Two lower bounds on distributive generation of languages

Juraj Hromkovič[+]
Dept. of Mathematics & Informatics, University of Paderborn
33098 Paderborn, Germany

Jarkko Kari, Lila Kari[**]
Dept. of Mathematics, University of Turku and Academy of Finland
20500 Turku, Finland

Dana Pardubská[+,*]
Faculty of Mathematics and Physics, Comenius University
84215 Bratislava, Slovakia

Abstract

The lower bounds on communication complexity measures of language generation by Parallel Communicating Grammar Systems (PCGS) are investigated. The first result shows that there exists a language that can be generated by some dag-PCGS (PCGS with communication structures realizable by directed acyclic graphs) consisting of 3 grammars, but by no PCGS with tree communication structure.

The second result shows that dag-PCGS have their communication complexity of language generation either constant or linear.

[+] These authors were partially supported by MSV SR grant and by EC Cooperative Action IC 1000: project ALTEC

[**] The work of this author has been supported by Project 11281 of the Academy of Finland.

[*] The part of the work of this author has been done during the author's stay at the Dept. of Mathematics & Informatics of the University of Paderborn.

1 Introduction

This work is devoted to the computational aspects of Parallel Communicating Grammar Systems ($PCGS$) introduced in [PS 89]. The concept of $PCGS$ differs from the previous models of parallel derivation of words (languages) like Lindenmayer systems [HR 75, RS 80] in that a $PCGS$ can be viewed as a typical distributive system consisting of a number of independent elements cooperating by the exchange of information via communication links. Thus, the derivation of a $PCGS$ is a sequence of parallel derivation steps and communication steps. As typical for distributive systems there are several complexity measures of importance for $PCGS$. The most investigated complexity measure for PCGS has been the number of grammars the PCGS consists of, which is clearly a descriptive complexity measure. The hierarchy results claiming that cPCGS (centralized PCGS) of regular $n+1$ grammars are more powerful than cPCGS of n regular grammars for any positive integer n have been established in [SK 92].

Because the complexity of the executed communication shows generally to be the crucial one in parallel systems, Hromkovič et. al.[HKK 93] proposed to consider two communication complexity measures for PCGS. The first one is the communication structure of PCGS (the shape of the graph consisting of the directed communication links between the grammars of the system) which can be considered as alternative descriptive complexity measure. As classes of structures of a principal interest, linear arrays (chains), rings, trees and directed acyclic graphs have been proposed in [HKK 93]. The second communication complexity measure introduced in [San 90, HKK 93] is the number of exchanged messages (strings) during the generation procedure. This complexity measure is clearly a computational complexity measure which may be considered as a function of the length of the generated word.

The aim of this paper is to study the two last mentioned communication complexity measures. First, the power of different communication structures is investigated. In [HKK 93] some special lower bound techniques were developed for special communication structures as rings, trees and dags. These techniques have enabled to show strong hierarchies on the number of grammars in these structures. However, no result showing that any type of communication structures is more powerful than another one has been achieved there. In [Luk 92] the $chain-PCGS$s (PCGSs with communication structure chain) have been shown to be more powerful than $cPCGS$s. In [Par 93], showing that rings and dags can generate only regular languages over one-letter alphabet (independently of the number of component grammars used) Pardubská gets that rings and dags are weaker than arbitrary structures (complete graphs). Since a nonregular language over one-letter alphabet can be achieved by 3 grammars, this shows that the number of grammars cannot compensate for suitable communication structure.

Here, we show, that $dag-PCGSs$ ($PCGS$s with communication structure corresponding to a directed acylic graph) are more powerful than tree-$PCGS$s ($PCGSs$ with tree communication structure). Namely, we prove that $L=\{ww \mid w \in Z^*\}$ can be generated by a dag-$PCGS$ consisting of three grammars, but by no tree-$PCGS$.

Our second result is devoted to the study of the power of communication complexity as computational complexity measure. In [HKK 93] the power of $PCGS$'s with special communication structures were studied from computational complexity point of view.

It was shown there, that there is an infinite hierarchy of constant communication complexity for $cPCGS$ and tree-$PCGS$. ($k+1$ communications are more powerful than k, where k is independent of the length of generated words). Such a hierarchy for unrestricted communication structure and for dags has been established in [Par 93]. Pardubska [Par 93] also showed, for unrestricted communication structures, that

(i) if the communication complexity cannot be bounded by a constant, then it is at least $\Omega(\log_2 n)$, and

(ii) for every $k \in \mathbb{N}, o(\sqrt[k]{n})$ communications are less powerful than $O(\sqrt[k]{n})$.

The last known result about communication complexity as an unbounded function has been established in [HKK 93] where a language requiring linear $\Omega(n)$ communication complexity to be generated by tree-$PCGS$s is constructed. Here, we deal with the problem, whether it is possible to get a result like (ii) for some unbounded function for tree-$PCGS$s and dag-$PCGS$s. The answer is negative, because we prove that every dag-$PCGS$ has either constant communication complexity or $\Omega(n)$ communication complexity.

The paper is organized as follows. The next section contains the basic definitions and fixes the notation used. In Section 3 we show that direct acyclic graphs are more powerful communication structures than trees, and in Section 4 we prove the non-existence of any hierarchy of communication complexity restricted by unbounded functions for dag-$PCGS$s.

