R_1^1 \dots, G_n^1, R_n^1, G_1^2, R_1^2 \dots, G_n^2, R_n^2, G_a, R_a)$, where G_0 is the master grammar and

$$\begin{aligned}
N' &= \{X, [X] \mid X \in N\} \cup \{A_0, S_0, S_0', S_a, S_a', S_a'', S_i^1, S_i^2 \mid 1 \leq i \leq n\},\\
R_0 &= \{A_0\},\\
P_0 &= \{X \to X \mid X \in N\} \cup \{S_0 \to S_0', S_0' \to Q_1^2\} \cup \{A_0 \to Q_1^2\},\\
R_i^1 &= \{A_i^1\},\\
P_i^1 &= \{S_i^1 \to [S_i], A_i^1 \to Q_i^2\} \cup \{X \to [X] \mid X \in N\},\\
R_i^2 &= \{S_i^2\},\\
P_i^2 &= \{S_i^2 \to Q_i^1\} \cup \{[X] \to \alpha \mid X \to \alpha \in P_i, |\alpha|_K = 0\} \cup\\
&\quad \{[X] \to \alpha_1 Q_{i_1}^1 \alpha_2 \dots \alpha_t Q_{i_t}^1 \alpha_{t+1} \mid X \to \alpha_1 Q_{i_1} \alpha_2 \dots \alpha_t Q_{i_t} \alpha_{t+1} \in P_i,\\
&\quad \alpha_j \in (N \cup T)^*, 1 \leq j \leq t+1\},\\
R_a &= \{S_a\},\\
P_a &= \{S_a \to S_a', S_a' \to S_a'', S_a'' \to Q_0 S_a'\},
\end{aligned}$$

for $1 \leq i \leq n$.

This Γ' system starts with the initial configuration

$$(S_0, S_1^1, \ldots, S_n^1, S_1^2, \ldots, S_n^2, S_a).$$

After a rewriting step we get

$$(S_0', [S_1], \ldots, [S_n], Q_1^1, \ldots, Q_n^1, S_a')$$

and after a communication

$$(S_0', A_1^1, \ldots, A_n^1, [S_1], \ldots, [S_n], S_a').$$

Now a rewriting step follows producing

$$(Q_1^2, Q_1^2, \ldots, Q_n^2, \alpha_1', \ldots, \alpha_n', S_a''),$$

where $\alpha_i' = \alpha_i$ if $S_i \rightarrow \alpha_i \in P_i$ and $\mid \alpha_i \mid_K = 0$ or if $\mid \alpha_i \mid_K \neq 0$, $\alpha_i = \alpha_{i_1} Q_{j_1} \alpha_{i_2} \ldots Q_{j_t} \alpha_{i_t}$, then $\alpha_i' = \alpha_{i_1} Q_{j_1}^1 \alpha_{i_2} \ldots Q_{j_t}^1 \alpha_{i_t}$. After the communication we have

$$(\delta_0, \beta_1, \ldots, \beta_n, S_1^2, \ldots, S_n^2, S_a''),$$

where β_i are the results of the communication sequence prescribed by $\alpha_1', \ldots, \alpha_n'$ with $\beta_j = S_j^1$ if the j-th component has returned to its axiom and δ_0 is either β_1 or if $\beta_1 = S_1^1$ then δ_0 is the string which was sent by G_1^1 during the communication before it has returned to its axiom. If δ_0 is terminal Γ' stops here, otherwise its work continues. After a rewriting step we get

$$(\delta_0, [\beta_1], \ldots, [\beta_n], Q_1^1, \ldots, Q_n^1, Q_0 S_a'),$$

where $[\beta_i]$ is β_i with one of its nonterminals X in brackets $[X]$ ($[\beta_i] = [S_i]$ if $\beta_i = S_i^1$) or if it does not contain any nonterminals then $[\beta_i] = \beta_i$ and δ_0 is the same as above. Now we get

$$(A_0, A_1^1, \ldots, A_n^1, [\beta_1], \ldots, [\beta_n], \delta_0 S_a')$$

and then

$$(Q_1^2, Q_1^2, \ldots, Q_n^2, \gamma_1', \ldots, \gamma_n', \delta_0 S_a''),$$

where $\gamma_i' = \gamma_i$ if $\beta_i \Rightarrow_{G_i} \gamma_i$ with one rewriting step and $\mid \gamma_i \mid_K = 0$, or if $\mid \gamma_i \mid_K \neq 0$, $\gamma_i = \gamma_{i_1} Q_{j_1} \gamma_{i_2} \ldots Q_{j_t} \gamma_{i_t}$, then $\gamma_i' = \gamma_{i_1} Q_{j_1}^1 \gamma_{i_2} \ldots Q_{j_t}^1 \gamma_{i_t}$. After the communication sequence we get

$$(\delta_0, \delta_1, \ldots, \delta_n, S_1^2, \ldots, S_n^2, \delta S_a''),$$

where δ_i, $1 \leq i \leq n$ are the results of the communication sequence prescribed by $\gamma_1', \ldots, \gamma_n'$ with $\delta_j = S_j^1$ if the j-th component has returned to its axiom and δ_0 is either δ_1 or if $\delta_1 = S_1^1$ then δ_0 is the string which was sent by G_1^1 during the communication before it has returned to its axiom. If δ_0 is terminal Γ' stops here, otherwise its work continues in the same manner. □

Now we need to define a notion we will use in the proof of the next lemma.

Definition 4.4 Let $\Gamma = (N, K, T, G_1, \ldots, G_n)$ a context-free PC grammar system with $K = \{Q_1, \ldots, Q_n\}$ and let α be a query, $\alpha = \alpha_1 Q_{i_1} \alpha_2 Q_{i_2} \ldots \alpha_t Q_{i_t} \alpha_{t+1}$, $|\alpha_k|_K = 0$, $1 \leq k \leq t+1$, $1 \leq i_j \leq n$, $1 \leq j \leq t$.

We define *the j-th portion* $1 \leq j \leq t+1$ of this query in the following way: If $j \leq t-1$ then the j-th portion is $\alpha_j Q_{i_j}$. Moreover, if $j = t$, then it is $\alpha_t Q_t \alpha_{t+1}$.

Now we are ready to prove the following:

Lemma 4.4 $\mathcal{L}_h(\overline{r}fPC_*CF) \subseteq \mathcal{L}(\overline{r}hPC_*CF)$

Proof: Let $\Gamma = (N,\ K,\ T,\ R_1, \ldots, R_n,\ G_1, \ldots, G_n) \in \overline{r}fPC_nCF$ be a PC grammar system with immediate communications, nonterminal alphabet N, set of query symbols K, terminal alphabet T, returning languages $R_1, \ldots, R_n$ and n context-free components $G_1, \ldots, G_n$.

Now we construct $\Gamma' \in \overline{r}hPC_mCF$, which generates the same language as Γ. Here $m = (t+2)n + 2u + 3$, where t and u are the following: t is the number of possible rule combinations that we can try to apply to the sentential forms of Γ, u is the sum of u^k, $1 \leq k \leq t$, where u^k is the sum of u_i^k, $1 \leq i \leq n$ and u_i^k is the number of query symbol occurrences on the right-hand side of the i-th rule of the k-th rule combination. Formally $t = |P_1| \prod_{i=2}^{n}(|P_i| + 1)$. If we denote the rules of the k-th rule combination with $X_1^k \to \alpha_1^k, \ldots, X_n^k \to \alpha_n^k$, then $u = \sum_{k=1}^{t} u^k$, $u^k = \sum_{i=1}^{n} u_i^k$, $u_i^k = |\alpha_i^k|_K$.

