Acta Cybernetica 12 (1996) 411-425.

On Regular Characterizations of Languages by Grammar Systems*

Lucian ILIE † Arto SALOMAA‡

Abstract

We show that grammar systems with communication by command and with extremely simple rewriting rules are able to generate all recursively enumerable languages. The result settles several open problems in the area of grammar systems.

1 Introduction

The purpose of this paper is to investigate the power of cooperation in rewriting systems. This is done using the abstract model of a grammar system, [3]. We show that grammar systems with the most simple components, all rewriting rules being letter-to-letter, possess the power of generating all recursively enumerable languages. This result and its corollaries settle several open problems in the area of grammar systems. We now describe the contents of the paper in non-technical terms.

A parallel communicating grammar system, as introduced in [12], consists of several grammars which work synchronously, each of them rewriting its own sentential form, the communication being made by *request*: when a component introduces a query symbol (from a special set) for another component, then the latter one sends its current sentential form to the former which rewrites it in place of the query symbol. The language generated by the system is the set of terminal strings generated (using communication or not) by a distinguished component called master. (For results and references see [3].)

Another kind of parallel communicating grammar systems, with communication by *command*, is introduced in [4] with suggestions from the WAVE paradigm for data flow in highly parallel machines ([5], [6], [14]), Boltzmann machine ([7]), the Connection Machine ([8], [15]), and other well-known parallel machines.

*Research supported by the Academy of Finland, Project 11281

†Turku Centre for Computer Science, FIN-20520 Turku, Finland

‡Academy of Finland and Mathematics Department, University of Turku, FIN-20014 Turku, Finland

The communication by command means that when the current sentential form derived in a component coresponds to another component, i.e., belongs to the regular language associated to the respective component or fits the pattern (in the sense of [1], [11]) associated to that component, then the sentential form is sent to the other component. The language generated by the system is also the set of terminal strings generated by a component designed as master. Here we investigate only the case when each component has associated a regular language.

In [4] it is proved that any context-sensitive language can be generated by a grammar system with communication by command with context-free components while in [10] it is shown that the grammar systems with context-sensitive components and the same type of communication can generate only context-sensitive languages. The characterization of the family of context-sensitive languages as the family of languages generated by grammar systems with context-free components and communication by command follows. We shall strengthen this result by showing that the family of context-sensitive languages is exactly the family of languages generated by the grammar systems with *regular* components and communication by command.

We consider also the case when the splitting is allowed in communication, that is, if the current sentential form of a component is a concatenation of strings each belonging to the regular language associated to another component, then the communication can still be performed: each factor of the sentential form can be sent to the respective component, with the restriction that only one factor can be sent to one component.

As already mentioned in [4], this type of communication is natural: following the *logic flow paradigm* proposed in [6] as a basic architecture for parallel symbolic processing, we deal with a symbolic process which develops in a virtually complete graph having processors which are able to handle data, in its nodes. The process starts by injecting data in a node and each node having data can perform a local processing; under well defined conditions, the local data are spread to other nodes by *replication* or by *splitting*.

In this case we prove a characterization of recursively enumerable languages by grammar systems with (non-erasing) regular rules. In fact, the rules have a particularly simple form: a letter (nonterminal) always goes to a letter (terminal or nonterminal).

2 Grammar systems

We shall denote by V^* the set of all finite strings over the alphabet V, the empty string is denoted by λ, and $V^+ = V^* - \{\lambda\}$. The set of regular, context-free, context-sensitive, and recursively enumerable languages will be denoted by REG, CF, CS, and RE, respectively. For further elements of formal language theory we refer to [9] and [13].

A *parallel communicating grammar system with communication by command* of

degree $n \geq 1$ is a construct of the form

$$\Gamma = (N, T, (S_1, P_1, R_1), \ldots, (S_n, P_n, R_n)),$$

where N is the *nonterminal* alphabet, T is the *terminal* alphabet, and (S_i, P_i, R_i), $1 \leq i \leq n$, are the *components* of the system: S_i is the *axiom*, P_i is the (finite) *set of rules*, (note that we do not allow λ-rules, that is rules in which the right-hand member is empty), and $R_i \in REG$ is the *selector* language for the component i.

Such a system works as follows:

– start from the initial configuration $(S_1, S_2, \ldots, S_n)$,

– at each step, the configuration of the system will be described by an n-tuple $(x_1, x_2, \ldots, x_n) \in ((N \cup T)^*)^n$,

– the configuration of the system can be modified either by *rewriting* steps or by *communication* steps,

– rewriting steps are performed componentwise and the derivation must be maximal in each component (that is the component can not rewrite its sentential form any longer),

– communication steps are performed as follows:

(i) communication *without splitting*: when (after maximal derivations) some components $S_{i_1}, S_{i_2}, \ldots, S_{i_k}, 1 \leq i_1 < i_2 < \cdots < i_k \leq n$, have derived the strings $w_1, w_2, \ldots, w_k \in R_i$, for some $1 \leq i \leq n, i \notin \{i_1, i_2, \ldots, i_k\}$ (a component may not communicate with itself) and these are all the components, at that moment, able to communicate their sentential forms to the component i, then the string $w_1 w_2 \ldots w_k$ will *replace* the sentential form of the component i becoming the current sentential form of this component; the components which send their sentential forms will *restart* from the initial symbol,

(ii) communication *with splitting*: similar to the one without splitting, the difference being that if the sentential form of a component is a catenation of strings each of them belonging to the regular set associated to another component, then each factor of the current string can be sent to the respective component with the following restrictions:

1. only one string can be sent to one component,

2. a component cannot send a factor of its current sentential form to itself (also not the entire string),

3. the catenation of the factors of the current string which are sent must be the entire string (nothing is lost).

– if, after a sequence of rewriting/communication steps, the string on the first position in the current configuration is a terminal one, then it belongs to the generated language (so the master is always the first component).

