Left-Forbidding Cooperating Distributed Grammar Systems

Filip Goldefus[a], Tomáš Masopust[b,*], Alexander Meduna[a]

[a] *Faculty of Information Technology, Brno University of Technology*
Božetěchova 2, Brno 61266, Czech Republic
[b] *Institute of Mathematics, Czech Academy of Sciences*
Žižkova 22, 616 62 Brno, Czech Republic

Abstract

A left-forbidding grammar, introduced in this paper, is a context-free grammar, where a set of nonterminal symbols is attached to each context-free production. Such a production can rewrite a nonterminal provided that no symbol from the attached set occurs to the left of the rewritten nonterminal in the current sentential form. The present paper discusses cooperating distributed grammar systems with left-forbidding grammars as components and gives some new characterizations of language families of the Chomsky hierarchy. In addition, it also proves that twelve nonterminals are enough for cooperating distributed grammar systems working in the terminal derivation mode with two left-forbidding components (including erasing productions) to characterize the family of recursively enumerable languages.

Keywords: Cooperating distributed grammar system, cooperating derivation mode, left-forbidding grammar, generative power, descriptional complexity.

1. Introduction

Recently, formal language theory has intensively investigated various types of cooperating distributed grammar systems, which are devices consisting of several cooperating components represented by grammars or other rewriting mechanisms that work in some prescribed derivation modes (the reader is referred to [1, 2, 3] for more information). The present paper continues with this investigation by discussing cooperating distributed grammar systems based upon components represented by slightly modified context-free grammars that perform left-restricted derivations, which fulfill an important role in most computer science areas that make use of grammars. Indeed, in practice, most top-down parsers are based upon left-restricted derivations, and, in theory, these left-restricted derivations frequently simplify the discussion concerning the performance of various grammatical derivations. In addition, they often result in an increase of the generative power, so the grammar systems that work in this left-restricted way are significant from a theoretical point of view as well.

More precisely, each component of the cooperating distributed grammar systems under investigation is a *left-forbidding grammar*, which in essence represents a context-free grammar in which a set of nonterminal symbols is attached to each context-free production. Such a production can rewrite a nonterminal provided that no symbol from its attached set occurs to the left of the rewritten nonterminal in the current sentential form. Here is the difference compared to the random forbidding context grammars (see [4, 5]) because, in random forbidding context grammars, symbols from the attached set are looked up in the whole sentential form. As the key topic of investigation, we concentrate our attention on the generative power of cooperating distributed grammar systems with left-forbidding components with respect to the number of components, derivation modes, and erasing productions. Furthermore, we also discuss the descriptional complexity of the erasing variant of these cooperating distributed grammar systems working in the terminal derivation mode.

*Corresponding author.
Email addresses: `igoldefu@fit.vutbr.cz` (Filip Goldefus), `masopust@math.cas.cz` (Tomáš Masopust), `meduna@fit.vutbr.cz` (Alexander Meduna)

Specifically, with respect to the terminal derivation mode (t-mode, for short), $=k$-mode, and $\geq k$-mode of cooperation, for all $k \geq 2$, this paper demonstrates that with a single component, cooperating distributed grammar systems with left-forbidding components generate only the family of context-free languages, while with two or more left-forbidding components, they generate the whole family of recursively enumerable languages. In addition, with two or more left-forbidding components and without erasing productions, these cooperating distributed grammar systems characterize the family of context sensitive languages. In comparison with the previous results, these main results are of some interest because cooperating distributed grammar systems with two context-free components generate only the family of context-free languages (considering the terminal derivation mode), or a subfamily of the family of matrix languages (considering the $=k$-mode and $\geq k$-mode, for $k \geq 2$), see Theorem 3.1 in [3]. Concentrating its attention on the economical description of these systems, this paper demonstrates that cooperating distributed grammar systems working in the terminal derivation mode with two left-forbidding components and no more than twelve nonterminals are computationally complete.

Recently, formal language theory has discussed some random forbidding/permitting context grammars that are not as powerful as random context grammars (see [5] and [6], respectively), and some cooperating distributed grammar systems with random forbidding/permitting context components working in the terminal derivation mode that are as powerful as random context grammars (see [7] and [8], respectively). In view of these results, the results of the present paper are of some interest because the cooperation increases the generative power of cooperating distributed grammar systems with left-forbidding components from the power of context-free grammars to the power of context sensitive or phrase structure grammars. On the other hand, however, this increase does not hold for cooperating distributed grammar systems with random context components because they are as powerful as random context grammars (see [9, 10]). Consequently, although random context grammars are more powerful than left-forbidding grammars, cooperating random context grammars are not as powerful as cooperating left-forbidding grammars.

Finally, left-permitting grammars, whose precise generative power has not been established yet, are defined analogously as left-forbidding grammars. Even though these grammars are more powerful than left-forbidding grammars, cooperating distributed grammar systems with left-permitting components working in the terminal derivation mode characterize the same family of languages as cooperating distributed grammar systems with left-forbidding components (see [8] for more information and results related to the descriptional complexity).