Γ' simulates the application of each rule combination of Γ in a different n-tuple of simulating components with the aid of assistants assigned to each of the simulating n-tuples. First an integer k, $1 \leq k \leq t$ is selected and the application of the k-th rule combination is simulated in the k-th n-tuple and in the k-th set of assistant components in p steps, with rules using only homogeneous queries. The integer p must be twice the number of necessary communication steps, which is at most $p = 2n - 2$. The simulating system contains the following components:

$$\begin{aligned}\Gamma' = (\ & N', K', T, R_1, \ldots\ldots, R_b, \\ & G_1, .., G_n, G_1^1, .., G_n^1, G_1^2, .., G_n^2, \ldots, G_1^t, .., G_n^t, \\ & G_{11}^1, .., G_{1u_1^1}^1, G_{21}^1, .., G_{2u_2^1}^1, \ldots, G_{n1}^t, .., G_{nu_n^t}^t, \\ & G_{a_1}, G_{a_2}, G_1', .., G_n', G_{11}^{1'}, .., G_{nu_n^t}^{t'}, G_b \quad)\end{aligned}$$

where the n-tuples simulating the k-th rule combination are denoted by G_i^k, $1 \leq i \leq n$ with their assistant components $G_{i\,1}^k, \ldots, G_{i\,u_i^k}^k$. G_{a_1} and G_{a_2} are involved in selecting the number of the rule combination to be simulated, $G_1', \ldots, G_n'$ are needed to help in sending back the sentential forms to $G_1, \ldots, G_n$ after the simulation of a rule combination, $G_{11}^{1'}, .., G_{nu_n^t}^{t'}$ are used to force a restart of the components $G_{11}^1, .., G_{nu_n^t}^t$ by querying them when necessary and G_b makes sure the system blocks if it simulates a rule combination which produces a circular query.

Let $C \subseteq \{1, \ldots, t\}$ be the set of those integers which number rule combinations that introduce circular queries and let the start symbol of the component $G_{\alpha\gamma}^{\beta}$ be

$A^{\beta}_{\alpha\gamma}$.

$$
\begin{aligned}
N' &= \{Z,B,F\} \cup \{(l)^j,(l),(S_i)^j \mid 1 \le l \le t, 1 \le j \le p+2, 1 \le i \le n\} \cup \\
& \quad \{S^{k\,\prime}_{ij}, S^{k\,\prime m}_{ij} \mid 1 \le k \le t, 1 \le m \le p+2, 1 \le i \le n, 1 \le j \le u^k_i\} \cup \\
& \quad \{S^{\beta}_{\alpha\gamma}, A^{\beta}_{\alpha\gamma} \mid G^{\beta}_{\alpha\gamma} \textit{ is a component of } \Gamma'\} \cup \{X,[X] \mid X \in N\} \text{ and} \\
R^{\beta}_{\alpha\gamma} &= \{S^{\beta}_{\alpha\gamma}\}, \text{ where } G^{\beta}_{\alpha\gamma} \text{ is a component of } \Gamma', \\
P_i &= \{A_i \to S_i\} \cup \{X \to [X] \mid X \to \alpha \in P_i\} \cup \{S_{a_1} \to [X] \mid X \in R_i\} \cup \\
& \quad \{S_i \to (S_i)^1, (S_i)^j \to (S_i)^{j+1}, (S_i)^{p+1} \to Q'_i \mid 1 \le j \le p\}, \\
P^j_i &= L1^j_i \cup \{S^j_i \to Q_{a_1}, A^j_i \to Q_{a_1}, S_{a_1} \to S_{a_1}, (j) \to Q_i\} \cup \\
& \quad \{(k) \to (k)^1, (k)^l \to (k)^{l+1}, (k)^{p+2} \to Q_{a_1} \mid 1 \le k \le t, k \ne j, \\
& \qquad 1 \le l \le p+1\}
\end{aligned}
$$

for all $1 \le i \le n$, $1 \le j \le t$ and

$$
\begin{aligned}
P^k_{ij} &= L2^k_{ij} \cup \{S^k_{ij} \to Q_{a_1}, A^k_{ij} \to Q_{a_1}, S_{a_1} \to S_{a_1}, (k) \to (k)^1\} \cup \\
& \quad \{(l) \to (l) \mid 1 \le l \le t, l \ne k\}
\end{aligned}
$$

for all $1 \le k \le t$, $1 \le i \le n$, $1 \le j \le u^k_i$,

$$
\begin{aligned}
P_{a_1} &= \{A_{a_1} \to (k), S_{a_1} \to (k), S_{a_1} \to S_{a_1} \mid 1 \le k \le t\}, \\
P_{a_2} &= \{A_{a_2} \to Q_{a_1}, S'_{a_2} \to Q_{a_1}, S_{a_2} \to S'_{a_2}, (k) \to (k) \mid 1 \le k \le t\}, \\
P'_i &= \{X \to X \mid X \in (N \cup \{S_{a_1}\})\} \cup \{(k) \to Q^k_i \mid 1 \le k \le t\} \cup \\
& \quad \{A'_i \to S'_i, S'_i \to S^{1\,\prime}_i, S^{l\,\prime}_i \to S^{l+1\,\prime}_i S^{p\,\prime}_i \to Q_{a_2} \mid 1 \le l \le p-1\},
\end{aligned}
$$

for $1 \le i \le n$,

$$
\begin{aligned}
P^{k\,\prime}_{ij} &= \{A^{k\prime}_{ij} \to S^{k\prime 1}_{ij}, S^{k\prime}_{ij} \to S^{k\prime 1}_{ij}, S^{k\prime m}_{ij} \to S^{k\prime m+1}_{ij}, S^{k\prime p+2}_{ij} \to Q^k_{ij} \\
& \qquad \mid 1 \le m \le p+1\} \cup \\
& \quad \{(l) \to S^{k\prime 1}_{ij}, S_{a_1} \to S^{k\prime 1}_{ij} \mid l \ne k, 1 \le l \le t\}
\end{aligned}
$$

for all $1 \le k \le t$, $1 \le i \le n$, $1 \le j \le u^k_i$ and

$$
\begin{aligned}
P_b &= \{A_b \to S_b, S_b \to S^1_b, S^l_b \to S^{l+1}_b, S^p_b \to Q_{a_2} \mid 1 \le l \le p-1\} \cup \\
& \quad \{(j) \to B, B \to F \mid j \in C\} \cup \\
& \quad \{(j) \to (j)^1, (j)^1 \to S_b \mid j \notin C\}.
\end{aligned}
$$

We construct the sets $L1^k_i$ and $L2^k_{ij}$ $1 \le i \le n$, $1 \le j \le u^k_i$ in the following way: Let us fix a k and observe the n rules of the k-th rule combination.

The right sides of the rules determine the communication sequence that would follow after rewriting with our certain rule combination.

We say that a sentential form *contains a query* at a certain point of the communication sequence if it contains query symbols which are not yet replaced at that point of the communication sequence.

If our k-th rule combination produces a circular query, we modify the rules. We replace those query symbols which participate in the circle with a new nonterminal Z and execute the following algorithm on this modified rule combination. (See the example at the end of this section.)