Formally, a *rewriting step* is

$$(x_1, \ldots, x_n) \Longrightarrow (y_1, \ldots, y_n) \text{ iff } x_i \Longrightarrow^* y_i \text{ in } P_i \text{ and there is no } z_i \in (N \cup T)^* \text{ with } y_i \Longrightarrow z_i \text{ in } P_i.$$

In order to define a communication step without splitting, let us denote

$$\delta(x_i, j) = \begin{cases} \lambda, & \text{if } x_i \notin R_j \text{ or } i = j, \\ x_i, & \text{if } x_i \in R_j \text{ and } i \neq j, \end{cases}$$

for $1 \leq i, j \leq n$,

$$\Delta(i) = \delta(x_1, i)\delta(x_2, i) \dots \delta(x_n, i),$$
$$\delta(i) = \delta(x_i, 1)\delta(x_i, 2) \dots \delta(x_i, n),$$

for $1 \leq i \leq n$.

A *communication step without splitting* is:

$$(x_1, \dots, x_n) \vdash (y_1, \dots, y_n) \text{ iff } y_i = \begin{cases} \Delta(i), & \text{if } \Delta(i) \neq \lambda, \\ x_i, & \text{if } \Delta(i) = \lambda \text{ and } \delta(i) = \lambda, \\ S_i, & \text{if } \Delta(i) = \lambda \text{ and } \delta(i) \neq \lambda. \end{cases}$$

Because the splitting will not be used very much, we define it rather informally. A *communication step with splitting* is

$$(x_1, x_2, \dots, x_n) \vdash_S (y_1, y_2, \dots, y_n)$$

if and only if there is a set $\emptyset \neq M \subseteq \{1, 2, \dots, n\}$ (M is the set of indices of those components which send their sentential forms) such that

(i) for any $i \in M$ there is a permutation of n elements $\pi_i \in \mathcal{S}_n$ and a decomposition $x_i = x_{i,\pi_i(1)}x_{i,\pi_i(2)} \dots x_{i,\pi_i(n)}$ such that $x_{i,i} = \lambda$ and, for any $1 \leq k \leq n, k \neq \pi_i^{-1}(i)$, $x_{i,\pi_i(k)} \in R_{\pi_i(k)}$ or $x_{i,\pi_i(k)} = \lambda$

(ii) for any $i \in \{1, 2, \dots, n\} - M$ and any $1 \leq j \leq n, x_{i,j} = \lambda$,

(iii) for any $1 \leq j \leq n$, if $\lambda \notin R_j$, then

$$y_j = \begin{cases} x_{1,j}x_{2,j} \dots x_{n,j}, & \text{if } x_{1,j}x_{2,j} \dots x_{n,j} \neq \lambda \\ x_j, & \text{if (there is no } i \in M \text{ with } x_{i,j} \in R_j) \text{ and } j \notin M, \\ S_j, & \text{if (there is no } i \in M \text{ with } x_{i,j} \in R_j) \text{ and } j \in M. \end{cases}$$

If $\lambda \in R_j$, then, nondeterministically, the component j can receive λ or can work as in the case when $\lambda \notin R_j$.

Note that the communication without splitting is a particular case of the communication with splitting and also that the empty string can be sent.

The generated language is

$$L_c(\Gamma) = \{w \in T^* \mid (S_1, \dots, S_n) \Longrightarrow (x_1^{(1)}, \dots, x_n^{(1)}) \vdash_c (y_1^{(1)}, \dots, y_n^{(1)}) \Longrightarrow (x_1^{(2)}, \dots, x_n^{(2)}) \vdash_c (y_1^{(2)}, \dots, y_n^{(2)}) \Longrightarrow \dots \Longrightarrow (x_1^{(k)}, \dots, x_n^{(k)}), \text{ for some } k \geq 1 \text{ such that } w = x_1^{(k)}\},$$

where, for $c = \lambda$, we identify $L_\lambda(\Gamma)$ with $L(\Gamma)$ and $\vdash_\lambda$ with $\vdash$ and, for $c = S$, we have $L_S(\Gamma)$ and $\vdash_S$.

We denote by $CCPC_nX$ the family of languages $L(\Gamma)$, generated by grammar systems of degree at most $n, n \geq 1$, with components of type $X \in \{REG, CF, CS\}$, working with communication without splitting, and by $SCCPC_nX$ the family of languages $L_S(\Gamma)$, generated by grammar systems of degree at most $n, n \geq 1$, with components of type X, working with communication with splitting. When the number of components is arbitrary, we write $CCPC_\infty X$ and, respectively, $SCCPC_\infty X$.

3 The characterization results

We begin with the following simple observation. Because in the case when the system has only two components no communication by splitting can be done, we have

Lemma 1 *For any family* X, $CCPC_2X = SCCPC_2X$.

Our first theorem shows that, in the case of communication with splitting, any recursively enumerable language can be generated using a system with *four* regular components. Because the languages associated to the components are regular too, we can say that this is a fully regular characterization of recursively enumerable languages. (Note that we do not allow λ-rules and also not the rewriting of the terminal symbols. The sets of nonterminals and terminals are defined at the level of the system.)

Actually, *three* regular components suffice, as seen in Theorem 2 below. From the point of view of exposition, it is convenient to consider first the weaker version. A further reduction to *two* components is not possible because of Lemma 1 and a result in [4] which shows that in the case of communication without splitting, using two components, only regular languges can be produced.

Theorem 1 $SCCPC_4REG = RE$.

Proof. Let L be a recursively enumerable language over the alphabet T. Then, by a slight modification of Theorem 9.9 in [13], there is a context-sensitive language L_1 and two symbols $a_1, a_2 \notin T$, such that:

(i) L_1 consists of words of the form $wa_2a_1^n, n \geq 0, w \in L$, and

(ii) for every $w \in L$, there is a $n \geq 0$ such that $wa_2a_1^n \in L_1$.

The main idea of our proof is: we construct a system (with four regular components) which generates in one component (which is not the master) any string $wa_2a_1^n \in L_1$ and then, by splitting, the string w is communicated to the master and the garbage $a_2a_1^n$ is communicated to another component. (In fact this is the only moment when the splitting is used, the entire derivation, excepting this, being as in a usual grammar system with communication by command.)