2 Preliminaries

We assume the reader to be familiar with basic definitions and notations in formal language theory and we specify only some of them related to the PCGS. We denote by ε the empty symbol (word) and, for any word x, $|x|$ denotes the length of x. For a set K of symbols and a word x, $|x|_K$ denotes the number of occurrences of symbols of K in x. Let R denote the set of regular languages.

First, we give the formal definition of Parallel Communicating Grammar System ($PCGS$).

Definition 2.1 *A PCGS of degree m, $m \geq 1$, is an $(m+1)$-tuple $\Pi = (G_1, \ldots, G_m, K)$, where*

- $G_i = (N_i, T, S_i, P_i)$ *are regular grammars satisfying*
  - $N_i \cap T = \emptyset$ *for all* $i \in \{1, \ldots, m\}$
  - $P_i \subset N \text{ x } T^* N \cup N \text{ x } T^+$
- $K \subseteq \{Q_1, \ldots, Q_m\} \bigcap \cup_{i=1}^{m} N_i$ *is a set of special symbols, called communication symbols,* $K_i = K \cap N_i$ *is the set of communication symbols of* G_i.

Now, we describe the work of $PCGS$s. The possible communications in a PCGS Π are determined by the communication graph. The vertices of this directed graph $G(\Pi)$ correspond to the individual component grammars and are labelled by their names $G_1, \dots, G_m$. The directed edges describe the possibility of inquiry. The edge (G_i, G_j) is present in $G(\Pi)$ iff the communication symbol Q_j belongs to the nonterminal alphabet of G_i.
An m-tuple $(x_1, \dots, x_m)$, $x_i = \alpha_i A_i$, $\alpha_i \in T^*, A_i \in (N_i \cup \varepsilon)$, is called configuration. With every configuration $C = (\alpha_1 A_1, \dots, \alpha_m A_m)$ its nonterminal cut $N(C) = (A_1, A_2, \dots, A_m)$ is associated. If the nonterminal cut of the configuration contains at least one communication symbol, then the so-called communication cut, that is m-tuple $(B_1, B_2, \dots, B_m)$, where $B_i = A_i$ for $A_i \in K$ and $B_i = \varepsilon$ for $A_i \notin K$, is associated with it, too.
We say a configuration $(x_1, \dots, x_m)$ directly derives a configuration $(y_1, \dots, y_m)$ and write $(x_1, \dots, x_m) \rightarrow (y_1, \dots, y_m)$, if one of the next two cases holds:

1. $|x|_K = 0$ for all i, $1 \leq i \leq m$, and either $x_i \rightarrow y_i$ in G_i when x_i contains nonterminal or x_i is the terminal word and $y_i = x_i$.
2. if $|x_i|_K > 0$ for some i, $1 \leq i \leq m$, then, for each i such that $x_i = z_i Q_{j_i}$, for some $z_i \in T^*, \forall Q_{j_i} \in K$, the following happens:
   (a) If $|x_{j_i}|_K = 0$ then $y_i = z_i x_{j_i}$ and $y_{j_i} = S_{j_i}$
   (b) If $|x_{j_i}|_K > 0$ then $y_i = x_i$.

For all the remaining indices t, for which x_t does not contain communication symbols and Q_t has not occurred in any of x_i's, we put $y_t = x_t$.

A derivation of a $PCGS$ Π is a sequence of configurations $X_1, X_2, \dots, X_t$, where $X_i \rightarrow X_{i+1}$ is a direct derivation in Π. It can be viewed as a sequence of *rewriting* and *communication steps*, too. If no communication symbol appears in any of the component grammars then we perform a rewriting step consisting of rewriting steps synchronously performed in each of the grammars. If some of the components is a terminal string, it is left unchanged. If some of the component grammars contains a nonterminal that cannot be rewritten, the derivation is blocked. If the first grammar G_1 contains a terminal word y, the derivation is finished and y is the word generated by Π in this derivation. If a communication symbol is present in any of the components, then a communication step is performed. It consists of replacing all communication symbols with the phrases they refer to, under condition that these phrases do not contain further communication symbols. If some communication symbols are not satisfied in this communication step, they may be satisfied in one of the next ones. Communication steps are performed until no more communication symbols are present or the derivation is blocked because no communication symbol has been satisfied in the last communication step. The language generated by a $PCGS$ consists of the terminal words generated in G_1 (in the cooperation with the other grammars).

Definition 2.2 *For any PCGS* Π,

$$L(\Pi) = \{\alpha \in T^* |\ (S_1, \dots, S_m) \rightarrow^* (\alpha, \beta_2, \dots, \beta_m)\}$$

Some examples of $PCGS$ generating specific languages can be found in [HKK 93, Luk 92, San 90].

Now, for our lower bound results, we need to go in more details concerning the derivations of $PCGS$s. Let $\mathcal{D}(w) = C_1, C_2, \ldots, C_t$ be a derivation of a word w. With this derivation two sequences of nonterminal cuts could be associated. The first one is that of all nonterminal cuts of this derivation. The second one is a sequence containing only communication cuts of the derivation. We will call the sequence of nonterminal cuts the trace of the derivation (resp. trace) and denote $T(\mathcal{D}(w))$. The sequence containing only communication cuts of the derivation will be called the communication sequence of the derivation and will be denoted by $C(\mathcal{D}(w))$.