The organization of this paper is as described below. In the next section, preliminary fundamental results and definitions from formal language theory needed in our paper are recalled. Section 3 presents the main results of this paper, and Section 4 then compares these results to the related results and the Chomsky hierarchy.

2. Preliminaries and Definitions

In this paper, we assume that the reader is familiar with formal language theory (see [11]). For a finite nonempty set (an alphabet) V, let V^* represent the free monoid generated by V. Let the unit of V^* be denoted by ε, and let $V^+ = V^* - \{\varepsilon\}$. For $w \in V^*$, let $|w|$ denote the length of w, w^R denote the mirror image of w, $alph(w)$ denote the set of all symbols occurring in w, and for any two strings $u, v \in V^*$, let $shuffle(u, v) = \{x_1y_1x_2y_2 \dots x_ny_n : n \geq 1,\ u = x_1x_2 \dots x_n,\ v = y_1y_2 \dots y_n,\ x_i, y_i \in V^*, \text{ for } i = 1, \dots, n\}$ be a shuffle of letters of u and v keeping the order of letters in u and v unchanged. Let $\mathscr{L}_{CF}$, $\mathscr{L}_{CS}$, $\mathscr{L}_{RE}$, $\mathscr{L}_{ET0L}$, and $\mathscr{L}^{\varepsilon}_{MAT}$ denote the families of context-free, context sensitive, recursively enumerable, $ET0L$, and matrix languages (generated by matrix grammars with context-free productions), respectively.

A *state grammar* (see [12]) is a construct $G = (N, T, Q, P, S, q_0)$, where N, T, and Q are pairwise disjoint alphabets of nonterminals, terminals, and states, respectively, $V = N \cup T$ is the total alphabet, $S \in N$ is the start symbol, $q_0 \in Q$ is the start state, and P is a finite set of productions of the form $(A, p) \rightarrow (x, q)$, where $p, q \in Q$ are states, $A \in N$ is a nonterminal symbol, and $x \in V^*$ is a string of symbols.

For any two strings $u, v \in V^*$ and any two states $p, q \in Q$, we define the relation $(uAv, p) \Rightarrow (uxv, q)$ provided that

1. $(A, p) \rightarrow (x, q) \in P$, and
2. for every $(B, p) \rightarrow (y, t) \in P$, we have that $B \notin alph(u)$.

Thus, according to condition (2), the leftmost derivation is considered, i.e., the leftmost nonterminal that can be replaced in the current state has to be replaced. As usual, the relation $\Rightarrow$ is extended to $\Rightarrow^n$, for $n \geq 0$, $\Rightarrow^+$, and $\Rightarrow^*$. The language generated by G is defined as $L(G) = \{w \in T^* : (S, q_0) \Rightarrow^* (w, q) \text{ for some } q \in Q\}$ and is said to be a *state language*. The family of all state languages is denoted by $\mathscr{L}_{ST}^{\varepsilon}$. In what follows, the superscript ε is omitted whenever the family of languages is generated by state grammars without erasing productions.

A *left-forbidding grammar* is a quadruple $G = (N, T, P, S)$, where N is the alphabet of nonterminals, T is the alphabet of terminals such that $N \cap T = \emptyset$, $V = N \cup T$ is the total alphabet, $S \in N$ is the start symbol, and P is a finite set of productions of the form $(A \to x, W)$, where $A \to x$ is a context-free production, $A \in N$ and $x \in V^*$, and $W \subseteq N$ is a finite set of nonterminals.

For any two strings $u, v \in V^*$ and a production $(A \to x, W) \in P$, we define the relation $uAv \Rightarrow uxv$ provided that

$$alph(u) \cap W = \emptyset.$$

In the standard manner, the relation $\Rightarrow$ is extended to $\Rightarrow^n$, for $n \geq 0$, $\Rightarrow^+$, and $\Rightarrow^*$. The language generated by G is defined as $L(G) = \{w \in T^* : S \Rightarrow^* w\}$ and is said to be a *left-forbidding language*. The family of all left-forbidding languages is denoted by $\mathscr{L}_{LF}^{\varepsilon}$. Again, the superscript ε is omitted whenever the family of languages is generated by left-forbidding grammars containing no erasing productions.

For any two strings $u, v \in V^*$, we define the relation *u terminally derives v* in G, written as $u \Rightarrow^t v$, provided that $u \Rightarrow^+ v$ in G and there is no $w \in V^*$ such that $v \Rightarrow w$. In addition, for $k \geq 1$, we define the relations $u \Rightarrow^{\leq k} v$, $u \Rightarrow^{=k} v$, and $u \Rightarrow^{\geq k} v$ in G provided that $u \Rightarrow^n v$ in G where $n \leq k$, $n = k$, and $n \geq k$, respectively.