So let $G = (N, T \cup \{a_1, a_2\}, S, P)$ be a context-sensitive grammar generating L_1. Suppose that G is in Kuroda normal form, that is, all productions in G are of the form $AB \longrightarrow CD, A \longrightarrow BC$, and $A \longrightarrow a$ where A, B, C, D are nonterminals and a is a terminal symbol. By introducing, whenever needed, productions of the form $A \longrightarrow B$, A, B nonterminals, we may suppose that if a production of the form $AB \longrightarrow CD$ appears in P, then $A \neq B$.

For a reason that will be seen later, we introduce also the production $S \longrightarrow S$. We label all productions in P by natural numbers $r, 1 \leq r \leq card(P)$. (We construct a bijection between P and the set $\{1, 2, \ldots, card(P)\}$, each production being uniquely identified by its associated number.)

Let S_1', S_2', S_3', X, and Y be symbols not in $N \cup T \cup \{a_1, a_2\}$ and let us put

$$N' = \{A' \mid A \in N\} \cup \{X'\}$$

and

$$\begin{aligned} V = \; & \{A_r \mid A \in N, r : AB \longrightarrow CD \in P \text{ or } r : BA \longrightarrow CD \in P\} \cup \\ & \cup \{A_r \mid A \in N, r : A \longrightarrow \alpha \in P\} \cup \\ & \cup \{X_r \mid r : A \longrightarrow BC \in P\} \cup \\ & \cup \{Z_A, W_A \mid A \in N\}. \end{aligned}$$

We consider the system

$$\begin{aligned} \Gamma = \; & (N \cup N' \cup \{S_1', S_2', S_3', X, Y\} \cup V, T \cup \{a_1, a_2\}, \\ & (S_1', P_1, R_1), (S_2', P_2, R_2), (S_3', P_3, R_3), (S_4, P_4, R_4)) \end{aligned}$$

where $S_4 = S$ and

$$\begin{aligned} P_1 &= \emptyset, \\ R_1 &= T^* \cup \{Y\}, \\ \\ P_2 &= \{S_2' \longrightarrow X, S_2' \longrightarrow Y\}, \\ R_2 &= a_2 a_1^*, \\ \\ P_3 &= \{A' \longrightarrow A \mid A \in N \cup \{X\}\} \cup \\ & \cup \{A_r \longrightarrow a \mid r : A \longrightarrow a \in P\} \cup \\ & \cup \{A_r \longrightarrow B \mid r : A \longrightarrow B \in P\} \cup \\ & \cup \{A_r \longrightarrow C, B_r \longrightarrow D \mid r : AB \longrightarrow CD \in P\} \cup \\ & \cup \{X_r \longrightarrow B, A_r \longrightarrow C \mid r : A \longrightarrow BC \in P\} \cup \\ & \cup \{Z_A \longrightarrow A, W_A \longrightarrow X \mid A \in N\}, \\ R_3 &= \{\alpha_1 A_r \alpha_2 \mid \alpha_1, \alpha_2 \in (N' \cup T \cup \{a_1, a_2\})^*, r : A \longrightarrow a \in P\} \cup \\ & \cup \{\alpha_1 A_r \alpha_2 \mid \alpha_1, \alpha_2 \in (N' \cup T \cup \{a_1, a_2\})^*, r : A \longrightarrow B \in P\} \cup \\ & \cup \{\alpha_1 A_r B_r \alpha_2 \mid \alpha_1, \alpha_2 \in (N' \cup T \cup \{a_1, a_2\})^*, r : AB \longrightarrow CD \in P\} \cup \\ & \cup \{\alpha_1 X_r A_r \alpha_2 \mid \alpha_1, \alpha_2 \in (N' \cup T \cup \{a_1, a_2\})^*, r : A \longrightarrow BC \in P\} \cup \\ & \cup \{\alpha_1 Z_A W_A \alpha_2 \mid \alpha_1, \alpha_2 \in (N' \cup T \cup \{a_1, a_2\})^*, A \in N\}, \\ \\ P_4 &= \{A \longrightarrow A' \mid A \in N \cup \{X\}\} \cup \\ & \cup \{A \longrightarrow A_r \mid r : A \longrightarrow a \in P \text{ or } r : A \longrightarrow B \in P\} \cup \\ & \cup \{X \longrightarrow X_r, A \longrightarrow A_r \mid r : A \longrightarrow BC \in P\} \cup \\ & \cup \{A \longrightarrow A_r, B \longrightarrow B_r \mid r : AB \longrightarrow CD \in P\} \cup \\ & \cup \{X \longrightarrow Z_A, A \longrightarrow W_A \mid A \in N\}, \\ R_4 &= (N \cup \{X\} \cup T \cup \{a_1, a_2\})^+. \end{aligned}$$

(Note that $\lambda \notin R_4$, hence the fourth component cannot be restarted by receiving λ in a communication with splitting.)

Let us prove that the construction is correct, that is $L_S(\Gamma) = L$. We shall do this by showing inclusion in both directions.

Claim 1. If $wa_2a_1^n \in L_1$, then the system Γ can reach a configuration which has in the third position the string $wa_2a_1^n$.

Remark. If Claim 1 holds, then $L \subseteq L_S(\Gamma)$. Indeed, for any $w \in L$, there is an $n \geq 0$ such that $wa_2a_1^n \in L_1$. But, by Claim 1, Γ can reach a configuration with $wa_2a_1^n$ as the current sentential form of the third component. In this case, as $w \in T^* \subset R_1$ and $a_2a_1^n \in R_2$, by splitting, w is communicated to the first component and $a_2a_1^n$ to the second one. Consequently, w is a terminal string and it is the current sentential form of the master, hence $w \in L_S(\Gamma)$.