Note that for a given $PCGS$ Π it is meaningful to speak about the set $\mathcal{D}(\Pi)$ of all derivations of Π, the set $\mathcal{T}(\Pi)$ of all traces of Π, and the set $\mathcal{C}(\Pi)$ of all communication sequences of Π. For a natural number k, $\mathcal{C}(\Pi, k)$ denotes the set of all communication sequences of Π with at most k communications in it. We note that there is no one-to-one relation between the sets $\mathcal{D}(\Pi)$ and $\mathcal{T}(\Pi)$. To every $d \in \mathcal{D}(\Pi)$ the $T(d) \in \mathcal{T}(\Pi)$ is given unambiguously. But, for any $t \in \mathcal{T}(\Pi)$, there is a set $T^{-1}(t)$ such that for every $d \in \mathcal{T}^{-1}(\Pi)(t)$, $T(d) = t$. The cardinality of the set $\mathcal{T}^{-1}(\Pi)(t)$ can be bounded by a constant that depends on Π and t. The relation between the sets $\mathcal{D}(\Pi)$ and $\mathcal{C}(\Pi)$ is not unambiguous, too. For every $d \in \mathcal{D}(\Pi)$ there exists precisely one $C(d) \in \mathcal{C}(\Pi)$. But there are $c \in \mathcal{C}(\Pi)$ for which the cardinality of the set $\mathcal{C}^{-1}(\Pi)(c) = \{d \in \mathcal{D}(\Pi) | C(d) = c\}$ is bounded by infinity only.

Let $N(\mathcal{D}(w)) = N(C_1), N(C_2), \ldots, N(C_t)$ be a sequence of nonterminal cuts of a computation $\mathcal{D}(w)$. Let $i, j \in \{1, 2, \ldots, t\}, i < j$ such that $N(C_i) = N(C_j)$ holds. Then the sequence of steps corresponding to the subderivation $C_i \rightarrow C_{i+1}, \ldots, C_{j-1} \rightarrow C_j$ forms a cycle of the derivation. If none of the nonterminal cuts $N(C_i), \ldots, N(C_j)$ contains a communication symbol, then the cycle of the derivation is called the generative cycle of the derivation (resp. generative cycle).

Let $I = \{t_1, t_2, ..., t_k\}, t_1 < t_2 < ... < t_k$, be the set of all communication steps of $\mathcal{D}(w)$.
• the i-th generative section of $\mathcal{D}(w)$, $1 \leq i \leq k+1$, is the subsequence $t_{i-1}+1$, $t_{i-1}+2, \ldots, t_i - 1$ of derivation steps, $t_0 = 0, t_{k+1} - 1 = t$.
• Let $C_{t_j+1} = (\alpha_1 A_1, \alpha_2 A_2, \ldots, \alpha_n A_n)$ be the configuration just at the beginning of the $j-th$ generative section (after the succesfuly performed $j-th$ communication step; resp. at the beginning of the derivation $\mathcal{D}(w)$).
Let $C_{t_{j+1}} = (\alpha_1 \beta_1 B_1, \alpha_2 \beta_2 B_2, \ldots, \alpha_n \beta_n B_n)$ be that of the end of the j-th generative section.
Then $g(i,j)(\mathcal{D}(w))$ is the substring β_i of $\alpha_i \beta_i B_i$. We prefer the abbreviation $g(i,j)$ if it is not misleading. The terminal word w generated in the derivation $\mathcal{D}(w)$ can be composed using some of $g(i,j)(\mathcal{D}(w))'s$. Thus, for a given $\mathcal{D}(w)$ and $i, j \in \mathbb{N}$ one can speak about the number of occurrences of $g(i,j)(\mathcal{D}(w))$ in the word w. But, in fact, this number does not depend on the whole derivation. It depends on the communication sequence $C(\mathcal{D}(w))$ only, so the following denotation is correct.
• Let $CC = (C_1, C_2, \ldots, C_p)$, $C_r \in \{\varepsilon, Q_1, \ldots, Q_m\}^m$ for some natural numbers r, m, p (where m can be considered as the number of component grammars of some $PCGS$ and CC can be considered as a communication sequence of the $PCGS$). Then we will denote

by $\underline{n(i,j)(CC)}$ (resp. $n(i,j)$) the number of occurrences of $g(i,j)(\mathcal{D}(w))$ (resp. $g(i,j)$) in w for every $\mathcal{D}(w)$ with communication sequence equal to CC.

The last notion recalled here is the notion of communication complexity measure, as defined in [Par 93].

Definition 2.3 *Let* Π *be a* $PCGS(m)$*,* $L = L(\Pi)$ *and* $\mathcal{D}(\Pi)$ *be the set of all derivations of* Π*. Let* $\mathcal{D}(\Pi, w)$ *be the set of all derivations of a word* w *by* Π*. Let* $D_\Pi(w) = C_0, C_1, C_2, \ldots, C_t, C_i = (C_{i1}, C_{i2}, \ldots, C_{im})$ *be a derivation of a terminal word* w *in* Π *and* $I = \{t_1, t_2, ..., t_k | t_1 < t_2 < ... < t_k\}$ *be the set of those communication steps of the derivation* $D_\Pi(w)$ *for which the* $(t_i - 1)$*-st step,* $i \in \{1, \ldots, k\}$*, is generative. Then*

$$\begin{aligned}
\underline{com(D_\Pi(w))} &= \sum_{i=1}^{k} |C_i|_K \\
\underline{com(w, \Pi)} &= max\{com(D) \mid D \in \mathcal{D}(\Pi, w)\} \\
\underline{com(n, \Pi)} &= max\{com(w, \Pi) \mid |w| = n\}.
\end{aligned}$$

Finally,

$$\underline{COM(f(n))} = \{L(\Pi) | \forall n \in \mathbb{N} : com(n, \Pi) \leq f(n)\}.$$

Let us denote by $\underline{x\text{-}PCGS(m)\text{-}f(n)}$ the $PCGS$ of degree m with the communication graph in the class of graphs x and at most $f(n)$ communications during the generation of any word of the length n. We consider $x \in \{$*tree, dag(directed acyclic graph), c*$\}$ where by tree-$PCGS$ one assumes that the output grammar G_1 is in the root of the tree, and *c*-$PCGS$s are the centralized $PCGS$s introduced in [PS 89]. We use the notation x-$PCGS$-k instead of x-$PCGS$-$g(n)$ if $g(n) = k$ for every $n \in \mathbb{N}$.

