International Journal of Foundations of Computer Science
© World Scientific Publishing Company

On the Terminating Derivation Mode in Cooperating Distributed Grammar Systems with Forbidding Components

Tomáš Masopust

Faculty of Information Technology, Brno University of Technology
Božetěchova 2, Brno 61266, Czech Republic
masopust@fit.vutbr.cz

Received (Day Month Year)
Accepted (Day Month Year)
Communicated by (xxxxxxxxxx)

This paper discusses the terminating derivation mode in cooperating distributed grammar systems where components are forbidding grammars instead of context-free grammars. Such systems are called forbidding cooperating distributed grammar systems, and it is demonstrated that the number of their components can be reduced to two without changing the generative power and that these systems are computationally complete. Without erasing productions, however, these systems are less powerful than context-sensitive grammars.

Keywords: Cooperating distributed grammar systems; terminating derivation mode; forbidding grammars; forbidding cooperating distributed grammar systems; generative power.

1991 Mathematics Subject Classification: 68Q42 and 68Q45

1. Introduction

In 1970, van der Walt [8] introduced and studied forbidding grammars as a special case of random context grammars with appearance checking, where all permitting sets are empty. Specifically, forbidding grammars are context-free grammars where a finite set of nonterminals (called a forbidding set) is associated to each production. Such a production is then applicable if no symbol from the associated forbidding set occurs in the current sentential form. It is well-known (see [1,6]) that the family of languages generated by forbidding grammars is properly included in the family of recursively enumerable (even recursive) languages and that the family of languages generated by λ-free forbidding grammars is properly included in the family of languages generated by λ-free programmed grammars with appearance checking (see [5]), which is properly included in the family of context-sensitive languages. Finally, note that the question of whether λ-productions are important for the generative power of forbidding grammars is a longstanding open problem in the theory of regulated rewriting.

To get an alternative insight into this problem, this paper introduces and studies forbidding cooperating distributed (CD) grammar systems with and without λ-productions. These systems consist of several cooperating components represented by forbidding gram-

mars that work in some prescribed derivation mode (see [2]). Specifically, in the terminating derivation mode (t-mode), each component makes derivation steps as long as it can, and if it cannot make a derivation step, another component works in the same way.

Although the problem of λ-productions is unsolved in case of forbidding grammars, this paper demonstrates that λ-productions play an important role in forbidding CD grammar systems. Specifically, with them, the family of languages generated by forbidding CD grammar systems coincides with the family of recursively enumerable languages, while without them, this family coincides with the family of languages generated by programmed grammars (random context grammars) with appearance checking, which is properly included in the family of context-sensitive languages. In addition, it is also well-known that forbidding grammars are not as powerful as random context grammars with appearance checking. Therefore, permitting sets (sets of nonterminals that have to appear in the current sentential form so that the productions are applicable) are necessary for them to obtain their full generative power. However, this paper demonstrates that forbidding CD grammar systems (with only two components) are able to compensate the absence of permitting sets.

Finally, in [4], the generative power of a modification of forbidding CD grammar systems considering only occurrences of nonterminals to the left of the rewritten nonterminal (so-called left-forbidding CD grammar systems) has been studied, and although the family of languages generated by them without λ-productions is properly included in the family of languages generated by them with λ-productions, the families of languages generated by left-forbidding grammars with and without λ-productions both coincide with the family of context-free languages. In fact, this is surprising in comparison with the common context-free CD grammar systems, where λ-productions have no impact on the generated language family. Specifically, it is well-known (see [2]) that regardless of λ-productions, the family of languages generated by context-free CD grammar systems (considering the t-mode) coincides either with the family of context-free languages (if they have no more than two components), or with the family of ET0L languages (if they have three or more components).

2. Preliminaries

This paper assumes that the reader is familiar with formal language theory (see [7]). For a set A, $|A|$ denotes the cardinality of A. Let $\subseteq$ and $\subset$ denote the inclusion and the proper inclusion, respectively. For an alphabet (finite nonempty set) V, V^* represents the free monoid generated by V. The unit of V^* is denoted by λ. Set $V^+ = V^* \setminus \{\lambda\}$. For $w \in V^*$, $|w|$ denotes the length of w, and $alph(w)$ denotes the set of all symbols occurring in w. Let $\mathscr{L}(CF)$, $\mathscr{L}(ET0L)$, $\mathscr{L}(CS)$, $\mathscr{L}(REC)$, and $\mathscr{L}(RE)$ denote the families of context-free, ET0L, context-sensitive, recursive, and recursively enumerable languages, respectively.