For each m, $1 \leq m \leq p/2$, we repeat the following steps. (Note that $p/2$ is the maximal number of communication steps in Γ.) If the j-th rule of our rule combination is the empty rule, then $L1_j^k$ is empty since $u_j^k = 0$, no assistant components G_{ji}^k are present, so we do not need to construct $L2_{ji}^k$. During the following algorithm we consider the j-th sentential form only if the j-th rule of the combination is not empty.

1.a. If the i-th sentential form does not contain a query at the beginning of the m-th communication step and it is not communicated in the m-th communication step then we put the rule $[X_i] \to \alpha_i[X_i]^1$ in $L1_i^k$ where $X_i \to \alpha_i$ is the i-th rule of the k-th rule combination if $m = 1$ and the rule $[X_i]^{2m-2} \to [X_i]^{2m-1}$ for all other m.

1.b. If the i-th sentential form does not contain a query and it is communicated in the m-th step, then we put $[X_i] \to \alpha_i$ in L_i^k if $m = 1$ and $[X_i]^{2m-2} \to \epsilon$ for all other m.

2. If the i-th sentential form contains a query which is not yet satisfied at the beginning of the m-th communication step, then we put $[X_i] \to [X_i]^1$ in $L1_i^k$ if $m = 1$ and $[X_i]^{2m-2} \to [X_i]^{2m-1}$ for all other m.

2.a. If the j-th query symbol of this query is replaced in the m-th communication step then we put $(k)^{2m-1} \to \alpha Q_l^k \beta (k)^{2m}$ in $L2_{ij}^k$, where $\alpha Q_l \beta$ is the j-th portion of the right side of the i-th rule of the k-th combination.

2.b. If the j-th query symbol was or will be replaced in a step different from the m-th, then we put $(k)^{2m-1} \to (k)^{2m}$ in $L2_{ij}^k$.

3. There must be queries that are completely satisfied during the m-th communication step. If the i-th sentential form contains a query which is satisfied completely during the m-th communication step, we put $[X_i]^{2m-1} \to Q_{i1}^k$ in $L1_i^k$ and we put $(k)^{2m} \to Q_{i(j+1)}^k$ in $L2_{ij}^k$ for all $1 \leq j \leq u_i^k - 1$ and $(k)^{2m} \to [X_i]^{2m}$ in $L2_{iu_i^k}^k$.

4. For all i we did not deal with in point **3**, we put $[X_i]^{2m-1} \to [X_i]^{2m}$ in $L1_i^k$. If the i-th sentential form contains a query which is not yet satisfied completely during the m-th communication step, we put $(k)^{2m} \to (k)^{2m+1}$ in all $L2_{ij}^k$, $1 \leq j \leq u_i^k$.

After repeating these steps for all $1 \leq m \leq p/2$, finally add $[X_i]^p \to \epsilon$ to $L1_i^k$, $1 \leq i \leq n$.

Now we turn to the proof of our lemma. First we concentrate on the overall architecture of the simulating system and show how it works. We will see how it provides p steps for simulating each rule combination with the rules of the sets $L1_i^k$ and $L2_{ij}^k$, $1 \leq k \leq t$, $1 \leq i \leq n$, $1 \leq j \leq u_i^k$. Γ' starts with the initial configuration

$$(A_1, .., A_n,\ A_1^1, .., A_n^1, ..., A_1^t, .., A_n^t,$$
$$A_{11}^1, .., A_{1u_1^1}^1, ..., A_{n1}^t, .., A_{nu_n^t}^t,$$
$$A_{a_1}, A_{a_2},\ A_1', .., A_n',\ A_{11}^{1'}, ..., A_{nu_n^t}^{t'}, A_b).$$

After one rewriting step we get

$$(S_1, .., S_n,\ Q_{a_1}, .., Q_{a_1}, ..., Q_{a_1}, .., Q_{a_1},$$

$$Q_{a_1}, .., Q_{a_1}, ..., Q_{a_1}, .., Q_{a_1},$$
$$(k), Q_{a_1},\ S'_1, .., S'_n,\ S_{11}^{1'^1}, ..., S_{nu_n^t}^{t'^1}, S_b),$$

where the component G_{a_1} introduced the nonterminal (k) $1 \leq k \leq n$. This selection of k means that the system will try to apply the k-th rule combination. Now a communication follows

$$(S_1, .., S_n,\ (k), .., (k), ..., (k), .., (k),$$
$$(k), .., (k), ..., (k), .., (k),$$
$$S_{a_1}, (k),\ S'_1, .., S'_n,\ S_{11}^{1'^1}, ..., S_{nu_n^t}^{t'^1}, S_b),$$

where k, $1 \leq k \leq t$ is the number of the rule combination to be applied to the start symbols. Next we get

$$(\delta_1, .., \delta_n,\ (k)^1, .., (k)^1\ , ...,\ Q_1, .., Q_n\ , ...,\ (k)^1, .., (k)^1,$$
$$(k), .., (k)\ , ...,\ (k)^1, .., (k)^1\ , ...,\ (k), .., (k),$$
$$\delta_{a_1}, (k),\ S_1^{1'}, .., S_n^{1'},\ S_{11}^{1'^2}, ..., S_{nu_n^t}^{t'^2}, S_b^1),$$

where δ_i is either $(S_i)^1$ or $[S_i]$, $1 \leq i \leq n$ and δ_{a_1} is either S_{a_1} or (l), $1 \leq l \leq t$. If δ_i is $(S_i)^1$ or δ_{a_1} is (l) then the system will get blocked, since G_i^k do not have rules with $(S_i)^1$ and G_{a_1} does not have rules with (l) on the left side. So we must have

$$([S_1], .., [S_n],\ (k)^1, .., (k)^1\ , ...,\ Q_1, .., Q_n\ , ...,\ (k)^1, .., (k)^1,$$
$$(k), .., (k)\ , ...,\ (k)^1, .., (k)^1\ , ...,\ (k), .., (k),$$
$$S_{a_1}, (k),\ S_1^{1'}, .., S_n^{1'},\ S_{11}^{1'^2}, ..., S_{nu_n^t}^{t'^2}, S_b^1).$$

The assistant grammars $G_1^k, ..., G_n^k$ for the k-th rule combination introduced $Q_1, ..., Q_n$, they will receive the sentential forms of G_i, $1 \leq i \leq n$ and G_{a_2} preserves the value of k for later use. After the communication we have

$$(S_1, .., S_n,\ (k)^1, .., (k)^1\ , ...,\ [S_1], .., [S_n]\ , ...,\ (k)^1, .., (k)^1,$$
$$(k), .., (k)\ , ...,\ (k)^1, .., (k)^1\ , ...,\ (k), .., (k),$$
$$S_{a_1}, (k),\ S_1^{1'}, .., S_n^{1'},\ S_{11}^{1'^2}, ..., S_{nu_n^t}^{t'^2}, S_b^1).$$

If the k-th rule combination is not applicable to the start symbols, then the rules of P_i^k are not applicable to $[S_i]$, $1 \leq i \leq n$. In this case the system is blocked, so let us assume that the k-th rule combination is applicable.

In the next rewriting step the system starts to simulate the effect of the k-th rule combination in p rewriting steps. We are going to show that if the k-th rule combination is applicable to the current sentential forms, then the system provides time for the simulation, takes the resulting sentential forms back to the first n-tuple and starts the process all over again with an other rule combination. The details of the simulation of the rule combinations will be discussed later, for now we denote the sentential forms of the active simulating components G_i^k and their assistants G_{im}^k, $1 \leq m \leq u_i^k$ by α_i^j and β_l^j, $1 \leq i \leq n$, $1 \leq l \leq u^k$, $1 \leq j \leq p$.