3 dag-$PCGS$ versus tree-$PCGS$

Pardubská [Par 93] has shown that unrestricted communication structures (complete, directed graphs) are more powerful than the communication structures restricted by the topology of directed acyclic graphs, and Lukáč [Luk 92] has shown that tree-$PCGS$s are more powerful than *c*-$PCGS$s. Here, we make a further step in getting a hierarchy on communication structures by proving *dag*-$PCGS$s are more powerful that tree-$PCGS$s.

Theorem 3.1 *The language* $L = \{ww \mid w \in \Sigma^*\}, |\Sigma| \geq 2$*, can be generated by a dag-PCGS of degree 3, and cannot be generated by any tree-PCGS.*

Proof. We start the proof of Theorem 3.1 by proving the upper bound.

Proposition 3.1.1 *The language*

$$L = \{ww | \; w \in \Sigma^*\}$$

can be generated by a regular dag-PCGS of degree 3.

Proof. Consider the dag-PCGS $\pi = (G_1, G_2, G_3)$ where $G_i = (N_i, \Sigma, S_i, P_i)$, $1 \leq i \leq 3$ and

$$
\begin{aligned}
N_1 = \;& \{S_1, Z, Z_1, Q_2, Q_3\},\\
N_2 = \;& \{S_2, Z, Z_1, Q_3\},\\
N_3 = \;& \{S_3, Z\},\\
P_1 = \;& \{S_1 \longrightarrow S_1, S_1 \longrightarrow Q_3, Z \longrightarrow Q_2, Z_1 \longrightarrow \lambda\},\\
P_2 = \;& \{S_2 \longrightarrow S_2, S_2 \longrightarrow Q_3, Z \longrightarrow Z_1\},\\
P_3 = \;& \{S_3 \longrightarrow aS_3 |\ a \in \Sigma\} \cup \{S_3 \longrightarrow Z\}.
\end{aligned}
$$

A terminating derivation according to π will have the following form:

$$
\begin{array}{llll}
(S_1, S_2, S_3) & \Longrightarrow^* & (S_1, S_2, wS_3) & \Longrightarrow\\
(Q_3, Q_3, wZ) & \Longrightarrow & (wZ, wZ, S_3) & \Longrightarrow\\
(wQ_2, wZ_1, \alpha) & \Longrightarrow & (wwZ_1, S_2, \alpha) & \Longrightarrow\\
(ww, \beta, \alpha'), & & &
\end{array}
$$

where $w \in \Sigma^*$ and $\beta \in N_2$, $\alpha, \alpha' \in \Sigma^* N_3$.

Informally, the word $w \in \Sigma^*$ is generated by the grammar G_3 and it is then simultaneously read by G_1 and G_2. In this way, two identical copies of w are created. Then the master grammar G_1 appends w from G_2 to its own w.

It is obvious from the above considerations that the dag-PCGS π generates the requested language. This completes the proof of Propositon 3.1.1 $\square$

The proof of Theorem 3.1 continued. In the sequel we will prove that the language L cannot be generated by any tree-PCGS. Before starting this main result, some notations and auxiliary results are needed.

Consider a tree-PCGS of degree $k \geq 1$, $\pi = (G_1, G_2, \ldots, G_k)$. The first component is the root of the tree. For a component x different from the root, $2 \leq x \leq k$, $\mathrm{fa}(x)$ will denote the father of the component x in the communication tree. The set $\mathrm{Des}(x)$ denotes the descendants of component x :

$$
\mathrm{Des}(x) = \begin{cases} \{x\}, & \text{if } x \text{ is a leaf, and}\\ \{x\} \cup \left(\bigcup\limits_{x = \mathrm{fa}(y)} \mathrm{Des}(y) \right), & \text{otherwise.} \end{cases}
$$

Let D be a derivation according to π, starting with the initial configuration c_0 and finishing with a configuration c_m (not necessarily terminal):

$$D : c_0 \Longrightarrow c_1 \Longrightarrow c_2 \Longrightarrow \ldots \Longrightarrow c_m.$$

In the following we count only the rewriting steps. Communication steps "come for free". Let d denote the total number of rewriting steps in D.

For every two components $1 \leq x, y \leq k$, where y is a descendant of x, we inductively define a number $F_{D,x}(y)$:

$$F_{D,x}(y) = \begin{cases} d\ (= \text{the number of rewriting steps in } D), & \text{if } y = x, \\ \text{the number of rewriting steps before the last communication step communicating } y \text{ to } \mathrm{fa}(y) \text{ which is preceeded by at most } F_{D,x}(\mathrm{fa}(y)) \text{ rewriting steps. This means } F_{D,x}(y) \leq F_{D,x}(\mathrm{fa}(y)). \text{ If no such communication step exists, } F_{D,x}(y) = 0. & \text{if } y \neq x. \end{cases}$$

Intuitively, $F_{D,x}(y)$ indicates the number of rewriting steps performed in the component y, that will have an effect on the string of component x of the last configuration c_m (component y has time to communicate its string to x before the derivation D is finished).