A *left-forbidding cooperating distributed grammar system* is a construct $\Gamma = (N, T, P_1, P_2, \ldots, P_n, S)$, for some $n \geq 1$, where each component (defined as) $G_i = (N, T, P_i, S)$, for $i = 1, 2, \ldots, n$, is a left-forbidding grammar.

For any two strings $u, v \in V^*$ and an integer $1 \leq i \leq n$, let $u \Rightarrow_i v$ denote a derivation step made by the ith component of Γ, i.e., by the left-forbidding grammar G_i. In addition, let $u \Rightarrow_i^t v$ in Γ if $u \Rightarrow^t v$ in G_i. We say that Γ generates $w \in T^*$ in the terminal derivation mode (t-mode, for short) ($\leq k$-mode, $=k$-mode, $\geq k$-mode, for $k \geq 1$) provided that there exist $m \geq 1$ and $\alpha_i \in V^*$, for $i = 1, \ldots, m$, such that $\alpha_i \Rightarrow^t \alpha_{i+1}$ ($\alpha_i \Rightarrow^{\leq k} \alpha_{i+1}$, $\alpha_i \Rightarrow^{=k} \alpha_{i+1}$, $\alpha_i \Rightarrow^{\geq k} \alpha_{i+1}$, respectively) in H_i, where $H_i \in \{G_1, \ldots, G_n\}$ is a component of Γ, for $i = 1, \ldots, m-1$, $\alpha_1 = S$, and $\alpha_m = w$. Symbolically, $S \Rightarrow_\Gamma^t w$, $S \Rightarrow_\Gamma^{\leq k} w$, $S \Rightarrow_\Gamma^{=k} w$, $S \Rightarrow_\Gamma^{\geq k} w$, respectively. As usual, Γ is omitted whenever the meaning is clear. The language generated by Γ in the f-mode, for $f \in \{t\} \cup \{\leq k, =k, \geq k : k \geq 1\}$, is defined as $L(\Gamma, f) = \{w \in T^* : S \Rightarrow_\Gamma^f w\}$. The family of languages generated by left-forbidding cooperating distributed grammar systems with n components working in the f-mode is denoted by $\mathscr{L}_{LF}^{\varepsilon}(n, f)$. Again, the superscript ε is omitted whenever the components are nonerasing.

3. Results

First, it is not hard to see that $\mathscr{L}_{LF}^{\varepsilon}(1, f) = \mathscr{L}_{LF}^{\varepsilon}$ and $\mathscr{L}_{LF}(1, f) = \mathscr{L}_{LF}$, for any derivation mode $f \in \{t\} \cup \{\leq k, = k, \geq k : k \geq 1\}$.

The following theorem proves that these language families coincide with the family of context-free languages.

Theorem 1. $\mathscr{L}_{LF}^{\varepsilon} = \mathscr{L}_{CF}$.

Proof. As any context-free grammar is also a left-forbidding grammar, where empty sets are attached to each of its productions, the inclusion $\mathscr{L}_{LF}^{\varepsilon} \supseteq \mathscr{L}_{CF}$ holds.

To prove the other inclusion, $\mathscr{L}_{LF}^{\varepsilon} \subseteq \mathscr{L}_{CF}$, let $G = (N, T, P, S)$ be a left-forbidding grammar, and let $G' = (N, T, P', S)$ be a context-free grammar, where $P' = \{A \to x : (A \to x, W) \in P\}$. As any successful derivation of G is also a successful derivation of G', the inclusion $L(G) \subseteq L(G')$ holds.

On the other hand, let $w \in L(G')$ be a string successfully generated by the context-free grammar G'. Then, it is well-known that there exists a successful leftmost derivation of w in G'. Such a successful leftmost derivation is, however, also possible in G because the leftmost nonterminal can always be rewritten. Thus, the other inclusion $L(G') \subseteq L(G)$ holds as well, which completes the proof. □

As an immediate consequence of this theorem, we have that erasing productions can be eliminated from any left-forbidding grammar.

Corollary 2. $\mathscr{L}_{LF} = \mathscr{L}_{LF}^{\varepsilon} = \mathscr{L}_{CF}$.

In comparison with the previous result and the fact that cooperating distributed grammar systems with two context-free components working in the terminal derivation mode characterize the family of context-free languages (see [3]), the following result is of some interest.

Theorem 3. $\mathscr{L}^{\varepsilon}_{LF}(2, t) = \mathscr{L}_{RE}$.

Proof. On the one hand, it is not hard to prove (by standard techniques) that $\mathscr{L}^{\varepsilon}_{LF}(2, t) \subseteq \mathscr{L}_{RE}$.