A *forbidding grammar* (see [8]) is a quadruple $G = (N,T,P,S)$, where N is the alphabet of nonterminals, T is the alphabet of terminals such that $N \cap T = \emptyset$, $V = N \cup T$, $S \in N$ is the start symbol, and P is a finite set of productions of the form $(A \to x, W)$, where $A \in N$, $x \in V^*$, and $W \subseteq N$. As usual, $(A \to x, W) \in P$ is said to be a λ-production if

$x = \lambda$. For $u, v \in V^*$ and $(A \to x, W) \in P$, $uAv \Rightarrow uxv$ provided that $alph(uAv) \cap W = \emptyset$.[a] Extend $\Rightarrow$ to $\Rightarrow^n$, for $n \geq 0$, $\Rightarrow^+$, and $\Rightarrow^*$. The language generated by G is defined as $L(G) = \{w \in T^* : S \Rightarrow^* w\}$. The family of languages generated by forbidding grammars is denoted by $\mathscr{L}(F)$, or $\mathscr{L}(F - \lambda)$ if they are λ-free, i.e., they do not contain any λ-productions.

A *left-forbidding grammar* (see [4]) is a quadruple $G = (N, T, P, S)$, where N, T, V, P, and S are defined as in a forbidding grammar. For $u, v \in V^*$ and $(A \to x, W) \in P$, $uAv \Rightarrow uxv$ provided that $alph(u) \cap W = \emptyset$. Note that in comparison with the previous definition, only the left context of A, u, is considered. Extend $\Rightarrow$ to $\Rightarrow^n$, for $n \geq 0$, $\Rightarrow^+$, and $\Rightarrow^*$. The language generated by G is defined as $L(G) = \{w \in T^* : S \Rightarrow^* w\}$. The family of languages generated by left-forbidding grammars is denoted by $\mathscr{L}(LF)$, or $\mathscr{L}(LF - \lambda)$ if they are λ-free.

A *programmed grammar with appearance checking* (introduced and studied in [5]) is a quadruple $G = (N, T, P, S)$, where N is the alphabet of nonterminals, T is the alphabet of terminals such that $N \cap T = \emptyset$, $V = N \cup T$, $S \in N$ is the start symbol, and P is a finite set of productions of the form $(r.A \to v, \sigma(r), \varphi(r))$, where r is a label of the context-free production $A \to v$, i.e., $A \in N$ and $v \in V^*$, and $\sigma(r), \varphi(r) \subseteq lab(P)$ are *success* and *failure fields*, respectively, where $lab(P) = \{r : (r.A \to v, \sigma(r), \varphi(r)) \in P\}$ is the set of all production labels. For $(x, q), (y, r) \in V^* \times lab(P)$, $(x, q) \Rightarrow (y, r)$ provided that $(q.A \to u, \sigma(q), \varphi(q)) \in P$ and

(1) either $x = x_1 A x_2$, $y = x_1 u x_2$, and $r \in \sigma(q)$, for some $x_1, x_2 \in V^*$,
(2) or $A \notin alph(x)$, $y = x$, and $r \in \varphi(q)$.

Extend $\Rightarrow$ to $\Rightarrow^n$, for $n \geq 0$, $\Rightarrow^+$, and $\Rightarrow^*$. The language generated by G is defined as $L(G) = \{w \in T^* : (S, q) \Rightarrow^* (w, r), \text{ for some } q, r \in lab(P)\}$. The family of languages generated by programmed grammars with appearance checking is denoted by $\mathscr{L}(P, ac)$, or $\mathscr{L}(P - \lambda, ac)$ if they are λ-free.

Let $n \geq 2$ be an integer. A *cooperating distributed* (CD) *grammar system* (see [2]) is a construct $\Gamma = (N, T, P_1, P_2, \ldots, P_n, S)$, where for $i = 1, \ldots, n$, each component (defined as) $G_i = (N, T, P_i, S)$ is a context-free grammar. We say that Γ is λ-free if all its components are λ-free. For $u, v \in V^*$ and $1 \leq k \leq n$, let $u \Rightarrow_k v$ denote a derivation step made by G_k. Extend $\Rightarrow_k$ to $\Rightarrow_k^n$, for $n \geq 0$, $\Rightarrow_k^+$, and $\Rightarrow_k^*$. In addition, we define the relation *u terminally derives v in* G_k, written as $u \Rightarrow_k^t v$, if $u \Rightarrow_k^+ v$ and there is no $w \in V^*$ such that $v \Rightarrow_k w$. The language generated by Γ in the terminating derivation mode (t-mode, for short) is defined as $L(\Gamma) = \{w \in T^* : \text{ there exists } \ell \geq 1 \text{ such that } \alpha_i \Rightarrow_{k_i}^t \alpha_{i+1}, 1 \leq k_i \leq n, \text{ for } i = 1, \ldots, \ell - 1, \alpha_1 = S, \text{ and } \alpha_\ell = w\}$. The family of languages generated by CD grammar systems with n components is denoted by $\mathscr{L}(CD, CF, n)$, or $\mathscr{L}(CD, CF - \lambda, n)$ if they are