The following lemma contains the main idea used in proving that L cannot be generated by a tree-PCGS. Informally, if the conditions of Lemma 3.1.1 are met, we can interchange any corresponding sentential forms of two reachable configurations and the result is still a reachable configuration. This will help in showing that, if a tree-PCGS can generate words of the form ww, then it can generate (due to the possibility of interchanging given by Lemma 3.1.1) also strings that are not of the form ww.

Lemma 3.1.1 *Let $\pi = (G_1, \ldots, G_k)$ be a regular tree-PCGS of degree $k \geq 1$, and*

$$D_1 : (S_1, \ldots, S_k) \Longrightarrow^* (\alpha_1^1, \ldots, \alpha_k^1) \text{ and } D_2 : (S_2, \ldots, S_k) \Longrightarrow^* (\alpha_1^2, \ldots, \alpha_k^2)$$

be two derivations according to π. If $F_{D_1,x}(y) = F_{D_2,x}(y)$ for all $1 \leq x, y \leq k$ with $y \in Des(x)$, then for any $i_1, i_2, \ldots i_k \in \{1, 2\}$ there exists a derivation

$$D : (S_1, \ldots, S_k) \Longrightarrow^* (\alpha_1^{i_1}, \ldots, \alpha_k^{i_k})$$

according to π. Moreover, $F_{D,x}(y) = F_{D_1,x}(y)$ for all $y \in Des(x)$.

Proof. Let x be a fixed component, different from the root, $2 \leq x \leq k$. We show how to construct a valid derivation for $(\alpha_1^{i_1}, \ldots, \alpha_k^{i_k})$ where $i_y = 2$ if $y \in \mathrm{Des}(x)$ and $i_y = 1$ otherwise. This means we can interchange the sentential forms of all components belonging to an arbitrary subtree of the communication tree. The general case follows by repeating the process for smaller and smaller subtrees.

Let us denote shortly $F(y)$ instead of $F_{D_1,\mathrm{fa}(x)}(y)$ (the number of rewriting steps performed in component y that have an effect outside the subtree rooted at x. According to the hypothesis of the lemma this number is equal in D_1 and D_2.) In derivation D an arbitrary component y will use on j'th rewriting step the same rewriting rule as D_i used on its j'th rewriting step, where

$$D_i = \begin{cases} D_1, & \text{if } y \notin \mathrm{Des}(x), \text{ or } j \leq F(y), \\ D_2, & \text{if } y \in \mathrm{Des}(x) \text{ and } j > F(y). \end{cases}$$

In other words, components not in the subtree rooted at x will follow the same rewriting steps as they do in D_1, while components that are in the subtree will first follow D_1 but after $F(y)$ steps they start to imitate D_2. That this is possible, and that it will lead to the correct outcome, will be demonstrated below.

Let $s_j(y)$, $s_j^{(1)}(y)$ and $s_j^{(2)}(y)$ denote the sentential forms in component y immediately after the j'th rewriting step in D, D_1 and D_2, respectively. For $j = 0$ we define $s_0(y) = s_0^{(1)}(y) = s_0^{(2)}(y) = S_y$, the axiom of component y. We show using induction on j that for all components y

$$s_j(y) \quad = \quad \begin{cases} s_j^{(1)}(y), & \text{if } y \notin \mathrm{Des}(x), \text{ or } j \leq F(y), \\ s_j^{(2)}(y), & \text{if } y \in \mathrm{Des}(x) \text{ and } j > F(y). \end{cases}$$

This proves the lemma, since after m rewriting steps all components in the subtree rooted in x will have the same sentential form as in D_2, while all other components have sentential forms from D_1.

If $j = 0$ the claim is trivially true: all sentential forms are equal to the axiom of the corresponding component. Assume that the claim has been proved for j, and consider the string of an arbitrary component y after $j + 1$ rewriting steps:

1° $\quad y \notin \mathrm{Des}(x)$ or $j < F(y)$: We first have to show that after all possible communications between j'th and $(j + 1)$'st rewriting steps, the sentential form in component y is the same in D and in D_1. There are different alternatives depending on the type of communications that occur. If component y communicates its string to its father, it resumes its axiom S_y. But the same happens in derivation D_1: according to the inductive hypothesis, after j rewriting steps there is in $\mathrm{fa}(y)$ the same nonterminal — namely Q_y — in both D and D_1.

If y is not communicated to $\mathrm{fa}(y)$, but y itself "reads" the sentential form of one of its children, say z, then necessarily $j \leq F(z)$. Namely, according to the definitions of the numbers $F_{D,x}(y)$, there cannot be a communication from z to y between rewriting steps $F(z)$ and $F(y)$. The same applies to any other components which communicate simultaneously their strings — via z — to y. According to the inductive hypothesis the same sentential forms are communicated to y in D and in D_1.

So, regardless what happens between the rewriting steps j and $j + 1$, in the beginning of the $(j+1)$'st rewriting step the component y has the same sentential form in both D and D_1. Since the same rewriting rule is applied, the same sentential form is reached.