On the other hand, let $L \in \mathscr{L}_{RE}$ be a recursively enumerable language. Then, as shown in [13], there is a state grammar $G = (N, T, Q, P, S, q_0)$ such that $L(G) = L$. Construct a left-forbidding cooperating distributed grammar system

$$\Gamma = (N_\Gamma, T, P_1, P_2, S')$$

with $N_\Gamma = N \cup N_1 \cup N_2 \cup \{S', \$, @, F\}$, where S', \$, @, and F are new symbols,

- $N_1 = \{[p, q, i] : p, q \in Q \text{ and } i \in \{1, 2\}\}$,
- $N_2 = \{\langle w\rangle : (X, p) \to (w, q) \in P\}$,

P_1 is constructed as follows:

1. for all $r \in Q$, add $(S' \to [q_0, r, 2]S\$, \emptyset)$ to P_1,
2. for all $r, s \in Q$, add $([r, s, 1] \to [r, s, 2], \emptyset)$ to P_1,
3. for all $(B, q) \to (w, h) \in P$, add $(B \to \langle w\rangle, W)$ to P_1, where

$$\begin{aligned} W = N_2 \cup \{@\} \quad &\cup \quad \{[r, s, 1] : r, s \in Q\} \\ &\cup \quad \{[r, s, 2] : r, s \in Q,\ r \neq q \text{ or } s \neq h\} \\ &\cup \quad \{X \in N : (X, q) \to (y, t) \in P\}, \end{aligned}$$

4. for all $r, s \in Q$, add $([r, s, 1] \to @, \emptyset)$ to P_1,
5. add $(\$ \to \varepsilon, N_\Gamma - \{@\})$ to P_1,
6. add $(\$ \to F, N_2)$ to P_1,

and P_2 is constructed as follows:

7. for all $p, q, r \in Q$, add $([p, q, 2] \to [q, r, 1], \emptyset)$ to P_2,
8. for all $\langle w\rangle \in N_2$ and $W = \{[p, q, 2] : p, q \in Q\}$, add $(\langle w\rangle \to w, W)$ to P_2,
9. add $(@ \to \varepsilon, \emptyset)$ to P_2.

Informally, Γ simulates a derivation of G so that it records the configuration of G in the first nonterminal of every sentential form except for the very last one. More precisely, this first nonterminal is of the form $[q, h, i]$, where q is the current state of G, h is a (guessed) state G moves to from q, and $i \in \{1, 2\}$ is an auxiliary symbol distinguishing between the two components of Γ. Then, production (1) starts the derivation, production (2) changes 1 to 2 in the configuration nonterminal to allow the application of production (3), which simulates the derivation step of G so that it verifies that

- no more than one simulation like this has been made (see the set N_2);
- $[r, s, 1]$ has been replaced with $[r, s, 2]$;
- $r = q$ and $s = h$ if simulating $(B, q) \to (w, h) \in P$;
- no nonterminal that can be rewritten occurs to the left of the rewritten symbol in the current sentential form.

Production (6) verifies that at least one production of G has been simulated; otherwise, as the terminal derivation mode is used, \$ has to be replaced with F, which blocks the derivation because F can never be replaced with a terminal string. A formal proof follows.

To prove that $L(G) \subseteq L(\Gamma)$, consider a derivation $(S, q_0) \Rightarrow^* (\alpha, q) \Rightarrow (\beta, h)$ in G. Let $\alpha = a_1a_2\dots a_n$ and $\beta = b_1b_2\dots b_m$, where for $i = 1,\dots,n$ and $j = 1,\dots,m$, $a_i, b_j \in V$. We prove that $[q,h,1]\alpha\$ \Rightarrow^+ [h,r,1]\beta\$$ in Γ, for any state $r \in Q$. Thus, assume that $(\alpha, q) \Rightarrow (\beta, h)$ by a production $(A,q) \to (w,h) \in P$, i.e., $a_i = A$, for some $1 \le i \le n$, and

$$(a_1 \dots a_{i-1}Aa_{i+1}\dots a_n, q) \Rightarrow (a_1 \dots a_{i-1}wa_{i+1}\dots a_n, h)$$

in G. Then,

$$\begin{aligned}
[q,h,1]a_1\dots a_{i-1}Aa_{i+1}\dots a_n\$ &\Rightarrow_1 [q,h,2]a_1\dots a_{i-1}Aa_{i+1}\dots a_n\$ \\
&\Rightarrow_1 [q,h,2]a_1\dots a_{i-1}\langle w\rangle a_{i+1}\dots a_n\$ \\
&\Rightarrow_2 [h,r,1]a_1\dots a_{i-1}\langle w\rangle a_{i+1}\dots a_n\$ \\
&\Rightarrow_2 [h,r,1]a_1\dots a_{i-1}wa_{i+1}\dots a_n\$
\end{aligned}$$

in Γ by productions constructed in (2), (3), (7), and (8), respectively, for any state $r \in Q$. Note that after the application of production (3), no production of P_1 is applicable. Of course, no further production of G can be simulated by production (3) because, considering the forbidding set W, there is no X to the left of A which can be replaced in G being in state q, and all symbols to the right of A which can be rewritten in G cannot be simulated in Γ (by productions constructed in (3)) because of the set N_2 in their forbidding sets. Then, after the application of production (8), no further production of P_2 is applicable. Thus, the derivation is a derivation in the terminal derivation mode. Furthermore, if $\alpha \in T^*$, then $[q,h,1]\alpha\$ \Rightarrow_1 @\alpha\$ \Rightarrow_1 @\alpha \Rightarrow_2 \alpha$ by productions constructed in (4), (5), and (9), respectively. Clearly, Γ simulates a derivation of G so that it starts by a production constructed in (1), i.e., $S' \Rightarrow_1 [q_0, r, 2]S\$$, for any state $r \in Q$. The derivation then proceeds as shown above. Hence, the inclusion holds.