[a] In the literature, the derivation is also defined as follows: $uAv \Rightarrow' uxv$ if $(A \to x, W) \in P$ and $alph(uv) \cap W = \emptyset$. However, these two definitions are equivalent: ($\Rightarrow$) Just remove all productions $(A \to x, W) \in P$ with $A \in W$ because they are not applicable in the definition from the paper. ($\Leftarrow$) Let $N' = \{A' : A \in N\}$, $N \cap N' = \emptyset$, and $G = (N \cup N', T, P', S)$ with $P' = \{(A \to A', N'), (A' \to x, W) : (A \to x, W) \in P\}$. Then, $uAv \Rightarrow' uxv$ if and only if $uAv \Rightarrow uA'v \Rightarrow uxv$.

λ-free. It is well-known (see [2]) that $\mathscr{L}(CD,CF,2) = \mathscr{L}(CD,CF-\lambda,2) = \mathscr{L}(CF)$ and that for $n \geq 3$, $\mathscr{L}(CD,CF,n) = \mathscr{L}(CD,CF-\lambda,n) = \mathscr{L}(ET0L)$.

Let $n \geq 2$. A *left-forbidding cooperating distributed grammar system* (see [4]) is a construct $\Gamma = (N,T,P_1,P_2,\ldots,P_n,S)$, where for $i = 1,\ldots,n$, each component $G_i = (N,T,P_i,S)$ is a left-forbidding grammar. The language generated by Γ in the t-mode is defined in the same way as in the definition of CD grammar systems. The family of languages generated by left-forbidding CD grammar systems with n components is denoted by $\mathscr{L}(CD,LF,n)$, or $\mathscr{L}(CD,LF-\lambda,n)$ if they are λ-free. It is shown in [4] that for $n \geq 2$, $\mathscr{L}(CD,LF,n) = \mathscr{L}(CD,LF,2)$ and $\mathscr{L}(CD,LF-\lambda,n) = \mathscr{L}(CD,LF-\lambda,2)$. Thus, define $\mathscr{L}(CD,LF) = \mathscr{L}(CD,LF,2)$ and $\mathscr{L}(CD,LF-\lambda) = \mathscr{L}(CD,LF-\lambda,2)$.

3. Definition

Let $n \geq 2$. A *forbidding cooperating distributed grammar system* is a construct

$$\Gamma = (N,T,P_1,P_2,\ldots,P_n,S),$$

where for $i = 1,\ldots,n$, each component $G_i = (N,T,P_i,S)$ is a forbidding grammar. The language generated by Γ in the t-mode is defined in the same way as in the definition of CD grammar systems. The family of languages generated by forbidding CD grammar systems with n components is denoted by $\mathscr{L}(CD,F,n)$, or $\mathscr{L}(CD,F-\lambda,n)$ if they are λ-free.

4. Results

This section presents the main results of this paper. First, it demonstrates that the number of components in forbidding CD grammar systems can be reduced to two without changing the generative power. Then, it describes the generative power of these systems with respect to whether λ-productions are allowed or not.

Theorem 1. *$\mathscr{L}(CD,F-\lambda,n) = \mathscr{L}(CD,F-\lambda,2)$ and $\mathscr{L}(CD,F,n) = \mathscr{L}(CD,F,2)$, $n \geq 3$.*

Proof. Let $n \geq 3$ and let $\Gamma = (N,T,P_1,P_2,\ldots,P_n,S)$ be a forbidding CD grammar system. Construct $\Gamma' = (N',T,P_1',P_2',S')$ so that $N' = N_{[]} \cup N_{\langle\rangle}$, where $N_{[]} = \{[A,i] : A \in N, 1 \leq i \leq n\}$, $N_{\langle\rangle} = \{\langle A,i\rangle : A \in N, 1 \leq i \leq n\}$, P_1' contains

(1) $(S' \to [S,i],\emptyset)$, $1 \leq i \leq n$,
(2) $(\langle A,i\rangle \to [B,j],\emptyset)$, for $A,B \in N$, $1 \leq i,j \leq n$,

and P_2' contains

(3) $([B,j] \to \langle B,j\rangle,\emptyset)$,
(4) $(\langle B,j\rangle \to h_j(v), N_{[]} \cup h_j(W) \cup \bigcup_{k \neq j} h_k(N))$, for $(B \to v,W) \in P_j$,

where h_j, $1 \leq j \leq n$, is a homomorphism such that $h_j(a) = a$, $a \in T$, and $h_j(A) = \langle A,j\rangle$, $A \in N$.