2° $\quad y \in \mathrm{Des}(x)$ and $j = F(y)$: After the j'th rewriting step, component y communicates its string to its father in both D_1 and D_2. Therefore the $(j+1)$'st rewriting in component y is done on the axiom. Since the same rule is used in D as in D_2 (because $j+1 > F(y)$), the same sentential form is reached.

3° $\quad y \in \mathrm{Des}(x)$ and $j > F(y)$: Since the rewriting rule of D_2 is used, the same sentential form is reached, provided the sentential forms were equal just before the rewriting. This is indeed the case according to the inductive hypothesis (possible communications cannot ruin it, as can be seen in the same fashion as in case 1°). This completes the proof of Lemma 3.1.1. □

The proof of Theorem 3.1 continued. Consider the minimal terminating derivation

$$D : c_0 \Longrightarrow c_1 \Longrightarrow \ldots \Longrightarrow (ww, \alpha_2, \ldots, \alpha_k)$$

for a word $ww \in L$. A component x is called *useful* in configuration c_i if it eventually communicates its string to the master grammar G_1 (it contributes to ww). In our earlier notation this is equivalent to $F_{D,1}(x) \geq$ the number of rewriting steps among the first i derivation steps. Note that usefulness depends on the derivation D.

Now, we prove still one useful technical lemma.

Lemma 3.1.2 *Let $\pi = (G_1, \ldots, G_k)$ be a tree-PCGS. There exist positive constants a and b such that for every minimal derivation D and natural number n, after the first n rewriting steps, the length l_n of the longest terminal string in any useful component satisfies $\lfloor a \cdot n \rfloor \leq l_n \leq b \cdot n$.*

Proof. Consider the sum of the lengths of the terminal strings in all useful components of D, denoted by l'_n. On each rewriting step l'_n can increase by at most $b = k\times$ the maximum number of terminal letters on the right hand sides of all productions. l'_n does not change on communication steps. Therefore $l_n \leq l'_n \leq b \cdot n$.

Consider then the lower limit for l'_n. On a minimal derivation there can be at most N^k consecutive rewriting steps that do not increase l'_n, where N denotes the number of non-terminals (a rewriting step that does not increase l'_n can only rename the non-terminals of useful components. In $N^k + 1$ steps the same configuration would appear twice.) Each terminal string can be communicated fewer than k times before it reaches the root. Therefore there must be fewer than k^2 communication steps separated by at most N^k consecutive rewriting steps. Altogether, l'_n has to increase in every $k^2 \cdot N^k$ rewriting steps. Because $l_n \geq l'_n/k$, we can choose $a = 1/(k^3 N^k)$. This completes the proof of Lemma 3.1.2. □

The proof of Theorem 3.1 continued. Two configurations along two terminating derivations are said to be *different in their useful components* if one of the following conditions holds:

(i) there is a component which is useful in exactly one of the configurations,

(ii) the configurations differ in a sentential form of some useful component.

In other words the configurations are considered similar if the same components are useful in both of them, and the contents of useful components are equal.

Now, we are prepared to prove the lower bound of Theorem 3.1.

Lemma 3.1.3 *The language $L = \{ww \mid w \in \Sigma^*\}$ cannot be generated by any tree-PCGS of degree $k \geq 1$.*

Proof. Assume that there exists a tree-PCGS $\pi = (G_1, \ldots, G_k)$ such that $L(\pi) = L$, and let a, b be the numbers defined in Lemma 3.1.2. Number b can be assumed to be an integer.

Let $f(n)$ denote the number of different configurations reachable in exactly n rewriting steps on any minimal derivation for ww, $|w| = bn$. By different configurations we mean different in their useful components.

There are two possible cases to consider, depending on the growth rate of $f(n)$: either $f(n)$ is bounded by a polynomial, or $f(n)$ grows faster than any polynomial. We will show that both cases lead to contradictions, which implies that our assumption that L can be generated by π was false.

Case 1. There exists a polynomial $p(n)$ such that $f(n) < p(n)$. In this case choose n big enough so that $p(n) < |\Sigma|^{\lfloor an \rfloor}/(bn)$. This means that there are less than $|\Sigma|^{\lfloor an \rfloor}/(bn)$ different configurations, say $e_1, e_2, \ldots, e_h$, $h < |\Sigma|^{\lfloor an \rfloor}/(bn)$, reachable in n rewriting steps on minimal derivations for words ww, $|w| = bn$, and each such configuration contains a useful component whose string contains at least $\lfloor an \rfloor$ terminal letters. Pick one such string for each e_i. Let $w_1, w_2, \ldots, w_h$ be their prefixes of length $\lfloor an \rfloor$.

Every ww, $|w| = bn$, contains one w_i as a subword. Each w_i can however be a subword of at most $bn|\Sigma|^{bn-\lfloor an \rfloor}$ of them. Indeed, the word w_i has to start in one of the $|w| = bn$ first positions in ww. It fixes $|w_i| = \lfloor an \rfloor$ letters of w — only the remaining $|w| - |w_i| = bn - \lfloor an \rfloor$ letters can be chosen arbitrarily.

Therefore only $h \cdot bn|\Sigma|^{bn-\lfloor an \rfloor}$ of the words ww, $|w| = bn$, can contain some w_i as a subword. But

$$h \cdot bn|\Sigma|^{bn-\lfloor an \rfloor} < |\Sigma|^{\lfloor an \rfloor}/(bn) \cdot bn|\Sigma|^{bn-\lfloor an \rfloor} = |\Sigma|^{bn},$$

the number of words ww with $|w| = bn$, a contradiction.