On the other hand, to prove that $L(\Gamma) \subseteq L(G)$, consider a successful derivation of Γ. Such a derivation is of the form

$$S \Rightarrow_1^t \dots \Rightarrow_2^t x_0 \Rightarrow_1^t x_1 \Rightarrow_2^t x_2 \Rightarrow_1^t \dots \Rightarrow_2^t [r,s,1]w\$ \Rightarrow_1^t @w \Rightarrow_2^t w\,,$$

for some $w \in T^*$ and $r, s \in Q$. Consider a subderivation $x_0 \Rightarrow_1^t x_1 \Rightarrow_2^t x_2$. If $x_2 = w$, then the subderivation is the end of the derivation. Thus, assume that $x_2 \neq w$. Then, $x_0 = [p,q,1]a_1a_2\dots a_n\$$, for some $n \ge 1$, $a_i \in V$, for $i = 1,\dots,n$, and $p, q \in Q$. Assume that $alph(a_1a_2\dots a_n) \cap N \neq \emptyset$. Then, only productions constructed in (2) and (4) are applicable. Assume that a production constructed in (4) is applied, i.e., $[p,q,1]a_1a_2\dots a_n\$ \Rightarrow @a_1a_2\dots a_n\$$. As there is a nonterminal symbol in the sentential form (different from \$ and @), production (5) cannot be applied, and production (6) blocks the derivation. Therefore, only production (2) is applicable to x_0, i.e.,

$$[p,q,1]a_1a_2\dots a_n\$ \Rightarrow [p,q,2]a_1a_2\dots a_n\$\,.$$

Then, production (3) has to be applied, otherwise production (6) blocks the derivation, i.e.,

$$[p,q,2]a_1\dots A\dots a_n\$ \Rightarrow [p,q,2]a_1\dots\langle w\rangle\dots a_n\$\,,$$

for some $(A,p) \to (w,q) \in P$. In addition, it follows from the forbidding set of production (3) that there is no applicable production $(X,p) \to (y,r) \in P$ with X appearing to the left of the rewritten symbol A. As there is no applicable production in P_1, $x_1 = [p,q,2]a_1\dots\langle w\rangle\dots a_n\$$. Then, in P_2, only a production constructed in (7), followed by a production constructed in (8), is applicable, i.e.,

$$\begin{aligned}
[p,q,2]a_1\dots\langle w\rangle\dots a_n\$ &\Rightarrow [q,r,1]a_1\dots\langle w\rangle\dots a_n\$ \\
&\Rightarrow [q,r,1]a_1\dots w\dots a_n\$\,,
\end{aligned}$$

for any state $r \in Q$. The proof now proceeds by induction. As any derivation of Γ starts by a production constructed in (1), i.e., $S' \Rightarrow_1 [q_0, r, 2]S\$$, for any $r \in Q$, and then proceeds as proved above, the sequences of classes of productions applied during successful derivations form a regular language described by the following regular expression

$$1378(2378)^*459\,.$$

Hence, the theorem holds. □

To prove the following consequence, recall that it is known that $\mathscr{L}_{ST} = \mathscr{L}_{CS}$ (see [12]). Then, we have the following corollary.

Corollary 4. *For all $n \geq 2$, $\mathscr{L}_{LF}(n,t) = \mathscr{L}_{CS}$.*

Proof. $\mathscr{L}_{LF}(n,t) \subseteq \mathscr{L}_{CS}$ follows from the workspace theorem (see Theorem III.10.1 in [11]). To prove the other inclusion, let $L \subseteq T^*$ be a context sensitive language. Then, $L = L_1 \cup \bigcup_{a,b \in T}(a \cdot {}_aL_b \cdot b)$, where L_1 is a finite language, ${}_aL = \{a\} \backslash L$ is the left quotient of L with the singleton language $\{a\}$, and ${}_aL_b = {}_aL/\{b\}$ is the right quotient of ${}_aL$ with the singleton language $\{b\}$. As $\mathscr{L}_{CS}$ is obviously closed under right and left quotient with a singleton language, $\mathscr{L}_{LF}(n,t)$ is obviously closed under union, and by an obvious modification of productions (5) and (9) in the proof of Theorem 3, we can easily show that $a \cdot L' \cdot b \in \mathscr{L}_{LF}(n,t)$ for any context sensitive language $L' \subseteq T^*$ and symbols $a, b \in T$. $\square$

As a consequence of Theorem 3 and Corollary 4, we have the following results concerning the other derivation modes.