We are only interested in the effect the active simulating components and their assistants can have on the rest of the system and this is the following: After communication they return to their axioms and then introduce the query symbol Q_{a_1} querying the "outside world", the component G_{a_1}.

If they receive S_{a_1} then they use the rule $S_{a_1} \to S_{a_1}$ and at the end of the p steps this nonterminal will be sent back to G_i, $1 \leq i \leq n$ with the other simulation result, where it behaves exactly as the original start symbol. We show that the system is blocked if they receive an other symbol. After one rewriting step we get

$$\begin{array}{l}(\delta_1,..,\delta_n,\ (k)^2,..,(k)^2\ ,...,\ \alpha_1^1,..,\alpha_n^1\ ,...,\ (k)^2,..,(k)^2,\\ (k),..,(k)\ ,...,\ \beta_1^1,..,\beta_{u^k}^1\ ,...,\ (k),..,(k),\\ \delta_{a_1},(k),\ S_1^{2'},..,S_n^{2'},\ S_{11}^{1'^3},...,S_{nu_n^t}^{t'^3},S_b^2),\end{array}$$

where δ_i is either $[S_i]$ or $(S_i)^1$, $1 \leq i \leq n$ and δ_{a_1} is either S_{a_1} or (l), $1 \leq l \leq t$. If δ_i is $[S_i]$ or δ_{a_1} is (l), then the system is blocked since P_i and P_{a_1} does not contain rules with $[S_i]$ or (l) on the left side, respectively and no other component (not even the active simulating components G_i^k and their assistants G_{ij}^k, $1 \leq i \leq n$, $1 \leq j \leq u_i^k$) could introduce queries requesting one of these δ_i or δ_{a_1} sentential forms. So we continue from

$$\begin{array}{l}((S_1)^1,..,(S_n)^1,\ (k)^2,..,(k)^2\ ,...,\ \alpha_1^1,..,\alpha_n^1\ ,...,\ (k)^2,..,(k)^2,\\ (k),..,(k)\ ,...,\ \beta_1^1,..,\beta_{u^k}^1\ ,...,\ (k),..,(k),\\ S_{a_1},(k),\ S_1^{2'},..,S_n^{2'},\ S_{11}^{1'^3},...,S_{nu_n^t}^{t'^3},S_b^2),\end{array}$$

and get

$$\begin{array}{l}((S_1)^2,..,(S_n)^2,\ (k)^3,..,(k)^3\ ,...,\ \alpha_1^2,..,\alpha_n^2\ ,...,\ (k)^3,..,(k)^3,\\ (k),..,(k)\ ,...,\ \beta_1^2,..,\beta_{u^k}^2\ ,...,\ (k),..,(k),\\ \delta_{a_1},(k),\ S_1^{3'},..,S_n^{3'},\ S_{11}^{1'^4},...,S_{nu_n^t}^{t'^4},S_b^3).\end{array}$$

where δ_{a_1} is the same as above. We claim that rewriting steps follow in this manner providing the time for the simulation of the rule combination:

$$\begin{array}{l}((S_1)^2,..,(S_n)^2,\ (k)^3,..,(k)^3\ ,...,\ \alpha_1^2,..,\alpha_n^2\ ,...,\ (k)^3,..,(k)^3,\\ (k),..,(k)\ ,...,\ \beta_1^2,..,\beta_{u^k}^2\ ,...,\ (k),..,(k),\\ \delta_{a_1},(k),\ S_1^{3'},..,S_n^{3'},\ S_{11}^{1'^4},...,S_{nu_n^t}^{t'^4},S_b^3) \Rightarrow ... \Rightarrow\end{array}$$

$$\begin{array}{l}((S_1)^{p-1},..,(S_n)^{p-1},\ (k)^p,..,(k)^p\ ,...,\ \alpha_1^{p-1},..,\alpha_n^{p-1}\ ,...,\ (k)^p,..,(k)^p,\\ (k),..,(k)\ ,...,\ \beta_1^{p-1},..,\beta_{u^k}^{p-1}\ ,...,\ (k),..,(k),\\ \delta_{a_1},(k),\ S_1^{p'},..,S_n^{p'},\ S_{11}^{1'^{p+1}},...,S_{nu_n^t}^{t'^{p+1}},S_b^p).\end{array}$$

To verify our claim we show that the active simulating components and their assistants can not interfere with the work of the other components. To do this we have to observe their rule sets.

If one of the simulating components G_i^k, $1 \leq i \leq n$ returns to its axiom during this series of rewriting steps, it introduces Q_{a_1} and receives δ_{a_1} from G_{a_1}. If δ_{a_1} is S_{a_1} then it uses the rule $S_{a_1} \to S_{a_1}$. If δ_{a_1} is (l), $l \neq k$, then it uses its rules $(l) \to (l)^1$ and $(l)^i \to (l)^{i+1}$, $1 \leq i \leq p+1$. If δ_{a_1} is (k), then it introduces Q_i in the next rewriting step and receive $(S_i)^m$, $1 \leq m \leq p-2$ from G_i. In this case the system is blocked since the simulating components do not have rules with $(S_i)^m$ on the left side.

Now let us look at the assistants of the simulating components G^k_{ij}, $1 \leq i \leq n$, $1 \leq j \leq u^k_i$. If one of them returns to its axiom it also introduces Q_{a_1} and receives δ_{a_1} from G_{a_1}. If δ_{a_1} is S_{a_1} or (l), $l \neq k$ then the same things happen as we explained above. If δ_{a_1} is (k) then it is rewritten to $(k)^1$ and then rules of $L2^k_{ij}$ must be used. These rules can only query the active simulating components or their assistants, so they do not interfere with the rest of the system.

From these considerations we see that the system is either blocked, or it reaches the following configuration:

$$((S_1)^{p-1},..,(S_n)^{p-1},\ (k)^p,..,(k)^p\ ,...,\ \alpha_1^{p-1},..,\alpha_n^{p-1}\ ,...,\ (k)^p,..,(k)^p,$$
$$(k),..,(k)\ ,...,\ \beta_1^{p-1},..,\beta_{u^k}^{p-1}\ ,...,\ (k),..,(k),$$
$$\delta_{a_1},(k),\ S_1^{p\prime},..,S_n^{p\prime},\ S_{11}^{1\prime p+1},...,S_{nu_n^t}^{t\prime p+1},S_b^p),$$

where α_i^{p-1} and β_j^{p-1}, $1 \leq i \leq n, 1 \leq j \leq u^k$ can be sentential forms of components that either returned to their axioms or not. If they did not, then we assume the sentential forms to be correct, if they did, then α_i^{p-1} and β_i^{p-1} can be either S_{a_1}, (l), $(l)^m$ or Q_i, $1 \leq l \leq t$, $1 \leq i \leq n$, $1 \leq m \leq p-1$. If any of them is Q_i then a communication step follows and the system is blocked. In the other cases rewriting is possible, so we get