Case 2. The function $f(n)$ grows faster than any polynomial. Then there exists (for large enough n) two different configurations c_1, c_2 reachable in n rewriting steps along minimal terminating derivations D_1, D_2 for some w_1w_1, w_2w_2, respectively, where $|w_1| = |w_2| = bn$, such that in c_1, c_2 :

- the same components are useful (at most 2^k alternatives),
- the strings of the useful components have the same nonterminals (at most $(|N|+1)^k$ alternatives),
- the lengths of the strings of the useful components are the same (at most $(bn)^k$ alternatives), and
- $F_{D'_1,x}(y) = F_{D'_2,x}(y)$ for all x, y, $y \in \mathrm{Des}(x)$, where D'_i denotes the initial part (containing n rewriting steps) of D_i that produces c_i, $i = 1, 2$ (at most n^{k^2} alternatives).

Namely, there are at most

$$p(n) = 2^k \cdot (|N| + 1)^k \cdot (bn)^k \cdot n^{k^2}$$

different choices for the four items above. Clearly $p(n)$ is a polynomial of n, and thus exceeded by $f(n)$ for some n. We may choose n so large that $\lfloor an \rfloor \geq 1$.

According to Lemma 3.1.1, any string of c_1 can be replaced with the corresponding string of c_2. Since c_1 and c_2 are different (meaning that they differ in the terminal string of one of their useful components) one terminal string, say α, in a useful component of c_1 can be replaced by a different, but equally long, word β. Note that $1 \leq |\alpha| = |\beta| \leq bn$.

Continuing the derivation according to D_1, a word is produced, that differs from $w_1 w_1$: exactly one occurence of α has been replaced by β. (At least one occurence since the change was done in a useful component; at most one occurence, since the communication graph is a tree.) This is a contradiction: one cannot obtain a word in L after replacing in ww a nonempty subword of length $\leq |w|$ with an equally long but different word. $\Box$

The assertion of Lemma 3.1.3 completes the proofof Theorem 3.1. $\Box$

Note, that Theorem 3.1 also shows that the weakness of a communication structure cannot be compensated by any increase in the number of components.

Corollary 3.1

$$\mathcal{L}(\textit{dag-PCGS}(3)) - \bigcup_{m \in \mathbb{N}} \mathcal{L}(\textit{tree-PCGS}(m)) \neq \emptyset.$$

Corollary 3.2 *for any positive integer $k \geq 3$,*

$$\mathcal{L}(\textit{tree-PCGS}(k)) \subsetneq \mathcal{L}(\textit{dag-PCGS}(k)).$$

4 Communication complexity of dag-$PCGS$s

In [HKK 93] it is shown that $k+1$ communications are more powerful than k communications for tree-$PCGS$s, and Pardubská [Par 93] has established the same hierarchy for dag-$PCGS$s. The next result shows, that these hierarchies have no continuation in a hierarchy of unbounded functions, i.e., the communication complexity of a dag-$PCGS$ is either a constant or a linear function. Note that each $L \in$ dag-$PCGS$ can be generated with $O(n)$ communication complexity.

Theorem 4.1 *Le $f(n)$ be a nondecreasing function from $\mathbb{N}$ to $\mathbb{N}$ and let Π be a dag-PCGS-$f(n)$ generating a language L such that $L \notin \mathcal{L}($ dag-PCGS-$O(1))$. Then $f(n) = \Omega(n)$.*

Before starting the proof of Theorem 4.1 we give two useful obervations and one definition.

Observation 4.1 *Let Π be a PCGS. Then there exists a constant $\ell(\Pi)$ such that for every $i, j \in \mathbb{N}$ and every derivation $\mathcal{D}(w)$ of Π the following is true:*
$|g(i,j)(\mathcal{D}(w))| \geq \ell(\Pi) \implies$ *there is a generative cycle in the $j-th$ generative section of the derivation $\mathcal{D}(w)$.*

Proof. It is sufficient to choose $\ell(\Pi)$ greater than the number of all nonterminal cuts of Π multiplied by the maximum of lengths of all right sides of rules in Π. Note, that $\ell(\Pi)$ depends only on Π (namely on the degree of Π, on the number of nonterminals in Π and on the lengths of rules of Π). □

Observation 4.2 *Let Π be a $dag - PCGS$. Then there exists a constant $n(\Pi)$ such that for all derivations of Π and all $i, j \in \mathbb{N}, n(i,j) \leq n(\Pi)$.*

Proof. For every i, j, $n(i,j)$ is at most the number of distinct paths between G_i and G_1. Since the communication structure of Π is acylic, $n(\Pi)$ is finite and can be bounded by $2^k \leq 2^{m^2}$, where k is the number of edges of the communication graph of Π and m is the degree of Π. □

Definition 4.1 *Let Π be a $PCGS$ and $\mathcal{D}(w)$ be a derivation of a word $w \in L(\Pi)$. We say the derivation $\mathcal{D}(w)$ is reduced, if every repetition of a generative cycle of $\mathcal{D}(w)$ leads to a longer terminal word.*

Note, that for each derivation $\mathcal{D}(w)$ of w one can construct the reduced derivation $\overline{\mathcal{D}}(w)$ of w in such a way that $\mathcal{D}(w)$ and $\overline{\mathcal{D}}(w)$ have the same communication sequence, so the same number of communications too.
Proof of Theorem 4.1. Assume, for the sake of contradiction, that Π is a $dag - PCGS(m) - f(n)$ generating the language $L \notin \mathcal{L}(dag - PCGS - O(1))$, $f(n) \neq \Omega(n)$. Since the function$f(n)$ is unbounded ($L \notin \mathcal{L}(dag - PCGS - O(1))$), nondecreasing and $f(n) \neq \Omega(n)$ we can suppose $f(n) = o(n)$.