The basic idea is that P_1' chooses a component of Γ to simulate, say i, and P_2' simulates a terminating derivation of this component.

To prove that $L(\Gamma) \subseteq L(\Gamma')$, let

$$\begin{aligned} x_1A_1x_2A_2\dots x_\ell A_\ell\dots x_uA_ux_{u+1} &\Rightarrow_i x_1A_1x_2A_2\dots x_\ell v\dots x_uA_ux_{u+1} \\ &\Rightarrow_i^t z_1B_1z_2B_2\dots z_{u'}B_{u'}z_{u'+1} \end{aligned}$$

for some $u,u' \geq 1$, where $x_j,z_k \in T^*$, $A_j,B_k \in N$, $(A \to v,W) \in P_i$, and $A_\ell = A$, for some $1 \leq \ell \leq u$. Assume that the component applied next in Γ is G_j. Then, by a sequence of (sets of) productions $(3)^u(4)^+(2)^{u'}$, we have

$$\begin{aligned} & x_1[A_1,i]x_2[A_2,i]\dots x_\ell[A_\ell,i]\dots x_u[A_u,i]x_{u+1} \\ \Rightarrow_2^u\ & x_1\langle A_1,i\rangle x_2\langle A_2,i\rangle\dots x_\ell\langle A_\ell,i\rangle\dots x_u\langle A_u,i\rangle x_{u+1} \\ \Rightarrow_2\ & x_1\langle A_1,i\rangle x_2\langle A_2,i\rangle\dots x_\ell h_i(v)\dots x_u\langle A_u,i\rangle x_{u+1} \\ & \vdots \\ \Rightarrow_2^t\ & z_1\langle B_1,i\rangle z_2\langle B_2,i\rangle\dots z_{u'}\langle B_{u'},i\rangle z_{u'+1} \\ \Rightarrow_1^{u'}\ & z_1[B_1,j]z_2[B_2,j]\dots z_{u'}[B_{u'},j]z_{u'+1} \end{aligned}$$

in Γ'. According to the t-mode, productions constructed in (4) are applied while there is an applicable production of Γ to be simulated. Then, productions constructed in (2) are applied, keeping the second parts of nonterminals equal, i.e., j. As $S' \to [S,i]$, $1 \leq i \leq n$, and $[S,i]$ is of the form described above, the proof proceeds by induction.

On the other hand, to prove that $L(\Gamma') \subseteq L(\Gamma)$, let $x_1[A_1,i_1]x_2[A_2,i_2]\dots x_u[A_u,i_u]x_{u+1}$ be a sentential form of Γ', where $x_i \in T^*$, $i = 1,\dots,u+1$, $A_j \in N$, and $1 \leq i_j \leq n$, $j = 1,\dots,u$. We prove that $i_1 = i_2 = \dots = i_u$ and that Γ' correctly simulates a terminating derivation of Γ.

First, note that productions constructed in (4) are not applicable because there are nonterminals from $N_{[]}$ in the sentential form. Therefore, only productions constructed in (3) are applicable, i.e.,

$$x_1[A_1,i_1]x_2[A_2,i_2]\dots x_u[A_u,i_u]x_{u+1} \Rightarrow^u x_1\langle A_1,i_1\rangle x_2\langle A_2,i_2\rangle\dots x_u\langle A_u,i_u\rangle x_{u+1}\,.$$

Now, productions constructed in (4) are applicable only if $i_1 = i_2 = \dots = i_u$. (If there are k,l such that $i_k \neq i_l$, the derivation continues by productions from P_1' rewriting these nonterminals by productions constructed in (2) changing the second parts of nonterminals.) Thus, assume that they are equal. Then, productions constructed in (4) simulate the productions from P_{i_1}, i.e.,

$$\begin{aligned} & x_1\langle A_1,i_1\rangle x_2\langle A_2,i_1\rangle\dots x_u\langle A_u,i_1\rangle x_{u+1} \\ \Rightarrow_2^t\ & x_1'\langle A_1',i_1\rangle x_2'\langle A_2',i_1\rangle\dots x_{u'}'\langle A_{u'}',i_1\rangle x_{u'+1}\,, \end{aligned}$$

for some $u' \geq 1$, where $x_i' \in T^*$, $i = 1,\dots,u'+1$, and $A_j' \in N$, $j = 1,\dots,u'$. Then, only productions from P_1' constructed in (2) are applicable. Therefore,

$$\begin{aligned} & x_1'\langle A_1',i_1\rangle x_2'\langle A_2',i_1\rangle\dots x_{u'}'\langle A_{u'}',i_1\rangle x_{u'+1} \\ \Rightarrow_1^t\ & x_1'[A_1',j_1]x_2'[A_2',j_2]\dots x_{u'}'[A_{u'}',j_{u'}]x_{u'+1}\,, \end{aligned}$$

$1 \leq j_k \leq n$, $k = 1,\dots,u'$.