Proof of Claim 1. Let $wa_2a_1^n$ be a string in L_1. It follows that there is a derivation in G generating it. We show that if α and β are two sentential forms of G such that $\alpha \Longrightarrow_G \beta$, then, having α as the current sentential form of the third component of Γ, we can obtain also β as the sentential form of the third component of Γ.

Because the case when $\alpha = S$ requires some additional explanations, we shall investigate it separately. (In fact, in the first case it will be shown that the derivation in Γ can simulate any beginning of a derivation in G, that is, we can obtain any β with $S \Longrightarrow_G \beta$ as the sentential form of the third component of Γ.)

Case 1 $\alpha = S$. Depending on the form of β, we have three cases:

(i) $\beta = a \in T \cup \{a_1, a_2\}$ and $r : S \longrightarrow a \in P$. (As an observation, because $L_1 \subseteq La_2a_1^*$, a cannot be a_1.) We simulate this in Γ by

$$(S_1', S_2', S_3', S) \Longrightarrow_\Gamma (S_1', Y, S_3', S_r) \vdash_\Gamma (Y, S_2', S_r, S) \Longrightarrow_\Gamma (Y, Y, a, S_r).$$

(ii) $\beta = A \in N$ and $r : S \longrightarrow A$. In Γ we have

$$(S_1', S_2', S_3', S) \Longrightarrow_\Gamma (S_1', Y, S_3', S_r) \vdash_\Gamma (Y, S_2', S_r, S) \Longrightarrow_\Gamma (Y, Y, A, S_r).$$

(iii) $\beta = AB, A, B \in N$ and $r : S \longrightarrow AB$. Supposing that $p : S \longrightarrow S \in P$, we perform in Γ

$$(S_1', S_2', S_3', S) \Longrightarrow_\Gamma (S_1', Y, S_3', S_p) \vdash_\Gamma (Y, S_2', S_p, S) \Longrightarrow_\Gamma (Y, X, S, S_p) \vdash_\Gamma$$
$$\vdash_\Gamma (Y, S_2', S_p, XS) \Longrightarrow_\Gamma (Y, Y, S, X_rS_r) \vdash_\Gamma (Y, S_2, X_rS_r, S) \Longrightarrow_\Gamma (Y, Y, AB, S_p).$$

In words, we have added the rule $S \longrightarrow S$ to P in order to be able to perform this type of rule ($S \longrightarrow AB$) with S on the left-hand side. If the rule $S \longrightarrow S$ is not provided, then we are forced to apply in the fourth component another rule instead of $S \longrightarrow S_p$ ($p : S \longrightarrow S$) and, as at this moment we did not yet get an X in the sentential form of the fourth component, after sending the current string of the last component to the third one, only rules of the form $S \longrightarrow a, a \in T$ or $S \longrightarrow A, A \in N$, can be applied. Consequently, we would not be able to apply a rule of the form $S \longrightarrow AB, A, B \in N$, in this case.

Case 2 $\alpha \in (N \cup T \cup \{a_1, a_2\})^+ - \{S\}$. Depending on the form of the applied production, we have four cases here.

(i) $\alpha = \alpha_1 A \alpha_2, A \in N, \beta = \alpha_1 a \alpha_2, a \in T \cup \{a_1, a_2\}$, $r : A \longrightarrow a \in P$. We simulate this in Γ as follows. If the current sentential form of a component is not important at some moment, we shall replace it by $-$.

$$(-, Y, \alpha_1 A \alpha_2, -) \vdash_\Gamma (Y, S_2', -, \alpha_1 A \alpha_2) \Longrightarrow_\Gamma (Y, Y, -, \alpha_1' A_r \alpha_2') \vdash_\Gamma$$
$$\vdash_\Gamma (Y, S_2', \alpha_1' A_r \alpha_2', -) \Longrightarrow_\Gamma (Y, Y, \alpha_1 a \alpha_2, -)$$

(where for $\alpha \in (N \cup \{X\} \cup T \cup \{a_1, a_2\})^*$ we have denoted by α' the string $h(\alpha)$ where $h : (N \cup \{X\} \cup T \cup \{a_1 a_2\})^* \longrightarrow (N' \cup T \cup \{a_1 a_2\})^*$ is the homomorphism defined by $h(A) = A'$, for any $A \in N \cup \{X\}$, $h(a) = a$, for any $a \in T \cup \{a_1, a_2\}$).

(ii) $\alpha = \alpha_1 A \alpha_2, \beta = \alpha_1 B \alpha_2, A, B \in N, r : A \longrightarrow B \in P$. This is handled as Case 2 (i).

(iii) $\alpha = \alpha_1 AB \alpha_2, \beta = \alpha_1 CD \alpha_2, A, B, C, D \in N, r : AB \longrightarrow CD \in P$. This rule is simulated in Γ by

$$(-, Y, \alpha_1 AB \alpha_2, -) \vdash_\Gamma (Y, S_2', -, \alpha_1 AB \alpha_2) \Longrightarrow_\Gamma (Y, Y, -, \alpha_1' A_r B_r \alpha_2') \vdash_\Gamma$$
$$\vdash_\Gamma (Y, S_2', \alpha_1' A_r B_r \alpha_2', -) \Longrightarrow_\Gamma (Y, Y, \alpha_1 CD \alpha_2, -).$$

(iv) $\alpha = \alpha_1 A \alpha_2, \beta = \alpha_1 BC \alpha_2, A, B, C \in N, r : A \longrightarrow BC \in P$. Because the string generated by P_2 (X or Y) is communicated by the second component (to the fourth component or to the first one, respectively) at each communication step, the derivation in the second component is restarted after each communication performed in the system. Therefore, after each communication step, the second component is able to produce a new X, if needed. (It can also produce a Y if an X is not needed.) As $\alpha \neq S$, there exists a sentential form γ of G such that $\gamma \Longrightarrow_G \alpha$ and we can suppose that (*) when the current sentential form of the third component of Γ is α, then the current string in the second component is X. (We can suppose, for instance, that the second component has introduced an X when γ was obtained in the third one. It is essential here that $\alpha \neq S$; we have seen in Case 1 (iii) how the alternative $\alpha = S$ is handled.)