$$((S_1)^p,..,(S_n)^p,\ (k)^{p+1},..,(k)^{p+1}\ ,...,\ \alpha_1^p,..,\alpha_n^p\ ,...,\ (k)^{p+1},..,(k)^{p+1},$$
$$(k),..,(k)\ ,...,\ \beta_1^p,..,\beta_{u^k}^p\ ,...,\ (k),..,(k),$$
$$\delta_{a_1},(k),\ Q_{a_2},..,Q_{a_2},\ S_{11}^{1\prime p+2},...,S_{nu_n^t}^{t\prime p+2},Q_{a_2}),$$

and then after a communication

$$((S_1)^p,..,(S_n)^p,\ (k)^{p+1},..,(k)^{p+1}\ ,...,\ \alpha_1^p,..,\alpha_n^p\ ,...,\ (k)^{p+1},..,(k)^{p+1},$$
$$(k),..,(k)\ ,...,\ \beta_1^p,..,\beta_{u^k}^p\ ,...,\ (k),..,(k),$$
$$\delta_{a_1},S_{a_2},\ (k),..,(k),\ S_{11}^{1\prime p+2},...,S_{nu_n^t}^{t\prime p+2},(k)),$$

where α_i^p and β_j^p are correct by our assumption (if their component grammars never returned to their axioms), or α_i^p and β_i^p can be either S_{a_1}, (l), $(l)^m$ or Q_i, $1 \leq i \leq n$, $1 \leq m \leq p$. If any of them is Q_i then after the replacement of this symbol the system is blocked. In the other cases rewriting is possible again, so we get

$$((S_1)^{p+1},..,(S_n)^{p+1},\ (k)^{p+2},..,(k)^{p+2}\ ,...,\ \alpha_1^{p+1},..,\alpha_n^{p+1}\ ,...,\ (k)^{p+2},..,(k)^{p+2},$$
$$(k),..,(k)\ ,...,\ \beta_1^{p+1},..,\beta_{u^k}^{p+1}\ ,...,\ (k),..,(k),$$
$$\delta_{a_1},S'_{a_2},\ Q_1^k,..,Q_n^k,\ Q_{11}^1,...,Q_{nu_n^t}^t,\delta_b),$$

and then after communication

$$((S_1)^{p+1},..,(S_n)^{p+1},\ (k)^{p+2},..,(k)^{p+2}\ ,...,\ S_1^k,..,S_n^k\ ,...,\ (k)^{p+2},..,(k)^{p+2},$$
$$S_{11}^1,...,S_{nu_n^t}^t,$$
$$\delta_{a_1},S'_{a_2},\ \alpha_1^{p+1},..,\alpha_n^{p+1},\ (k),...,\beta_1^{p+1},..,\beta_n^{p+1},...,(k),\delta_b),$$

where δ_b is either B if the system should block after the simulation of the k-th rule combination or $(k)^1$ if it should not. Now if any of the α_i^{p+1}, $1 \leq i \leq n$ whose component grammar has returned to its axiom is not S_{a_1} or some β_i^{p+1} was not S_{a_1} before the communication, then the system is blocked. Otherwise we get

$$(Q'_1, .., Q'_n, \; Q_{a_1}, ..., Q_{a_1},$$
$$Q_{a_1}, ..., Q_{a_1},$$
$$\delta_{a_1}, Q_{a_1}, \; \alpha_1^{p+1}, .., \alpha_n^{p+1}, \; S_{11}^{1'1}, ..., S_{nu_n^t}^{t'1}, \delta'_b),$$

and then

$$(\alpha_1^{p+1}, .., \alpha_n^{p+1}, \; \delta_{a_1}, ..., \delta_{a_1},$$
$$\delta_{a_1}, ..., \delta_{a_1},$$
$$S_{a_1}, \delta_{a_1}, \; S'_1, .., S'_n, \; S_{11}^{1'1}, ..., S_{nu_n^t}^{t'1}, \delta'_b),$$

where δ_{a_1} is either S_{a_1} or (l), $1 \leq l \leq t$. If it is S_{a_1} then the system is blocked, since G_{a_2} does not have a rule with S_{a_1} on the left. So we have

$$(\alpha_1^{p+1}, .., \alpha_n^{p+1}, \; (l), ..., (l),$$
$$(l), ..., (l),$$
$$S_{a_1}, (l), \; S'_1, .., S'_n, \; S_{11}^{1'1}, ..., S_{nu_n^t}^{t'1}, \delta'_b),$$

where α_i^{p+1}, $1 \leq i \leq n$ is the result of the k-th rule combination with S_{a_1} instead of S_i if the i-th component has returned to its axiom after a communication and δ'_b is either F or S_b. If α_1 is terminal the system stops here, if it is not, then it can continue in the same manner with the simulation of the l-th rule combination if δ'_b is not F. δ'_b is F only if the k-th rule combination introduces a circular query in Γ in which case Γ' should be blocked. If $\alpha_j^{p+1} = S_{a_1}$ for some j, then the j-th component should return to an element of R_j. This is simulated by using a rule $S_{a_1} \to [X]$, $X \in R_j$ in the next step.

Now we show how the p step simulation of the rule combinations is done. We have two cases. If the rule combination to be simulated does not introduce a query, then no assistant components are present. At the beginning of the simulation we get

$$(..., [S_1], .., [S_n], ...) \Rightarrow (..., \alpha_1[S_1]^1, .., \alpha_n[S_n]^1, ...),$$

in G_i^k using the rules of L_i^k, $1 \leq i \leq n$, where α_i are the right sides of the rules of the k-th rule combination. Now p rewriting step follows, we get

$$(..., \alpha_1[S_1]^1, .., \alpha_n[S_n]^1, ...) \Rightarrow ... \Rightarrow (..., \alpha_1[S_1]^p, .., \alpha_n[S_n]^p, ...),$$

and in the next step

$$(..., \alpha_1, .., \alpha_n, ...)$$

using the rules $[S_i]^p \to \epsilon$, $1 \leq i \leq n$. Here α_i is the result of the application of the i-th rule of the simulated rule combination, the system deals with it as we previously described.

If the k-th rule combination introduces queries, the situation is more complicated. At the beginning the sentential forms of the simulating n-tuple and the assistants are

$$(..., [S_1], .., [S_n], ..., (k)^1, .., (k)^1, ...).$$

The sentential forms of the components G_i^k and G_{ij}^k after the l-th rewriting step will be denoted by ${\alpha_i}^l$ and ${\alpha_{ij}}^l$, $1 \leq i \leq n$, $1 \leq j \leq u_i^k$, $1 \leq l \leq p+1$.

After the first rewriting step, the sentential forms of the simulating n-tuple and their assistants are the following:

If the i-th sentential form in Γ is communicated in the first step then the sentential form of G_i^k, ${\alpha_i}^1$ is ω_i, the right side of the i-th rule of the rule combination:

$$(..., \alpha_1^1, .., \omega_{i_1}, .., \alpha_n^1, ..., (k)^2, .., (k)^2, ...).$$

If the i-th sentential form in Γ does not contain a query and it is not communicated in the first step then ${\alpha_i}^1$ is $\omega_i[S_i]^1$, ω_i is as above:

$$(..., \alpha_1^1, .., \omega_{i_1}, .., \omega_{i_2}[S_{i_2}]^1, .., \alpha_n^1, ..., (k)^2, .., (k)^2, ...).$$

In these two cases $u_i^k = 0$, so there are no corresponding assistant components.