In Γ, $x_1A_1x_2A_2\dots x_nA_nx_{n+1} \Rightarrow^t x'_1A'_1x'_2A'_2\dots x'_nA'_nx_{n'+1}$ by the corresponding productions from P_{i_1}. As any derivation starts by a production constructed in (1) of the form $S' \to [S,i]$, $1 \leq i \leq n$, and $[S,i]$ is of the form considered above, the proof proceeds by induction. □

As the number of components has no effect on the generative power of forbidding CD grammar systems, we establish the following definitions.

Definition 2. *Define* $\mathscr{L}(CD,F) = \mathscr{L}(CD,F,2)$ *and* $\mathscr{L}(CD,F-\lambda) = \mathscr{L}(CD,F-\lambda,2)$.

To prove the other results, note that it is well-known that $\mathscr{L}(P,ac) = \mathscr{L}(RE)$ (see [5]), and it is not hard to construct (by standard techniques) a Turing machine accepting a language from $\mathscr{L}(CD,F)$. Therefore, the following lemma is obvious.

Lemma 3. $\mathscr{L}(CD,F) \subseteq \mathscr{L}(P,ac)$.

Analogously, it is not hard to show that $\mathscr{L}(CD,F-\lambda) \subseteq \mathscr{L}(CS)$. Moreover, the following lemma demonstrates that this inclusion is proper, which is surprising in comparison with λ-free left-forbidding CD grammar systems that generate the whole family of context-sensitive languages (cf. [4]).

Lemma 4. $\mathscr{L}(CD,F-\lambda) \subseteq \mathscr{L}(P-\lambda,ac)$.

Proof. Let $L \in \mathscr{L}(CD,F-\lambda)$ and let $\Gamma = (N,T,P_1,P_2,S)$ be a λ-free forbidding CD grammar system such that $L(\Gamma) = L$. Let $G = (N,T,P,S')$ be a programmed grammar with appearance checking, where P is constructed, starting from an empty set, as follows. Assume that all productions of Γ are labeled by different labels, i.e., for $z \in \{1,2\}$, set $m_z = |P_z|$ and $lab(P_z) = \{z'_1,z'_2,\dots,z'_{m_z}\}$. Set $\ell(z) = \{\ell : (\ell.A \to x,\emptyset) \in P_z\}$.

(1) For $(\ell.A \to x,W) \in P_z$, $W = \{X_1,X_2,\dots,X_s\}$, $s \geq 1$,
- add $(p_i.X_i \to X_i,\emptyset,\{p_{i+1}\})$ to P, p_i is a new label, for $i = 1,\dots,s$, $p_{s+1} = \ell$,
- add $(\ell.A \to x,\{c_z\},\emptyset)$ to P, where c_z is a special symbol defined in 3(ii) below,
- set $\ell(z) = \ell(z) \cup \{p_1\}$.

(2) Add $(s.S' \to S,\ell(1) \cup \ell(2),\emptyset)$ to P, where s is a new label.

(3) Let $M_z = \{z_i : (z'_i.A_i \to x_i,W_i) \in P_z\}$ be a set of new labels and assume that $W_i = \{X_{i,1},X_{i,2},\dots,X_{i,s_i}\}$.
- For $i = 1,\dots,m_z$ and $j = 1,\dots,s_i$,
  (i) let $z_{i,j}$ be new labels,
  (ii) set $c_z = z_{1,1}$, where $z_{1,1} = z_1$ if $s_1 = 0$ (i.e., $W_1 = \emptyset$),
  (iii) add $(z_{i,j}.X_{i,j} \to X_{i,j},\{z_{i+1,1}\},\{z_{i,j+1}\})$ to P, $z_{i,s_i+1} = z_i$, $z_{m_z+1,1} = \ell(r)$, $r \neq z$, and if $s_i = 0$, then $z_{i,1} = z_i$.
- For $i = 1,\dots,m_z-1$, add $(z_i.A_i \to A_i,\ell(z),\{z_{i+1,1}\})$ to P.
- Add $(z_{m_z}.A_{m_z} \to A_{m_z},\ell(z),\ell(r))$, $r \neq z$.