Theorem 5. *$\mathscr{L}^{\varepsilon}_{LF}(2,{=}2) = \mathscr{L}_{RE}$ and $\mathscr{L}_{LF}(2,{=}2) = \mathscr{L}_{CS}$.*

Proof. To prove this theorem, consider the proof of Theorem 3 and replace production (9) with two productions $(@ \to @', \emptyset)$ and $(@' \to \varepsilon, \emptyset)$, for some new symbol $@'$. $\square$

Corollary 6. *For all $n \geq 2$ and $f \in \{\geq k, =k : k \geq 2\}$, $\mathscr{L}^{\varepsilon}_{LF}(n,f) = \mathscr{L}_{RE}$ and $\mathscr{L}_{LF}(n,f) = \mathscr{L}_{CS}$.*

Proof. To prove this corollary, consider the proof of Theorem 3 and modify (analogously as in the proof of the previous theorem) the productions constructed in (1), (2), (4), and (7) to generate the same strings but in $k-1$ steps, production (9) to generate the same string in k steps, and, in addition, correspondingly modify the forbidding sets in productions constructed in (3) and (8). $\square$

Finally, the following lemma shows the power of the remaining derivation modes.

Theorem 7. *For $n \geq 2$ and $f \in \{=1, \geq 1\} \cup \{\leq k : k \geq 1\}$, $\mathscr{L}^{\varepsilon}_{LF}(n,f) = \mathscr{L}_{CF}$.*

Proof. Consider a left-forbidding cooperating distributed grammar system $\Gamma = (N, T, P_1, P_2, \ldots, P_n, S)$, and let $G = (N, T, P_1 \cup P_2 \cup \ldots \cup P_n, S)$ be a left-forbidding grammar. Clearly, any derivation of Γ is a derivation of G and vice versa. The proof now follows from Theorem 1. $\square$

Using the terminal derivation mode, the following result demonstrates that if erasing productions are allowed, the number of nonterminal symbols can be bounded.

Theorem 8. *Every recursively enumerable language is generated by a cooperating distributed grammar system working in the terminal derivation mode with two left-forbidding components and twelve nonterminals.*

Proof. Let L be a recursively enumerable language. By [14], there exists a grammar $G = (\{S, A, B, C, D\}, T, P \cup \{AB \to \varepsilon, CD \to \varepsilon\}, S)$ such that the set of productions P contains only context-free productions of these three forms: (i) $S \to uSa$, (ii) $S \to uSv$, (iii) $S \to \varepsilon$, where $u \in \{A, C\}^*$, $v \in \{B, D\}^*$, $a \in T$, and $L = L(G)$. In addition, any successful derivation of G is divided into two parts. The first part is of the form $S \Rightarrow^* w_1 S w_2 w \Rightarrow w_1 w_2 w$, made only by context-free productions from P, where $w_1 \in \{A, C\}^*$, $w_2 \in \{B, D\}^*$, and $w \in T^*$. The other part is of the form $w_1 w_2 w \Rightarrow^* w$, made only by productions $AB \to \varepsilon$ and $CD \to \varepsilon$.

Let $\Gamma = (N, T, P_1, P_2, S'')$ be a left-forbidding cooperating distributed grammar system, where

$$N = \{S'', S', S, A, A', B, B', C, C', D, D', \$\},$$

P_1 contains the following productions:

1. $(S'' \to S'\$, \emptyset)$,
2. $(S' \to S'u^R a, \emptyset)$ if $S \to uSa \in P$,

3. $(S' \to S, \emptyset)$,
4. $(S \to Su^Rv, \emptyset)$ if $S \to uSv \in P$,
5. $(S \to \varepsilon, \emptyset)$,
6. $(X' \to \varepsilon, \emptyset)$, for all $X \in \{A, B, C, D\}$,
7. $(\$ \to \varepsilon, \{A, B, C, D, A', B', C', D', S', S\})$,

and P_2 contains these productions:

8. $(A \to A', \{A, C, A', C'\})$,
9. $(C \to C', \{A, C, A', C'\})$,
10. $(B \to B', \{B, D, B', D'\})$,
11. $(D \to D', \{B, D, B', D'\})$,
12. $(\$ \to \$, \{B', C'\})$,
13. $(\$ \to \$, \{A', D'\})$.