We may also suppose that the string α contains only nonterminal symbols. (We may obviously suppose that, in a derivation in G, we can apply first only productions of the form $A \longrightarrow B$ or $A \longrightarrow BC$ or $AB \longrightarrow CD$, $A, B, C, D \in N$, and, after that, only productions of the form $A \longrightarrow a$, $A \in N, a \in T \cup \{a_1, a_2\}$.) Consequently, we can put $\alpha_1 = A_1 A_2 \ldots A_k, A_1, A_2, \ldots, A_k \in N, k \geq 0$ ($k = 0$ implies $\alpha_1 = \lambda$) and we can write (using (*))

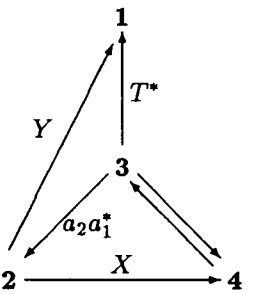
$$\begin{aligned}
&(-, X, \alpha, -) = (-, X, A_1 A_2 \ldots A_k A \alpha_2, -) \vdash_\Gamma (-, S_2', -, X A_1 A_2 \ldots A_k A \alpha_2) \Longrightarrow_\Gamma \\
&\Longrightarrow_\Gamma (-, Y, -, Z_{A_1} W_{A_1} A_2' \ldots A_k' A' \alpha_2') \vdash_\Gamma (Y, S_2', Z_{A_1} W_{A_1} A_2' \ldots A_k' A' \alpha_2', -) \Longrightarrow_\Gamma \\
&\Longrightarrow_\Gamma (Y, Y, A_1 X A_2 \ldots A_k A \alpha_2, -) \vdash_\Gamma (Y, S_2', -, A_1 X A_2 \ldots A_k A \alpha_2) \Longrightarrow_\Gamma \\
&\Longrightarrow_\Gamma \cdots\cdots\cdots\cdots \Longrightarrow_\Gamma \\
&\Longrightarrow_\Gamma (Y, Y, A_1 \ldots A_{k-1} X A_k A \alpha_2, -) \vdash_\Gamma (Y, S_2', -, A_1 \ldots A_{k-1} X A_k A \alpha_2) \Longrightarrow_\Gamma \\
&\Longrightarrow_\Gamma (Y, Y, -, A_1' \ldots A_{k-1}' Z_{A_k} W_{A_k} A' \alpha_2') \vdash_\Gamma (Y, S_2', A_1' \ldots A_{k-1}' Z_{A_k} W_{A_k} A' \alpha_2', -) \Longrightarrow_\Gamma \\
&\Longrightarrow_\Gamma (Y, Y, A_1 \ldots A_{k-1} A_k X A \alpha_2, -) \vdash_\Gamma (Y, S_2', -, A_1 \ldots A_{k-1} A_k X A \alpha_2) \Longrightarrow_\Gamma \\
&\Longrightarrow_\Gamma (Y, Y, -, A_1' \ldots A_k' X_r A_r \alpha_2') \vdash_\Gamma (Y, S_2', A_1' \ldots A_k' X_r A_r \alpha_2', -) \Longrightarrow_\Gamma \\
&\Longrightarrow_\Gamma (Y, Y, A_1 A_2 \ldots A_k B C \alpha_2, -) = (Y, Y, \beta, -).
\end{aligned} \tag{1}$$

Thus Claim 1 is proved.

Claim 2. If $w \in T^*$ was communicated to the master component in Γ (by the third one – this is the only possibility), then, at the moment of communication, the current sentential form of the third component was $wa_2a_1^n \in L_1$ and , by splitting, w was communicated to the master and $a_2a_1^n$ to the second component.

Remark. Obviously, Claim 2 implies $L_S(\Gamma) \subseteq L$.

Proof of Claim 2. Observe that the only possible communications among the components of Γ are represented by the following graph. (An arrow from i to j is present if and only if it is possible that the component i communicates, at some moment, its sentential form to the component j; some arrows are labeled by the regular sets which control the communication.)



We make the following further observations:

1. The second component can communicate to the first one only the string Y which is not terminal. (This communication takes place in order to restart the second component, making it able to produce an X at any moment.)

2. The second component can communicate to the fourth one only the string X.

3. The communication from the third component to the first and the second ones can be done only in the same time by splitting and only when the sentential form of the third component is of the form $wa_2a_1^n$, $n \geq 0$, w being communicated to the master and $a_2a_1^n$ to the second component. (Note that the string communicated to the first component can be empty.)

4. Always, after a maximal derivation in the third component, its current sentential form can be communicated to the fourth component.

5. Due to the form of R_3, if the current sentential form of the fourth component is communicated to the third one (and only to the third one) then a production in P will be correctly applied at the next step in the third component. Indeed, everything should be clear in what concerns the productions of the form $A \longrightarrow a$ or $A \longrightarrow B, A, B \in N, a \in T \cup \{a_1, a_2\}$. A discussion is needed only for the other two types of productions.

(i) For $r : AB \longrightarrow CD \in P; A, B, C, D \in N$. In order to apply this production, in the fourth component one performs $A \longrightarrow A_r$ and $B \longrightarrow B_r$ (providing, of

course, that these productions can be applied). After that, the current sentential form is communicated to the third component if and only if the occurrences of A_r and B_r appear consecutively and in this order (i.e., A_rB_r). In the third component, using the rules $A_r \longrightarrow C$ and $B_r \longrightarrow D$, the string CD is obtained. Because we have supposed that $A \neq B$, there is no danger to apply the production $AB \longrightarrow DC$ instead of $AB \longrightarrow CD$.

(ii) For $r : A \longrightarrow BC \in P; A, B, C \in N$. As it was already seen, for applying a production of this type an occurrence of an X in needed. Without it, the fourth component applies $A \longrightarrow A_r$ but the current sentential form cannot be communicated to the third component because an occurrence of the string A_rX_r is asked by R_3.