To prove that $L(G) \subseteq L(\Gamma)$, consider the sequences of productions constructed in (1), (2), and (3). Clearly, the production constructed in (2) starts the derivation, and after that it is never applied again. Then, a production from $\ell(z)$, $z \in \{1,2\}$ is applied, i.e., either one of the productions with empty forbidding sets, or the first production of a sequence of productions constructed in (1). Such a sequence of productions is of the form $p_1 p_2 \dots p_s \ell$, where $(\ell.A \to x, \{X_1, \dots, X_s\}) \in P_z$, and it verifies that there is no symbol from $\{X_1, \dots, X_s\}$ in the current sentential form. If there is such a symbol in the current sentential form, say X_i, then the derivation is blocked because the success field of production p_i is empty and the derivation cannot continue (to replace X_i with a terminal string). Thus, assume that there is no such symbol in the current sentential form. Then, ℓ is applied, i.e., A is replaced with x. As there is no symbol from $\{X_1, \dots, X_s\}$ in the sentential form, Γ applies ℓ, too.

The derivation in G now continues by production c_z, which is the first production of one of the two sequences of productions constructed in (3). By this sequence, the grammar looks for an applicable production so that for $(A \to x, W) \in P_z$, G tries to replace all symbols from W by themselves one by one. If it fails for all these symbols, then it tries to replace A with itself. If it succeeds, the production is applicable; otherwise, the production is not applicable. If it finds an applicable production, the derivation continues in the simulation of a production from P_z (see (1)); otherwise, there is no applicable production in P_z, and the grammar starts to simulate a production from the other component. The proof then proceeds by induction.

The inclusion $L(\Gamma) \subseteq L(G)$ is proved analogously. □

Lemma 5. *$L(G) \in \mathscr{L}(CD,F)$ for any programmed grammar G with appearance checking.*

Proof. Let $G = (N, T, P, S)$ be a programmed grammar with appearance checking and let $n = |P|$. Without loss of generality, assume that $lab(P) = \{1, \dots, n\}$. Let

$$\Gamma = (N', T, P_0, P_{1,1}, P_{1,2}, P_{1,3}, P_{2,1}, P_{2,2}, P_{2,3}, \dots, P_{n,1}, P_{n,2}, P_{n,3}, S')$$

be a forbidding CD grammar system defined as follows.

$$\begin{aligned} N' = N \cup \{S'\} \cup \{&[X,a], [X,i,a] : X \subseteq lab(P), 1 \leq i \leq n, a \in V \cup \{\lambda\}\} \\ &\cup \{\langle u \rangle, \langle X, u \rangle : (i.A \to u, \sigma(i), \varphi(i)) \in P, X \subseteq lab(P)\}, \end{aligned}$$

P_0 contains

(1) $(S' \to [Z,S], \emptyset)$, where $Z = \{i \in lab(P) : (i.S \to v, \sigma(i), \varphi(i)) \in P\}$,
(2) $(\langle u \rangle \to u, \emptyset)$, for $\langle u \rangle \in N'$,
(3) $(\langle X, a_1 a_2 \dots a_k \rangle \to [X,a_1] a_2 \dots a_k, \emptyset)$, for $\langle X, a_1 \dots a_k \rangle \in N'$, where $a_i \in V$, $i = 1, \dots, k$,
(4) $(\langle X, \lambda \rangle \to [X,\lambda], \emptyset)$, for $\langle X, \lambda \rangle \in N'$,
(5) $([X,i,a] \to [X,a], \emptyset)$, for $[X,i,a] \in N'$, and
(6) $([X,a] \to a, N' \setminus \{[X,a]\})$, for $[X,a] \in N'$, $a \in T \cup \{\lambda\}$.

For $(i.A \to u, \sigma(i), \varphi(i)) \in P$, $i = 1, \dots, n$, set

$$P_{i,1} = \{([Z,a] \to [\varphi(i), a], \{A, [Z,A]\}) : Z \subseteq lab(P), a \in V \cup \{\lambda\}, i \in Z\},$$

$P_{i,2}$ contains

(7) $([Z,a] \to [Z,a], \emptyset)$, for $[Z,a] \in N'$, $i \notin Z$,
(8) $([Z,a] \to [\sigma(i),i,a], \emptyset)$, for $[Z,a] \in N'$, $i \in Z$,
(9) $(A \to \langle u \rangle, \{\langle v \rangle, \langle Z,v \rangle, [Z,a] \in N' : v \in V^*, Z \subseteq lab(P), a \in V \cup \{\lambda\}\})$,
(10) $([\sigma(i),i,a] \to [\sigma(i),i,a], \{\langle u \rangle\})$,
(11) $([X,j,a] \to [X,j,a], \emptyset)$, for $[X,j,a] \in N'$, $j \neq i$,

and

$$P_{i,3} = \{([Z,A] \to \langle \sigma(i),u \rangle, \emptyset) : Z \subseteq lab(P), i \in Z\}.$$