If the i-th sentential form of Γ contains a query and the j-th query symbol of this query is replaced in the first step, then ${\alpha_i}^1 = [S_i]^1$, and the sentential form of the assistant component corresponding to this query symbol, α_{ij}^1 is $\alpha_1 Q_l^k \alpha_2 (k)^2$, where $\alpha_1 Q_l \alpha_2$ is the j-th portion of righthand side of the i-th rule:

$$(..., \alpha_1^1, .., \omega_{i_1}, .., \omega_{i_2}[S_{i_2}]^1, .., [S_{i_3}]^1, .., \alpha_n^1, ...$$
$$..., (k)^2, .., \alpha_1 Q_l^k \alpha_2 (k)^2, .., (k)^2, ...).$$

If the j-th query symbol of the i-th sentential form is not replaced in the first step, then ${\alpha_{ij}}^1 = (k)^2$:

$$(..., \alpha_1^1, .., \omega_{i_1}, .., \omega_{i_2}[S_{i_2}]^1, .., [S_{i_3}]^1, .., \alpha_n^1, ...$$
$$..., (k)^2; .., \alpha_1 Q_l^k \alpha_2 (k)^2, .., (k)^2, .., (k)^2, ...).$$

Now a communication follows in Γ'. If the l-th sentential form replaces the j-th query symbol of the i-th sentential form in the first communication step of Γ, then in Γ' ${\alpha_l}^1$ becomes S_l^k, ${\alpha_i}^1$ remains $[S_i]^1$ and ${\alpha_{ij}}^1$ becomes $\alpha_1 {\alpha_l}^1 \alpha_2 (k)^2$:

$$(..., \alpha_1^1, .., S_{i_1}, .., \omega_{i_2}[S_{i_2}]^1, .., [S_{i_3}]^1, .., \alpha_n^1, ...$$
$$..., (k)^2, .., \alpha_1 \omega_{i_1} \alpha_2 (k)^2, .., (k)^2, .., (k)^2, ...).$$

Now a rewriting step follows in Γ'. If the i-th sentential form was communicated in the first step then $\alpha_i^2 = Q_{a_1}$.

If the i-th sentential form was not communicated in the first step and it does not contain a query, then $\alpha_i^2 = \omega_i[S_i]^2$.

If the i-th sentential form contains a query but it is not completely satisfied in the first step, then $\alpha_i^2 = [S_i]^2$. If the j-th sentential form of this query was replaced in the first step then ${\alpha_{ij}}^2 = \alpha_1 {\alpha_l}^1 \alpha_2 (k)^3$:

$$(..., \alpha_1^2, ..Q_{a_1}, .., \omega_{i_2}[S_{i_2}]^2, .., [S_{i_3}]^2, .., \alpha_n^2, ...$$
$$..., (k)^3, .., \alpha_1 \omega_{i_1} \alpha_2 (k)^3, .., (k)^3, .., (k)^3, ...).$$

If the i-th sentential form contains a query which is completely satisfied in the first step, then $\alpha_i^2 = {Q_{i1}}^k$ and $\alpha_{ij}^2 = \omega_j Q_{i(j+1)}^k$, $1 \leq j \leq u_i^k - 1$ and $\alpha_{iu_i^k}^2 = \omega_{u_i^k}[S_i]^2$ where ω_l, $1 \leq l \leq u_i^k$ is the satisfied l-th portion of the righthand side of the query of the i-th rule. In this case a communication step follows in which the results of the query are collected and passed back from the assistants to G_i^k:

$$(\ldots, \alpha_1^2, .., Q_{a_1}, .., \omega_{i_2}[S_{i_2}]^2, .., Q_{i_4}^k, .., \alpha_n^2, \ldots$$
$$\ldots, (k)^3, .., \omega_1 Q_{i_4 2}, \omega_2 [S_{i_4}]^2, .., (k)^3, .., (k)^3, \ldots),$$

$$(\ldots, \alpha_1^2, .., \delta_{i_1}, .., \omega_{i_2}[S_{i_2}]^2, .., \omega_1\omega_2[S_{i_4}]^2, .., \alpha_n^2, \ldots$$
$$\ldots, (k)^3, .., S_{i_4 1}^k, S_{i_4 2}^k, .., (k)^3, .., (k)^3, \ldots),$$

where δ_{i_1} is either S_{a_1}, (l) with $l \neq k$ or (k).

Now the simulation of the first communication step of Γ is complete, the system begins to simulate the second one in the same manner. A rewriting step follows. If the i-th sentential form in Γ is communicated in the second step then $[S_i]^2$ is erased from the sentential form of G_i^k. If the i-th sentential form in Γ does not contain a query after the first communication step and it is not communicated in the second step then either $[S_i]^2$ is changed to $[S_i]^3$ or if the i-th component has returned to its axiom after the first communication step of Γ then there are three possibilities. If Q_{a_1} was replaced by S_{a_1}, then it is not changed. If Q_{a_1} was replaced by (l), $l \neq k$, then it is rewritten to $(l)^1$. If Q_{a_1} was replaced by (k), then it is rewritten to Q_i and after this communication no further rewriting will be possible:

$$(\ldots, \alpha_1^3, .., \delta_{i_1}^1, .., \omega_{i_2}, .., [S_{i_3}]^3, .., \omega_1\omega_2[S_{i_4}]^3, .., \alpha_n^3, \ldots\ldots\ldots),$$

where $\delta_{i_1}^1$ is either S_{a_1}, (l) with $l \neq k$ or Q_{i_1}. If $\delta_{i_1}^1 = Q_{i_1}$ then the system is blocked after the communication.

Now let us look at the assistant components. If $u_i^k \neq 0$ (the i-th sentential form contained a query which was completely satisfied in the first step), then G_{ij}^k, $1 \leq j \leq u_i^k$, the assistant components of G_i^k have also returned to their axiom and now have Q_{a_1} as their sentential form. If Q_{a_1} is replaced by S_{a_1} or by (l), $l \neq k$, then it will not be changed later. If Q_{a_1} is replaced by (k), then it will be rewritten to $(k)^1$ and the assistant will begin to repeat what it previously had done. This will not interfere with the rest of the simulation process, since the i-th sentential form was already communicated.

If the i-th sentential form of Γ contains a query and the j-th query symbol of this query is replaced in the second step, then ${\alpha_i}^3 = [S_i]^3$, and the sentential form of the assistant component corresponding to this query symbol, ${\alpha_{ij}}^3$ is $\alpha_1 Q_l^k \alpha_2 (k)^4$, where $\alpha_1 Q_l \alpha_2$ is the j-th portion of righthand side of the i-th rule. If the j-th query symbol of the i-th sentential form is not replaced in the second step, then ${\alpha_{ij}}^3 = (k)^4$

$$(\ldots, \alpha_1^3, .., \delta_{i_1}^1, .., \omega_{i_2}, .., [S_{i_3}]^3, .., \omega_1\omega_2[S_{i_4}]^3, .., \alpha_n^3, \ldots$$
$$\ldots, (k)^4, .., Q_{a_1}, Q_{a_1}, .., \alpha_1 Q_{i_2}^k \alpha_2 (k)^4, .., (k)^4, \ldots).$$

Now a communication follows in Γ'. If the l-th sentential form replaces the j-th query symbol of the i-th sentential form in the second communication step of Γ, then in Γ' ${\alpha_l}^3$ becomes S_l^k, ${\alpha_i}^3$ remains $[S_i]^3$ and ${\alpha_{ij}}^3$ becomes $\alpha_1 {\alpha_l}^3 \alpha_2 (k)^4$.

$$(\ldots, \alpha_1^3, .., \delta_{i_1}^1, .., S_{i_2}^k, .., [S_{i_3}]^3, .., \omega_1\omega_2[S_{i_4}]^3, .., \alpha_n^3, \ldots$$
$$\ldots, (k)^4, .., \delta_{i_4 1}, \delta_{i_4 2}, .., \alpha_1 \omega_{i_2} \alpha_2 (k)^4, .., (k)^4, \ldots).$$

Now a rewriting step follows in Γ'. If the i-th sentential form was communicated in the second step then $\alpha_i^4 = Q_{a_1}$. If the i-th sentential form was not communicated in the second step and it does not contain a query, then $[S_i]^3$ is changed to $[S_i]^4$ in α_i^4.