Because an occurrence of the symbol X can be communicated by the second component to the fourth one at each communication step (we can apply in P_2 only $S_2' \longrightarrow X$) there is only the danger that too many X's are contained in the sentential form communicated between the last two components. But if the number of X's communicated by the second component to the last one is strictly greater than the number of productions of the form $A \longrightarrow BC$ applied, then the string can be never communicated to the master (no string in R_1 contains X). Hence nothing will be produced in this case.

From the observations above, it should be clear that no parasitic string can be obtained in Γ. Consequently, Claim 2 is proved so we have $L_S(\Gamma) = L$.

□

As said before, the number of the components can be reduced to three.

Theorem 2 $SCCPC_3REG = RE$.

Proof. We use the same notations as in Theorem 1 with the only difference that we consider here one new nonterminal symbol, Z, which is added to the set of nonterminals of the system Γ (with three components)

$$\Gamma = (N \cup N' \cup \{S_1', S_2', X, Y, Z\} \cup V, T \cup \{a_1, a_2\}, (S_1', P_1, R_1), (S_2', P_2, R_2), (S_3, P_3, R_3))$$

where $S_3 = S$ and, supposing that $p : S \longrightarrow S \in P$,

$$\begin{aligned} P_1 &= \{Y \longrightarrow X, Y \longrightarrow Z\}, \\ R_1 &= T^* \cup \{Y\}, \end{aligned}$$

$$\begin{aligned} P_2 = {} & \{S_p \longrightarrow Y\} \cup \\ & \cup\{A' \longrightarrow A \mid A \in N \cup \{X\}\} \cup \\ & \cup\{A_r \longrightarrow a \mid r : A \longrightarrow a \in P\} \cup \\ & \cup\{A_r \longrightarrow B \mid r : A \longrightarrow B \in P\} \cup \\ & \cup\{A_r \longrightarrow C, B_r \longrightarrow D \mid r : AB \longrightarrow CD \in P\} \cup \\ & \cup\{X_r \longrightarrow B, A_r \longrightarrow C \mid r : A \longrightarrow BC \in P\} \cup \\ & \cup\{Z_A \longrightarrow A, W_A \longrightarrow X \mid A \in N\}, \end{aligned}$$

$$
\begin{aligned}
R_2 \;=\; & \{\alpha_1 A_r \alpha_2 \mid \alpha_1, \alpha_2 \in (N' \cup T \cup \{a_1, a_2\})^*, r : A \longrightarrow a \in P\} \cup \\
& \cup \{\alpha_1 A_r \alpha_2 \mid \alpha_1, \alpha_2 \in (N' \cup T \cup \{a_1, a_2\})^*, r : A \longrightarrow B \in P\} \cup \\
& \cup \{\alpha_1 A_r B_r \alpha_2 \mid \alpha_1, \alpha_2 \in (N' \cup T \cup \{a_1, a_2\})^*, r : AB \longrightarrow CD \in P\} \cup \\
& \cup \{\alpha_1 X_r A_r \alpha_2 \mid \alpha_1, \alpha_2 \in (N' \cup T \cup \{a_1, a_2\})^*, r : A \longrightarrow BC \in P\} \cup \\
& \cup \{\alpha_1 Z_A W_A \alpha_2 \mid \alpha_1, \alpha_2 \in (N' \cup T \cup \{a_1, a_2\})^*, A \in N\},
\end{aligned}
$$

$$
\begin{aligned}
P_3 \;=\; & \cup \{A \longrightarrow A' \mid A \in N \cup \{X\}\} \cup \\
& \cup \{A \longrightarrow A_r \mid r : A \longrightarrow a \in P \text{ or } r : A \longrightarrow B \in P\} \cup \\
& \cup \{X \longrightarrow X_r, A \longrightarrow A_r \mid r : A \longrightarrow BC \in P\} \cup \\
& \cup \{A \longrightarrow A_r, B \longrightarrow B_r \mid r : AB \longrightarrow CD \in P\} \cup \\
& \cup \{X \longrightarrow Z_A, A \longrightarrow W_A \mid A \in N\}, \\
R_3 \;=\; & (N \cup \{X\} \cup T \cup \{a_1, a_2\})^+ \cup a_2 a_1^*.
\end{aligned}
$$

The system is working similarly to the one in the proof of Theorem 1. The only differences are the following two:

1. Any string $wa_2a_1^n \in L_1$ is produced here in the second component (instead of the third) and, by splitting, w is sent to the master and $a_2a_1^n$ to the third component (instead of the fourth one). But, because the communication by splitting from the second component to the other two is made only in the case when the sentential form of the second component is of the form $wa_2a_1^n$, w being necessarily sent to the master and $a_2a_1^n$ to the last component, this step is correctly performed.

2. The way in which the occurrences of X are handled in order to help us to use the productions of the form $A \longrightarrow BC, A, B, C \in N$, is slightly different. However, if the number of X's is too big, then no string will be produced (see observation 5 (ii) in the proof of Claim 2 above). We have only to show that indeed we can get sufficiently many X's to be able to apply a rule of the form $A \longrightarrow BC$ anytime it is needed. Supposing that the derivation in G is $\alpha_1 A \alpha_2 \Longrightarrow_G \alpha_1 BC \alpha_2$, we have two cases:

(i) $\alpha_1 = \alpha_2 = \lambda, A = S, r : S \longrightarrow BC \in P; B, C \in N$. We have in Γ (with $p : S \longrightarrow S \in P$)

$$
\begin{aligned}
& (S_1', S_2', S) \Longrightarrow_\Gamma (S_1', S_2', S_p) \vdash_\Gamma (S_1', S_p, S) \Longrightarrow_\Gamma (S_1', Y, S_p) \vdash_\Gamma \\
& \vdash_\Gamma (Y, S_p, S) \Longrightarrow_\Gamma (X, S, S_p) \vdash_\Gamma (S_1', S_p, XS) \Longrightarrow_\Gamma (S_1', S, X_r S_r) \vdash_\Gamma \\
& \vdash_\Gamma (S_1', X_r S_r, S) \Longrightarrow_\Gamma (S_1', BC, S_p).
\end{aligned} \tag{2}
$$

(ii) $\alpha_1 A \alpha_2 \neq S, r : A \longrightarrow BC \in P, A, B, C \in N$. Let us prove first that we can have an X as the current sentential form of the first component anytime needed.