The first nonterminal of each sentential form, $[X,a]$ or $[X,i,a]$, contains the set of labels of all productions that are applicable in the current sentential form in G. Each of the sets $P_{i,1}$ simulates an application of $(i.A \to u, \sigma(i), \varphi(i)) \in P$ if there is no A in the current sentential form. Therefore, the derivation continues by a production from $\varphi(i)$. On the other hand, each of the sets $P_{i,2}$ and $P_{i,3}$ simulates an application of i if there is A in the current sentential form.

The system simulates an application of i in $P_{i,2}$ so that if $i \in Z$, then $[Z,a]$ is replaced with $[\sigma(i),i,a]$. (If $i \notin Z$, then it replaces $[Z,a]$ with itself for ever, see (7).) Now, $A \to u$ has to be simulated by replacing A with $\langle u \rangle$; otherwise, it replaces $[\sigma(i),i,a]$ with itself for ever, see (10). Thus, assume that $A \to \langle u \rangle$ has been applied. Then, the current component is blocked, and another component is chosen to continue. If $P_{j,2}$ for $j \neq i$ is chosen, then it never stops replacing $[\sigma(i),i,a]$ with itself, see (11). Clearly, as there is no nonterminal of the form $[X,a]$ in the current sentential form, neither $P_{k,1}$ nor $P_{k,3}$ are applicable for any k. Therefore, P_0 has to be chosen in which $\langle u \rangle$ is replaced with u, $[\sigma(i),i,a]$ with $[\sigma(i),a]$, and if there is no nonterminal except for $[\sigma(i),a]$ with a being a terminal or λ, then it is replaced with a as well.

In $P_{i,3}$, the system simulates the derivation step replacing the first nonterminal of the sentential form, say $[Z',A]$, with, say, $\langle Z,u \rangle$. Then, P_0 continues the derivation by replacing $\langle Z,u \rangle$ with $[Z,u_1]u_2 \ldots u_{|u|}$, where $u_i \in V$, $i = 1,\ldots,|u|$, see (3) and (4). The proof then proceeds by induction.

Finally, the lemma follows from Theorem 1. □

Corollary 6. *$L(G) \in \mathscr{L}(CD,F-\lambda)$ for any λ-free programmed grammar G with appearance checking.*

Proof. This follows from the proof of the previous lemma so that all productions constructed in (4) are removed, and no second component of any nonterminal of the form $[Z,a]$ or $\langle Z,a \rangle$ is allowed to be λ. This is correct because there is no λ-production in G and if the system replaces the first nonterminal of the sentential form, a nonterminal of the form $\langle Z,u \rangle$ is obtained, where $u \neq \lambda$. Then, P_0 continues the derivation by replacing $\langle Z,u \rangle$ with $[Z,u_1]u_2 \ldots u_{|u|}$, where $u_i \in V$, $i = 1,\ldots,|u|$, $|u| \geq 1$. □

Corollary 7. *$\mathscr{L}(P,ac) \subseteq \mathscr{L}(CD,F)$ and $\mathscr{L}(P-\lambda,ac) \subseteq \mathscr{L}(CD,F-\lambda)$.*

Proof. It follows from Lemma 5 and Corollary 6, respectively. □

As a result, we have the following theorem. For a proof of the proper inclusion see [5].

Theorem 8.

(1) $\mathscr{L}(CD,F) = \mathscr{L}(RE)$,
(2) $\mathscr{L}(CD,F-\lambda) = \mathscr{L}(P-\lambda,ac) \subset \mathscr{L}(CS)$.

5. Conclusion

This section summarizes the results and open problems concerning forbidding CD grammar systems working in the terminating derivation mode (see Fig. 1).

(1) $\mathscr{L}(F) \subset \mathscr{L}(REC) \subset \mathscr{L}(CD,F) = \mathscr{L}(RE)$.
(2) $\mathscr{L}(P-\lambda,ac) \not\subseteq \mathscr{L}(F)$.
(3) $\mathscr{L}(ET0L) \subset \mathscr{L}(F-\lambda) \subset \mathscr{L}(CD,F-\lambda) = \mathscr{L}(P-\lambda,ac) \subset \mathscr{L}(CS)$.
(4) $\mathscr{L}(LF-\lambda) = \mathscr{L}(LF) = \mathscr{L}(CF)$.
(5) $\mathscr{L}(CD,LF) = \mathscr{L}(RE)$.
(6) $\mathscr{L}(CD,LF-\lambda) = \mathscr{L}(CS)$.