To prove that $L(G) \subseteq L(\Gamma)$, consider a successful derivation of the form $S \Rightarrow^* w_1Sw_2w \Rightarrow w_1w_2w \Rightarrow^* w$ in G. Then, it is easy to see that $S'' \Rightarrow_1 S'\$ \Rightarrow_1^* S\omega\$ \Rightarrow_1 \omega\$$ in Γ, where $\omega \in \mathit{shuffle}(w_1^R, w_2w)$. Note that this is a terminal derivation of P_1 allowing to switch to P_2. Without loss of generality, we can assume that $w_1w_2w = w_1'ABw_2'w$, in which AB is rewritten by production $AB \to \varepsilon$ in G. Then, by productions (8) and (10), $\omega\$ \Rightarrow_2^t \omega'\$$, where $\omega' \in \mathit{shuffle}(A'w_1'^R, B'w_2'w)$. After that, $\omega'\$ \Rightarrow_1^t \omega''\$$, by production (6), where $\omega'' \in \mathit{shuffle}(w_1'^R, w_2'w)$ and $\mathit{alph}(\omega'') \cap N \neq \emptyset$. If $\mathit{alph}(\omega'') \cap N = \emptyset$, then also production (7) has to be applied, finishing the derivation. A completion of this part of the proof is simple and left to the reader. Thus, we have $S'' \Rightarrow_\Gamma^t w$.

On the other hand, to prove that $L(\Gamma) \subseteq L(G)$, note that the derivation of Γ starts as $S'' \Rightarrow_1^t \omega\$$ by productions (1) to (5), where $\omega \in \mathit{shuffle}(w_1, w_2w)$, $w_1 \in \{A, C\}^*$, $w_2 \in \{B, D\}^*$, and $w \in T^*$. Thus, it is easy to see that $S \Rightarrow^* w_1^RSw_2w \Rightarrow w_1^Rw_2w$ in G. By examining the form of w_1, we will see that $w_1 = h(w_2)$, for a homomorphism $h : \{B, D\}^* \to \{A, C\}^*$ defined as $h(B) = A$ and $h(D) = C$. To show this, assume that $w_1 = Aw_1'$, for some $w_1' \in \{A, C\}^*$. Then, as no production of P_1 is applicable, a production of P_2 is applied. Clearly, production (9) is not applicable. If production (11) is applied, which means that $w_2 = Dw_2'$, for some $w_2' \in \{B, D\}^*$, then production (10) is not applicable. However, production (12) is applicable, which blocks the derivation. Therefore, production (10) has to be applied. That is, $w_2 = Bw_2'$, for some $w_2' \in \{B, D\}^*$. As the derivation is terminal, production (8) is also applied. Now, neither production (12) nor (13) is applicable. Hence, $w_1 = Aw_1'$ implies that $w_2 = Bw_2'$, for some $w_2' \in \{B, D\}^*$. After that, the component P_2 is blocked, and P_1 proceeds the derivation, removing A' and B' by productions constructed in (6). Analogously, it can be proved that $w_1 = Cw_1'$, for some $w_1' \in \{A, C\}^*$, implies that $w_2 = Dw_2'$, for some $w_2' \in \{B, D\}^*$, and that if $w_1 = \varepsilon$, then also $w_2 = \varepsilon$. Thus, by induction, it follows that $w_1 = h(w_2)$. Moreover, if $w_1' = w_2' = \varepsilon$, the terminal derivation also removes \$ by production (7). Summarized, $w_1^Rw_2w = w_1'^RABw_2'w \Rightarrow w_1'^Rw_2'w \Rightarrow^* w$ in G by productions $AB \to \varepsilon$ and $CD \to \varepsilon$, i.e., $S \Rightarrow^* w$ in G. □

4. Conclusion

Let $n \geq 1$ be an integer, and let $\mathscr{L}_{CF}^{\varepsilon}(n, f)$ denote the family of languages generated by cooperating distributed grammar systems with n context-free components working in the f-mode, for $f \in \{t\} \cup \{\leq k, =k, \geq k : k \geq 1\}$. As usual, the superscript ε is omitted whenever the erasing productions are not allowed in context-free components. Recall that it is well-known (see [3]) that

- $\mathscr{L}_{CF} = \mathscr{L}_{CF}^{\varepsilon}(2, t) \subset \mathscr{L}_{CF}^{\varepsilon}(3, t) = \mathscr{L}_{CF}^{\varepsilon}(4, t) = \ldots = \mathscr{L}_{ET0L} \subset \mathscr{L}_{CS}$ and
- $\mathscr{L}_{CF} = \mathscr{L}_{CF}^{\varepsilon}(1, f) \subset \mathscr{L}_{CF}^{\varepsilon}(2, f) \subseteq \mathscr{L}_{CF}^{\varepsilon}(3, f) \subseteq \ldots \subseteq \mathscr{L}_{MAT}^{\varepsilon} \subset \mathscr{L}_{RE}$,

for all derivation modes $f \in \{=k, \geq k : k \geq 2\}$. The last proper inclusion is shown in [15]. Note also that the first item holds unchanged if ε is removed, while, in the second item, $\mathscr{L}_{MAT}^{\varepsilon} \subset \mathscr{L}_{RE}$ can be replaced with $\mathscr{L}_{MAT} \subset \mathscr{L}_{CS}$. Thus, surprisingly, although $\mathscr{L}_{CF} = \mathscr{L}_{LF}^{\varepsilon}$, we have shown that, for all $n \geq 1$, $\mathscr{L}_{CF}^{\varepsilon}(n, t) \subset \mathscr{L}_{LF}^{\varepsilon}(2, t)$.