Any simulation in Γ of a derivation in one step, say $\alpha \Longrightarrow_G \beta$, consists of one or several iterations of the following sequence of steps: being the current sentential form of the second component, α is sent to the third one, is rewritten there, is sent back to the second component, and again rewritten. Because $p : S \longrightarrow S \in P$, we have $S_p \longrightarrow S \in P_2$ and $S \longrightarrow S_p \in P_3$. Thus, we can suppose that when

the main string (that is the string which is at the beginning α and, rewritten and communicated between the last two components, will be β) is communicated from component 2 to the component 3 (or from 3 to 2), then the string S_p is communicated from the component 3 to the component 2 (or from 2 to 3, respectively). That can be also seen in (2).

Because we can perform in Γ

$$(-,-,S_p) \vdash_\Gamma (-,S_p,-) \Longrightarrow_\Gamma (-,Y,-) \vdash_\Gamma (Y,-,S) \Longrightarrow_\Gamma (X,-,S_p),$$

using the observations above, it should be clear that we can get an X as the current sentential form of the first component whenever we need one. (It is also seen that the role of the production $S \longrightarrow S$ introduced in P is much more important here.)

Going back to our case, we can suppose (as in the proof of Claim 1, Case 2 (iv)) that when the current sentential form of the second component is $\alpha_1 A \alpha_2$, then the current string in the first component is X. We can also suppose (also as in the proof of Claim 1, Case 2 (iv)) that $\alpha_1 = A_1 A_2 \ldots A_k \in N^*$. The derivation goes now similarly to (1).

Consequently, the system constructed here generates the same language as the one in the proof of Theorem 1. It follows that $L_S(\Gamma) = L$ and the proof is over. □

We notice that in the system Γ in Theorem 2, the splitting communication is used only at the end when the string $w \in L$ is sent to the master and it will be the output of the system and the garbage $a_2 a_1^n$ is sent to the third component. In fact, the splitting communication is done in order to allow a workspace as big as needed.

If the splitting communication is not allowed, we can still obtain (using only regular rules) any context-sensitive language. The following result is a strengthening of Theorem 1 in [4] or of Corollary 3.4 in [10] (which establish that $CCPC_\infty CF = CS$.) It solves also the problem, open so far, of the hierarchy $(CCPC_n REG)_{n \geq 0}$.

Theorem 3 $CCPC_3 REG = CS$.

Proof. The construction is very similar to the one in Theorem 2. The difference is that the second component there is the master one here because we do not need any communication after obtaining the terminal string in the given language.

Let L be a context-sensitive language and let $G = (N, T, S, P)$ be a context-sensitive grammar generating L. We have seen in the proof of Theorem 1 that any production of G can be supposed to be of one of the following forms: $AB \longrightarrow CD$ with $A \neq B$, $A \longrightarrow BC$, $A \longrightarrow B$, or $A \longrightarrow a$ for some $A, B, C, D \in N, a \in T$.

Let S_2', S_3', X, and Y be symbols not in $N \cup T$ and

$$N' = \{A \mid A \in N\} \cup \{X'\},$$

$$\begin{aligned} V = \; & \{A_r \mid A \in N, r : AB \longrightarrow CD \in P \text{ or } r : BA \longrightarrow CD \in P\} \cup \\ & \cup \{A_r \mid A \in N, r : A \longrightarrow \alpha \in P\} \cup \\ & \cup \{X_r \mid r : A \longrightarrow BC \in P\} \cup \\ & \cup \{Z_A, W_A \mid A \in N\}. \end{aligned}$$

The system is here:

$$\Gamma = (N \cup N' \cup \{S_2', S_3', X, Y\} \cup V, T, (S_1, P_1, R_1), (S_2', P_2, R_2), (S_3', P_3, R_3))$$