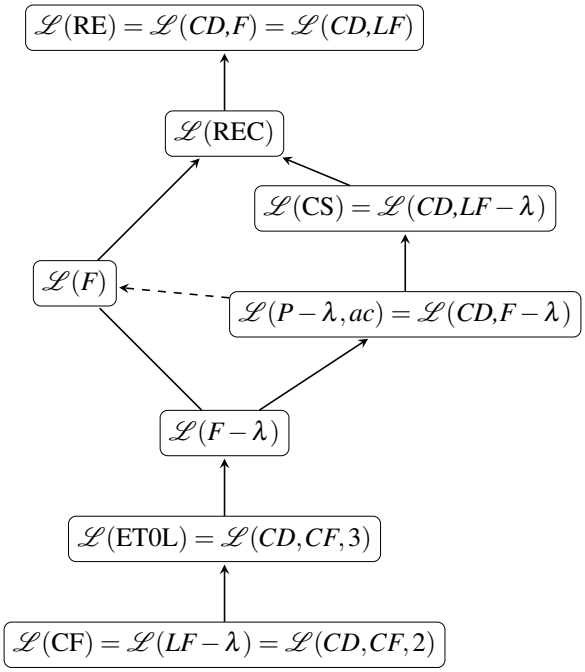

Fig. 1. A hierarchy of language families. If two families are connected by a line (an arrow), then the upper family includes (includes properly) the lower family. If two families are not connected, then they are not necessarily incomparable. If two families are connected by a dashed arrow, then the inclusion indicated by the arrow cannot hold.

Results (1)–(3) (except for the equalities proved in this paper) are taken from [1] and [6]. Results (4)–(6) are proved in [4].

Open Problems

(1) Is $\mathscr{L}(F-\lambda) \subset \mathscr{L}(F)$?
(2) Is $\mathscr{L}(F) \subseteq \mathscr{L}(CS)$?
(3) Even more, is $\mathscr{L}(F) \subseteq \mathscr{L}(P-\lambda, ac)$?

Note that it is known that the emptiness problem for $\mathscr{L}(F)$ is decidable and that $\mathscr{L}(F)$ is closed under intersection with regular languages (see [1] and the references therein). Furthermore, for $X \in \{\mathscr{L}(P-\lambda, ac), \mathscr{L}(CS)\}$, every recursively enumerable language is the homomorphic image of a language in X. Then, by Theorem 3(b) in [3], $X \setminus \mathscr{L}(F) \neq \emptyset$.

Finally, it is well-known (see [2]) that the family of languages generated by CD grammar systems with context-free components (considering the t-mode) coincides either with the family of context-free languages (if they have no more than two components), or with the family of ET0L languages (if they have three or more components). Therefore, it is surprising that although

$$\mathscr{L}(CF) = \mathscr{L}(LF-\lambda) \subset \mathscr{L}(F-\lambda),$$

it holds that for any $n \geq 2$,

$$\mathscr{L}(CD, CF, n) \subset \mathscr{L}(CD, F-\lambda) \subset \mathscr{L}(CD, LF-\lambda).$$

Acknowledgments

This work was supported by the Czech Ministry of Education under the Research Plan No. MSM 0021630528. The author thanks both referees for their helpful suggestions improving the presentation of this paper.

References

[1] H. Bordihn and H. Fernau. Accepting grammars and systems. Technical Report 9/94, Universitat Karlsruhe, Fakultat fur Informatik, 1994.
[2] E. Csuhaj-Varjú, J. Dassow, J. Kelemen, and Gh. Păun. *Grammar Systems: A Grammatical Approach to Distribution and Cooperation*. Gordon and Breach Science Publishers, Topics in Computer Mathematics 5, Yverdon, 1994.
[3] F. Hinz and J. Dassow. An undecidability result for regular languages and its application to regulated rewriting. *EATCS Bulletin*, 38:168–173, 1989.
[4] A. Meduna, T. Masopust, and F. Goldefus. Left-forbidding cooperating distributed grammar systems. Submitted.
[5] D. J. Rosenkrantz. Programmed grammars and classes of formal languages. *J. ACM*, 16(1):107–131, 1969.
[6] G. Rozenberg and A. Salomaa, editors. *Handbook of Formal Languages*, volume 1–3. Springer-Verlag, Berlin, 1997.
[7] A. Salomaa. *Formal languages*. Academic Press, New York, 1973.
[8] A. P. J. van der Walt. Random context grammars. In *Proceedings of the Symposium on Formal Languages*. 1970.