Analogously, for all $n \geq 3$, $m \geq 2$, and $f \in \{=k, \geq k : k \geq 2\}$,

- $\mathscr{L}_{LF}^{\varepsilon}(1, t) = \mathscr{L}_{CF}^{\varepsilon}(2, t) \subset \mathscr{L}_{CF}^{\varepsilon}(n, t) \subset \mathscr{L}_{LF}^{\varepsilon}(2, t) = \mathscr{L}_{RE}$,

- $\mathscr{L}_{LF}(1,t) = \mathscr{L}_{CF}(2,t) \subset \mathscr{L}_{CF}(n,t) \subset \mathscr{L}_{LF}(2,t) = \mathscr{L}_{CS}$,
- $\mathscr{L}^{\varepsilon}_{LF}(1,f) = \mathscr{L}^{\varepsilon}_{CF}(1,f) \subset \mathscr{L}^{\varepsilon}_{CF}(m,f) \subset \mathscr{L}^{\varepsilon}_{LF}(2,f) = \mathscr{L}_{RE}$,
- $\mathscr{L}_{LF}(1,f) = \mathscr{L}_{CF}(1,f) \subset \mathscr{L}_{CF}(m,f) \subset \mathscr{L}_{LF}(2,f) = \mathscr{L}_{CS}$.

Finally, for all $n \geq 2$, we have a new characterization of the Chomsky hierarchy $\mathscr{L}_{CF} \subset \mathscr{L}_{CS} \subset \mathscr{L}_{RE}$ in terms of left-forbidding cooperating distributed grammar systems: $\mathscr{L}_{LF}(1,f) = \mathscr{L}^{\varepsilon}_{LF}(1,f) \subset \mathscr{L}_{LF}(n,f) \subset \mathscr{L}^{\varepsilon}_{LF}(n,f)$, for all $f \in \{t\} \cup \{=k, \geq k : k \geq 2\}$.

Acknowledgments

The authors gratefully acknowledge very useful suggestions and comments of the anonymous referees.

This work was supported by the Czech Ministry of Education Research Plan No. MSM 0021630528, by the Czech Grant Agency project No. 201/07/0005, and by the Czech Academy of Sciences Institutional Research Plan No. AV0Z10190503.

References

[1] E. Csuhaj-Varjú, J. Dassow, On cooperating/distributed grammar systems, Journal of Information Processing and Cybernetics (EIK) 26 (1-2) (1990) 49–63.
[2] E. Csuhaj-Varjú, J. Dassow, J. Kelemen, G. Păun, Grammar Systems: A Grammatical Approach to Distribution and Cooperation, Gordon and Breach Science Publishers, Topics in Computer Mathematics 5, Yverdon, 1994.
[3] J. Dassow, G. Păun, G. Rozenberg, Grammar systems, in: Handbook of formal languages, Volume 2, Springer, Berlin, 1997, pp. 155–213.
[4] A. P. J. van der Walt, Random context languages, in: IFIP Congress (1), 1971, pp. 66–68.
[5] A. P. J. van der Walt, S. Ewert, A shrinking lemma for random forbidding context languages, Theoretical Computer Science 237 (1-2) (2000) 149–158.
[6] S. Ewert, A. P. J. van der Walt, A pumping lemma for random permitting context languages, Theoretical Computer Science 270 (1-2) (2002) 959–967.
[7] T. Masopust, On the terminating derivation mode in cooperating distributed grammar systems with forbidding components, International Journal of Foundations of Computer Science 20 (2) (2009) 331–340.
[8] E. Csuhaj-Varjú, T. Masopust, G. Vaszil, Cooperating distributed grammar systems with permitting grammars as components, Romanian Journal of Information Science and Technology 12 (2) (2009) 175–189.
[9] Z. Křivka, T. Masopust, A note on the cooperation in rewriting systems with context-dependency checking, in: 11th Italian Conference on Theoretical Computer Science, 2009, pp. 129–135.
[10] Z. Křivka, T. Masopust, Cooperating distributed grammar systems with random context grammars as components, manuscript.
[11] A. Salomaa, Formal languages, Academic Press, New York, 1973.
[12] T. Kasai, An hierarchy between context-free and context-sensitive languages, Journal of Computer and System Sciences 4 (5) (1970) 492–508.
[13] A. Meduna, G. Horvath, On state grammars, Acta Cybernetica 8 (3) (1988) 237–245.
[14] V. Geffert, Context-free-like forms for the phrase-structure grammars, in: M. Chytil, L. Janiga, V. Koubek (Eds.), MFCS, Vol. 324 of Lecture Notes in Computer Science, Springer, 1988, pp. 309–317.
[15] D. Hauschildt, M. Jantzen, Petri net algorithms in the theory of matrix grammars, Acta Informatica 31 (8) (1994) 719–728.