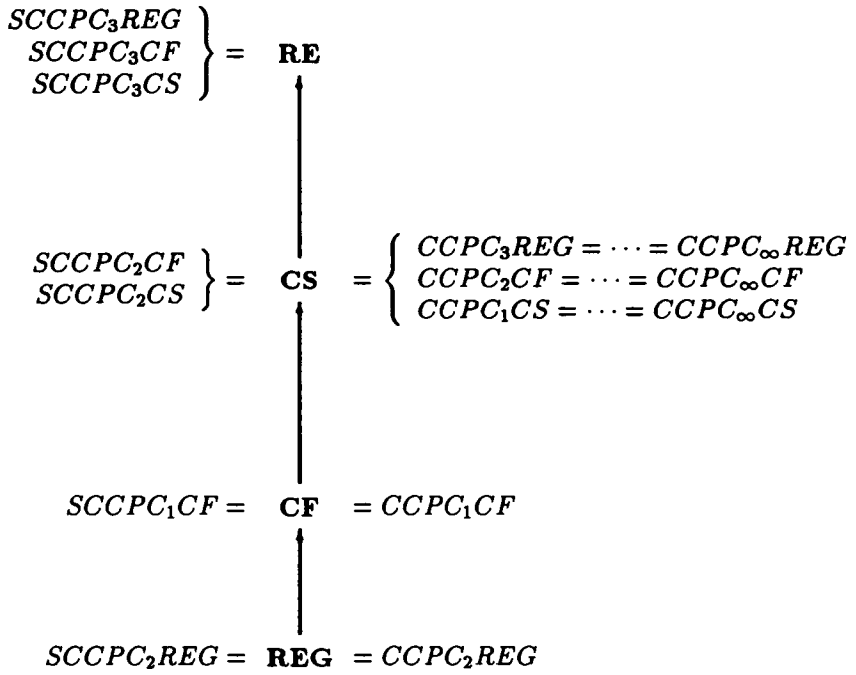
where $S_1 = S$ and

$$\begin{aligned}
P_1 &= \{S_p \longrightarrow Y\} \cup \\
&\quad \cup\{A' \longrightarrow A \mid A \in N \cup \{X\}\} \cup \\
&\quad \cup\{A_r \longrightarrow a \mid r : A \longrightarrow a \in P\} \cup \\
&\quad \cup\{A_r \longrightarrow B \mid r : A \longrightarrow B \in P\} \cup \\
&\quad \cup\{A_r \longrightarrow C, B_r \longrightarrow D \mid r : AB \longrightarrow CD \in P\} \cup \\
&\quad \cup\{X_r \longrightarrow B, A_r \longrightarrow C \mid R : A \longrightarrow BC \in P\} \cup \\
&\quad \cup\{Z_A \longrightarrow A, W_A \longrightarrow X \mid A \in N\}, \\
R_1 &= \{\alpha_1 A_r \alpha_2 \mid \alpha_1, \alpha_2 \in (N' \cup T)^*, r : A \longrightarrow a \in P\} \cup \\
&\quad \cup\{\alpha_1 A_r \alpha_2 \mid \alpha_1, \alpha_2 \in (N' \cup T)^*, r : A \longrightarrow B \in P\} \cup \\
&\quad \cup\{\alpha_1 A_r B_r \alpha_2 \mid \alpha_1, \alpha_2 \in (N' \cup T)^*, r : AB \longrightarrow CD \in P\} \cup \\
&\quad \cup\{\alpha_1 X_r A_r \alpha_2 \mid \alpha_1, \alpha_2 \in (N' \cup T)^*, r : A \longrightarrow BC \in P\} \cup \\
&\quad \cup\{\alpha_1 Z_A W_A \alpha_2 \mid \alpha_1, \alpha_2 \in (N' \cup T)^*, A \in N\}, \\
\\
P_2 &= \{A \longrightarrow A' \mid A \in N \cup \{X\}\} \cup \\
&\quad \cup\{A \longrightarrow A_r \mid r : A \longrightarrow a \in P \text{ or } r : A \longrightarrow B \in P\} \cup \\
&\quad \cup\{X \longrightarrow X_r, A \longrightarrow A_r \mid r : A \longrightarrow BC \in P\} \cup \\
&\quad \cup\{A \longrightarrow A_r, B \longrightarrow B_r \mid r : AB \longrightarrow CD \in P\} \cup \\
&\quad \cup\{X \longrightarrow Z_A, A \longrightarrow W_A \mid A \in N\}, \\
R_2 &= (N \cup \{X\} \cup T)^+. \\
\\
P_3 &= \{Y \longrightarrow X, Y \longrightarrow Z\}, \\
R_3 &= \{Y\},
\end{aligned}$$

The proof for $L(\Gamma) = L$ is very similar to the proof of Theorem 1 and therefore omitted.

□

It is proved in [4] that $CCPC_2REG = REG$ hence, using Lemma 1, we obtain that the results in Theorem 2 and Theorem 3 are optimal. Using also the results $CCPC_\infty CS = CS$ from [10] and $CS \subseteq CCPC_2CF$ from [4], we can draw the following diagram which shows the generative power of all types of systems with communication by command investigated so far by comparing them with the families in Chomsky hierarchy. (The place of the families $SCCPC_nX, CCPC_nX$ not mentioned in the diagram is obvious.)



References

[1] D. Angluin, Finding patterns common to a set of strings, *J. Comput. Syst. Sci.*, **21**(1980), 46 – 62.

[2] K. Culik II, A purely homomorphic characterization of recursively enumerable sets, *Journal of the Association for Computing Machinery*, **26**(1979), 345–350.

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

[4] E. Csuhaj-Varjú, J. Kelemen, Gh. Păun, Grammar systems with WAVE-like communication, *Computers and AI*, **15** (1996), 419-436.

[5] L. Errico, "WAVE: An Overview of the Model and the Language", CSRG, Dept. Electronic and Electr. Eng., Univ. of Surrey, UK, 1993.

[6] L. Errico, C. Jesshope, Towards a new architecture for symbolic processing, *in* "Artificial Intelligence and Information-Control Systems of Robots '94" (I. Plander, ed.), World Sci. Publ., Singapore, 1994, 31 – 40.

[7] S. E. Fahlman, G. E. Hinton, T. J. Seijnowski, Massively parallel architectures for AI: NETL, THISTLE and Boltzmann machines, *in* "Proc. AAAI National Conf. on AI", William Kaufman, Los Altos, 1983, 109 – 113.

[8] W. D. Hillis, "The Connection Machine", MIT Press, Cambridge, 1985.

[9] J. E. Hopcroft, J. D. Ullman, "Introduction to automata theory, languages, and computation", Addison-Wesley, Reading, Mass., 1979.

[10] L. Ilie, Collapsing hierarchies in parallel communicating grammar systems with communication by command, *Computers and AI*, **15, 2-3**(1996), 173 – 184.

[11] T. Jiang, E. Kinber, A Salomaa, K. Salomaa, S. Yu, Pattern languages with and without erasing, *Intern. J. Computer Math.*, **50**(1994), 147 – 163.

[12] Gh. Păun, L. Sântean, Parallel communicating grammar systems: the regular case, *Ann. Univ. Buc., Math.-Informatics Series*, **38, 2**(1989), 55 – 63.

[13] A. Salomaa, "Formal Languages", Academic Press, New York, 1973.

[14] P. S. Sapaty, "The WAVE Paradigm", Internal Report 17/92, Dept. Informatics, Univ. of Karlsruhe, Germany, 1992.

[15] ***, "Connection Machine, Model CM-2. Tehnical Summary", Thinking Machines T. R. HA 87 – 4, MIT, Cambridge, USA, 1987.