According to the assumption about the language L the communication complexity of Π cannot be bounded by any constant. So, for every natural number k there is a word ω such that some of its derivations use at least k communications. Hence, for every $k \in \mathbb{N}$, we can fixe one of them in the following way.

(1) ω_k is the first word in the lexicographical order (therefore one of the shortest ones) that contains at least k communications in some of its reduced derivations.

As we cannot bound the function $f(n)$ by any constant and f is nondecreasing the following is true.

(2) The set $\{\omega_k | k \in \mathbb{N}\}$ is infinite.

Since $f(n) = o(n)$,

(3) for every $\ell \in \mathbb{N}$ there exists a positive integer $n_\ell \in \mathbb{N}$ such that $\forall n \geq n_\ell$: $n \geq \ell \cdot f(n)$.

In what follows we need a word ω_β satisfying (1) for some suitable constant β and moreover, the length of the word ω_β has to satisfy (3) for some other constant α. Let

$\alpha = 4m \cdot \ell(\Pi) \cdot n(\Pi)$, where

m is the number of component grammars of Π

$\ell(\Pi)$ is the constant bounding the length of words generated without the repetition of any nonterminal cut (see Observation 4.1).

$n(\Pi)$ is the constant bounding the number of occurences of individual $g(i,j)$'s in the resulting terminal word (see Observation 4.2).

Let w be a ω_β (see (1)) such that $|\omega_\beta| \geq n_\alpha$ for some $\beta \in \mathbb{N}$. Let $|w| = n$. According to (2), the word w with the required length exists. Suppose $\mathcal{D}(w)$ is a reduced derivation of w containing at least β communications (but at most $f(n)$, $(n/f(n)) \geq \alpha$). Since the derivation $\mathcal{D}(w)$ contains at most $f(n)$ communications,

(4) at most $(f(n)+1)\cdot m$ different $g(i,j)$'s form the word w.

Then, $(n)/((f(n)+1)\cdot m \cdot n(\Pi))$ is the lower bound on the average length of $g(i,j)$ for the word w.

$$\textbf{(5)}\quad \frac{n}{(f(n)+1)\cdot m\cdot n(\Pi)} \geq \frac{n}{2\cdot m\cdot f(n)\cdot n(\Pi)} \geq \frac{\alpha\cdot f(n)}{2\cdot m\cdot f(n)\cdot n(\Pi)} = \frac{\alpha}{2\cdot m\cdot n(\Pi)}$$
$$= \frac{4m\cdot\ell(\Pi)\cdot n(\Pi)}{2\cdot m\cdot n(\Pi)} = 2\cdot\ell(\Pi)$$

¿From (5) it follows that there exist some values i_0, j_0 such that $|g(i_0,j_0)| > \ell(\Pi)$. Following Observation 4.1 we get that the derivation $\mathcal{D}(w)$ contains a generative cycle. Removing this generative cycle from the derivation $\mathcal{D}(w)$ another derivation $\mathcal{D}(w')$ of a terminal word w' is obtained. Since $\mathcal{D}(w)$ is reduced, the derivation $\mathcal{D}(w')$ is reduced too and moreover the word w' is shorter than the word w. But the number of communications in both derivations $\mathcal{D}(w)$ and $\mathcal{D}(w')$ is the same (no generative cycle contains communications). This fact contradicts the assumption that w is the shortest word containing at least β communications in some of its reduced derivations. □

References

[PS 89] Gh.Paun, L.Santean: Parallel communicating grammar systems: the regular case. *Ann.Univ.Buc.Ser.Mat.-Inform.*37 vol.2(1989), pp.55-63.

[HU 69] J.E.Hopcroft, J.D.Ullman: Formal Languages and Their Relation to Automata. Addison-Wesley Publishing Company, Reading, Massachusetts, 1969.

[HR 75] G.T.Herman, G.Rozenberg: Developmental Systems and Languages, North-Holland, Amsterdam, 1975.

[HKK 93] J.Hromkovič, J.Kari, L.Kari: Some hierarchies for the communication complexity measures of cooperating grammar systems. *Theoretical Computer Science*, to appear (extended abstract in: Proc. of the MFCS'93 Lecture Notes in Comp. Science 711, pp.495 - 505).

[Luk 92] M.Lukáč: About two communication structures of PCGS.Master Thesis(1992), Dept.of Comp.Sci., Faculty of Mathematics and Physics, Comenius University, Bratislava (in Slovak).

[Par 92] D.Pardubská: The communication complexity hierarchy of parallel communicating systems. Presented at IMYCS'92.

[Par 93] D.Pardubská: On the power of communication structure for distributive generation of languages. In: Developments in Language Theory. Preproceedings, University of Turku, Turku 1993, pp.30 - 32.

[RS 80] G.Rozenberg, A.Salomaa: The Mathematical Theory of L Systems. Academic Press, 1980.

[San 90] L.Santean: Parallel Communicating Systems. *EATCS Bulletin*, Num.42 (1990), 160 - 171.

[SK 92] L.Santean, J.Kari: The impact of the number of cooperative grammars on the generative power, *Theoretical Computer Science* 98(1992) pp. 249-263.