If the i-th sentential form contains a query but it is not completely satisfied in the second step, then $\alpha_i^4 = [S_i]^4$. If the j-th sentential form of this query was replaced in the second step then ${\alpha_{ij}}^4 = \alpha_1{\alpha_l}^3\alpha_2(k)^5$.

If the i-th sentential form contains a query which is completely satisfied in the first step, then $\alpha_i^4 = {Q_{i1}}^k$ and $\alpha_{ij}^4 = \omega_j Q_{i(j+1)}^k$, $1 \leq j \leq u_i^k - 1$ and ${\alpha_{iu_i^k}}^4 = \omega_{u_i^k}[S_i]^4$ where ω_l, $1 \leq l \leq u_i^k$ is the satisfied l-th portion of the righthand side of the i-th rule:

$$(\ldots, \alpha_1^4, .., \delta_{i_1}^2, .., Q_{a_1}, .., Q_{i_3 1}^k, .., \omega_1\omega_2[S_{i_4}]^4, .., \alpha_n^4, \ldots$$
$$\ldots, (k)^5, .., \delta_{i_4 1}^1, \delta_{i_4 2}^1, .., \alpha_1\omega_{i_2}\alpha_2[S_{i_3}]^4, .., (k)^5, \ldots).$$

In this case a communication step follows in which the results of the query are collected and passed back from the assistants to G_i^k:

$$(\ldots, \alpha_1^4, .., \delta_{i_1}^2, .., \delta_{i_2}, .., \alpha_1\omega_{i_2}\alpha_2[S_{i_3}]^4, .., \omega_1\omega_2[S_{i_4}]^4, ..\alpha_n^4, \ldots$$
$$\ldots, (k)^5, .., \delta_{i_4 1}^1, \delta_{i_4 2}^1, .., S_{i_2 1}^k, .., (k)^5, \ldots).$$

Now the simulation of the second communication step of Γ is complete, the system begins to simulate the third one in the same manner, and so on.

If the simulation of all communication step is complete, then the system uses the rules $[S_i]^m \Rightarrow [S_i]^{m+1}$, $1 \leq i \leq n$, $1 \leq m \leq p-1$, and finally when G_i' get ready to receive the result, it erases $[S_i]^p$, $1 \leq i \leq n$.

It is clear that all our arguments about the simulation of the first rewriting step and the following communication sequence of Γ by Γ' also hold for all other rewriting steps and communication sequences, where all of the sentential forms contain at least one non-terminal.

Now let us consider the case when one or more of the sentential forms $\alpha_1, \ldots, \alpha_n$ of $G_1, \ldots, G_n$ is terminal and the system chose to simulate the application of a rule combination to these sentential forms.

If α_j is terminal for some j and the j-th rule of the chosen combination is empty, the simulation is correct. Now we show that the simulation is also correct, if α_j is terminal and the j-th rule of the chosen combination is not empty, but it is $X_j \to \omega_j$.

If $|\omega_j|_K = 0$, the j-th rule does not introduce queries, then the simulation would consist of rewriting $[X_j]$ to $\omega_j[X_j]^1$, $\omega_j[X_j]^2$ and so on, until the bracketed nonterminal $[X_j]^l$ is finally erased. Using these rules on $\alpha_j \in T^*$ has the same effect as if the chosen combination contained the empty rule instead of $X_j \to \omega_j$.

If $|\omega_j|_K \neq 0$, the j-th rule introduces queries, then the assistant components $G_{j1}^k, .., G_{ju_j^k}^k$ begin to collect the result of the query. The system will get blocked when they are ready to send the result to G_j^k, because G_{j1}^k can not rewrite the bracketed nonterminals $[X_j]^l$, $1 \leq l \leq p$. □

We demonstrate this construction on a simple example.

Example Consider the following PC grammar system $\Gamma \in \overline{r}fPC_4CF$ generating the language $\{aa\}$.

$\Gamma = (N, K, T, R_1, ..., R_4, G_1, ..., G_4)$, $N = \{S_i \mid 1 \leq i \leq 4\}$, $T = \{a, b\}$,

$P_1 = \{S_1 \rightarrow Q_3Q_2\}$, $P_2 = \{S_2 \rightarrow Q_3\}$, $P_3 = \{S_3 \rightarrow a\}$, $P_4 = \{S_4 \rightarrow b\}$.

Since we have only one rule in each rule set, our rule combinations contain the rule of P_1 and we are free to choose the empty rule instead of one or more rules of the other components. This gives us a total number of 8 combinations, of which only that one is applicable which contains the four rules of the four components. Let this one be the 8-th one and let us concentrate only on this combination.

Now $t = 8$, $u = 20$, $u_1^8 = 2$, $u_2^8 = 1$, $u_3^8 = 0$, $u_4^8 = 0$, the simulating system $\Gamma' \in \overline{r}hPC_{83}CF$ contains the following components:

$$\begin{aligned}\Gamma' = (\;& N', K', T, R_1, .., R_b,\\ & G_1, .., G_4,\; G_1^1, .., G_4^1, ..., G_1^8, .., G_4^8,\\ & G_{11}^1, \ldots, G_{21}^7,\; G_{11}^8, G_{12}^8, G_{21}^8,\\ & G_{a_1}, G_{a_2},\; G_1', .., G_n',\; G_{11}^{1'}, ..., G_{21}^{8'},\; G_b \quad).\end{aligned}$$

The longest communication sequence of the original system contains 2 communications steps so the choise of $p = 4$ is appropriate. The rest of the system Γ' is:

$$\begin{aligned}
N' &= \{S_i,\ [S_i] \mid 1 \leq i \leq 4\} \cup \{S_{\alpha\gamma}^{\beta}, A_{\alpha\gamma}^{\beta} \mid G_{\alpha\gamma}^{\beta} \mathit{is\ a\ component\ of}\ \Gamma'\} \cup\\
&\quad \{S_{ij}^{k\,\prime}, S_{ij}^{k\,\prime m} \mid 1 \leq k \leq 8,\ 1 \leq i \leq 4,\ 1 \leq j \leq u_i^k,\ 1 \leq m \leq 6\} \cup\\
&\quad \{(l), (l)^j, (S_i)^j \mid 1 \leq l \leq 8,\ 1 \leq j \leq 6,\ 1 \leq i \leq 4\} \cup\\
&\quad \{Z, B, F\},\\
R_{\alpha\gamma}^{\beta} &= \{S_{\alpha\gamma}^{\beta}\},\ \mathrm{G}_{\alpha\gamma}^{\beta} \text{ is a component of } \Gamma',\\
P_i &= \{A_i \rightarrow S_i\} \cup \{S_i \rightarrow [S_i]\} \cup \{S_{a_1} \rightarrow [S_i]\} \cup\\
&\quad \{S_i \rightarrow S_i^1, S_i^1 \rightarrow S_i^2, S_i^2 \rightarrow S_i^3, S_i^3 \rightarrow S_i^4, S_i^4 \rightarrow S_i^5, S_i^5 \rightarrow Q_i'\},\\
P_i^j &= L1_i^j \cup \{S_i^j \rightarrow Q_{a_1}, A_i^j \rightarrow Q_{a_1}, (j) \rightarrow Q_i, S_{a_1} \rightarrow S_{a_1}\} \cup\\
&\quad \{(k) \rightarrow (k)^1, (k)^1 \rightarrow (k)^2, (k)^2 \rightarrow (k)^3, (k)^3 \rightarrow (k)^4, (k)^4 \rightarrow (k)^5,\\
&\quad (k)^5 \rightarrow (k)^6, (k)^6 \rightarrow Q_{a_1} \mid 1 \leq k \leq 8, k \neq j\},\\
&\text{for all } 1 \leq i \leq 4,\ 1 \leq j \leq 8 \text{ and}\\
P_{ij}^k &= L2_{ij}^k \cup \{S_{ij}^k \rightarrow Q_{a_1}, A_{ij}^k \rightarrow Q_{a_1}, (k) \rightarrow (k)^1, S_{a_1} \rightarrow S_{a_1}\} \cup\\
&\quad \{(l) \rightarrow (l) \mid 1 \leq l \leq 8, l \neq k\},\\
&\text{for all } 1 \leq k \leq 8,\ 1 \leq i \leq 4,\ 1 \leq j \leq u_i^k.\\
P_{a_1} &= \{A_{a_1} \rightarrow (k), S_{a_1} \rightarrow (k), S_{a_1} \rightarrow S_{a_1} \mid 1 \leq k \leq 8\},\\
P_{a_2} &= \{A_{a_2} \rightarrow Q_{a_1}, S_{a_2}' \rightarrow Q_{a_1}, S_{a_2} \rightarrow S_{a_2}', (k) \rightarrow (k) \mid 1 \leq k \leq 8\},\\
P_i' &= \{S_i \rightarrow S_i, S_{a_1} \rightarrow S_{a_1}\} \cup \{A_i' \rightarrow S_i', (k) \rightarrow Q_i^k \mid 1 \leq k \leq 8\} \cup
\end{aligned}$$

$$\{S_i' \to S_i^{1'}, S_i^{1'} \to S_i^{2'}, S_i^{2'} \to S_i^{3'}, S_i^{3'} \to S_i^{4'}, S_i^{4'} \to Q_{a_2}\},$$

for $1 \leq i \leq 4$ and

$$P_{ij}^{k'} = \{A_{ij}^{k'} \to S_{ij}^{k'1}, S_{ij}^{k'} \to S_{ij}^{k'1}, S_{ij}^{k'm} \to S_{ij}^{k'm+1}, S_{ij}^{k'6} \to Q_{ij}^k \mid 1 \leq m \leq 5\} \cup \{(l) \to S_{ij}^{k'1}, S_{a_1} \to S_{ij}^{k'1} \mid l \neq k, 1 \leq l \leq 8\},$$

for all $1 \leq k \leq 8,\ 1 \leq i \leq 4,\ 1 \leq j \leq u_i^k$,

$$P_b = \{A_b \to S_b, S_b \to S_b^1, S_b^1 \to S_b^2, S_b^2 \to S_b^3, S_b^3 \to S_b^4, S_b^4 \to Q_{a_2}\} \cup \{(j) \to (j)^1, (j)^1 \to S_b \mid 1 \leq j \leq 8\}.$$

Now if we construct the sets $L1_i^8$ and $L2_{ij}^8$ according to the algorithm given above, we get the following result:

$$L_1^8 = \{[S_1] \to [S_1]^1, [S_1]^1 \to [S_1]^2, [S_1]^2 \to [S_1]^3, [S_1]^3 \to Q_{11}^8, [S_1]^4 \to \epsilon\},$$
$$L_2^8 = \{[S_2] \to [S_2]^1, [S_2]^1 \to Q_{21}^8, [S_2]^2 \to \epsilon\},$$
$$L_3^8 = \{[S_3] \to a, [S_3]^1 \to [S_3]^2, [S_3]^2 \to [S_3]^3, [S_3]^3 \to [S_3]^4, [S_3]^4 \to \epsilon\},$$
$$L_4^8 = \{[S_4] \to b[S_4]^1, [S_4]^1 \to [S_4]^2, [S_4]^2 \to [S_4]^3, [S_4]^3 \to [S_4]^4, [S_4]^4 \to \epsilon\},$$
$$L2_{11}^8 = \{(8)^1 \to Q_3^8(8)^2, (8)^2 \to (8)^3, (8)^3 \to (8)^4, (8)^4 \to Q_{12}^8\},$$
$$L2_{12}^8 = \{(8)^1 \to (8)^2, (8)^2 \to (8)^3(8)^3 \to Q_2^8(8)^4, (8)^4 \to [S_1]^4\},$$
$$L2_{21}^8 = \{(8)^1 \to Q_3^8(8)^2, (8)^2 \to [S_2]^2\}.$$

□

By corollary 4.2, lemma 4.3 , lemma 4.4 and by observation 2 we have the following theorem:

Theorem 4.5 $\mathcal{L}(fPC_*CF) = \mathcal{L}(hPC_*CF)$

Proof: The inclusion $\mathcal{L}(hPC_*CF) \subseteq \mathcal{L}(fPC_*CF)$ holds by observation 2. To show the converse inclusion, we have $\mathcal{L}(fPC_*CF) \subseteq \mathcal{L}_h(\overline{r}fPC_*CF)$ by lemma 4.3, $\mathcal{L}_h(\overline{r}fPC_*CF) \subseteq \mathcal{L}(\overline{r}hPC_*CF)$ by lemma 4.4 and $\mathcal{L}(\overline{r}hPC_*CF) = \mathcal{L}(hPC_*CF)$ by corollary 4.2. □

5 Conclusion

In this paper we have introduced immediate communication in parallel communicating grammar systems. Since it differs only slightly from previously existing communications, the generative power of these systems do not change in most cases. To study the generative power of non-centralized, returning systems, we generalized the idea of "returning to the axiom after communication" and we have shown that the use of immediate communications in non-centralized returning PC grammar systems results in the same generative power as if we only used homogeneous queries with the usual communication protocol.

References

[1] E. Csuhaj-Varjú, J. Dassow, J. Kelemen, Gh. Păun, *Grammar Systems. A Grammatical Approach to Distribution and Cooperation,* Gordon and Breach, London, 1994.

[2] J. Kari, L. Santean, The impact of the number of cooperating grammars on the generative power, *Theor. Computer Sci.* 98 (1992), 249-263.

[3] V. Mihalache, On parallel communicating grammar systems with context-free components, in *Mathematical Linguistics and Related Topics* (Gh. Păun, ed.), The Publ. House of the Romanian Academy, Bucharest, 1995, 258-270.

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

[5] Gh. Păun, A. Salomaa, S. Vicolov, On the generative capacity of parallel communicating grammar systems, *Intern. J. Computer Math.* 45 (1992), 49-59.

[6] Gh. Păun, L. Santean, Parallel communicating grammar systems: the regular case, *Ann. Univ. Bucharest, Ser. Matem.-Inform.* 38, 2 (1989), 55-63.

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