SCHEDAE INFORMATICAE
VOLUME 19 2010

Some Power-Decreasing Derivation Restrictions in Grammar Systems

Alexander Meduna, Martin Čermák, Tomáš Masopust
Brno University of Technology, Faculty of Information Technology,
Božetěchova 2, 612 66 Brno, Czech Republic
e-mail: {*meduna, icermak, masopust*}*@fit.vutbr.cz*

Abstract. In this paper, we place some left restrictions on derivations in CD grammar systems with phrase-structure grammars, controlled by the regular languages. The first restriction requires that every production is always applied within the first k nonterminals in every sentential form, for some $k \geq 1$. The second restriction says how many blocks of non-terminals can be in each sentential form. The third restriction extends the second restriction and says how many blocks of non-terminals with limited length can be in each sentential form. We demonstrate that under these restrictions, the grammar systems generate different families of languages. Indeed, under the first restriction, these systems generate only context-free languages. Under the second restriction, even one-component systems characterize the entire family of recursively enumerable languages. In the end, the family of languages generated by grammar systems under the third restriction is equal to the family of languages generated by programmed grammars with context-free rules without ϵ-rules of finite index.

Keywords: grammar systems, derivation restriction, generative power.

1. Introduction

The formal language theory has investigated various left restrictions placed on derivations in grammars working in a context-free way. In ordinary context-free grammars, these restrictions have no effect on the generative power. In terms of regulated context-free grammars, the formal language theory has introduced a broad variety of leftmost derivation restrictions, many of which change their generative power (see [1, 2, 3, 4, 5, 7, 9, 10, 11, 12, 13, 14]). In terms of grammars, working in a context-sensitive way, significantly fewer left derivation restrictions have been discussed in the language theory. Indirectly, this theory has placed some restrictions on the productions so the resulting grammars make only derivations in a left way (see [1, 2]). This theory also directly restricted derivations in the strictly leftmost way so the rewritten symbols are preceded only by terminals in the sentential form during each derivation step (see [11]). In essence, all these restrictions result in decreasing the generative power to the power of context-free grammars (see page 198 in [16]). The present paper generalises the discussion of this topic by investigating regularly controlled cooperating distributed (CD) grammar systems (see chapter 4 in [16]) whose components are phrase-structure grammars restricted in some new left ways.

More specifically, the first restriction requires that each production is always applied within the first k nonterminals in every sentential form, for some $k \geq 1$. The second restriction says how many blocks of non-terminals can be in each sentential form. The third restriction extends the second restriction and says how many blocks of non-terminals with limited length can be in each sentential form. As already stated, we investigate these restrictions in terms of CD grammar systems which are controlled by regular languages and whose components are phrase-structure grammars. Without these restrictions, these systems generate the family of recursively enumerable languages. We demonstrate that under these restrictions, the grammar systems generate different families of languages. Indeed, under the first restriction, these systems generate only the family of context-free languages. Under the second restriction, even one-component versions of these systems generate the entire family of recursively enumerable languages. In the end, the family of languages generated by grammars systems under the third restriction is equal to the family of languages generated by programmed grammars with context-free rules without ϵ-rules of finite index.

2. Preliminaries

In this paper, we assume that the reader is familiar with the formal language theory (see [15]). For a set, Q, $|Q|$ denotes the cardinality of Q. For an alphabet, V, V^* represents the free monoid generated by V. The identity of V^* is denoted by ε. Set $V^+ = V^* - \{\varepsilon\}$; algebraically, V^+ is thus the free semigroup generated by V. For $w \in V^*$, $|w|$ denotes the length of w, w^R denotes the mirror image of w, $alph(w) = \{a_1, \dots, a_n \in V : \ w = a_1 \dots a_n\}$, $sub(w)$ denotes the set of all substrings of w, and $suf(w)$ denotes the set of all suffixes of w. For $\Lambda \subseteq V^*$, let $suf(\Lambda) = \{w \in suf(w') : w' \in \Lambda\}$. Analogously for a set of prefixes, we define $pref(w)$ and $pref(\Lambda)$. For $W \subseteq V$, $occur(w, W)$ denotes the number of occurrences of symbols from W in w.

A *finite automaton* is a quintuple $M = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of states, Σ is an alphabet, $q_0 \in Q$ is the initial state, δ is a finite set of rules of the form $qa \to p$, where $p, q \in Q$ and $a \in \Sigma \cup \{\varepsilon\}$, $F \subseteq Q$ is a set of final states. A configuration of M is any word from $Q\Sigma^*$. For any configuration qay, where $q \in Q$, $y \in \Sigma^*$, $a \in \Sigma \cup \{\varepsilon\}$ and any $qa \to p \in \delta$, M makes a move from configuration qay to configuration py according to $qa \to p$, written as $qay \Rightarrow py\, [qa \to p]$, or, simply, $qay \Rightarrow py$. If $x, y \in Q\Sigma^*$ and $m > 0$, then $x \Rightarrow^m y$ if there exists a sequence $x_0 \Rightarrow x_1 \Rightarrow \cdots \Rightarrow x_m$, where $x_0 = x$ and $x_m = y$. Then, we say $x \Rightarrow^+ y$ if there exists $m > 0$ such that $x \Rightarrow^m y$, and $x \Rightarrow^* y$ if $x = y$ or $x \Rightarrow^+ y$. If $w \in \Sigma^*$ and $q_0 w \Rightarrow^* f$, where $f \in F$, then w is accepted by M, and $q_0 w \Rightarrow^* f$ is an acceptance of w in M. The language of M is defined as $\mathcal{L}(M) = \{w \in \Sigma^* : q_0 w \Rightarrow^* f \text{ is an acceptance of } w\}$. Let **REG** denote the family of regular languages.

A *pushdown automaton* is a septuple $M = (Q, \Sigma, \Omega, \delta, q_0, Z_0, F)$, where Q is a finite set of states, Σ is an alphabet, $q_0 \in Q$ is the initial state, Ω is a pushdown alphabet, δ is a finite set of rules of the form $Zqa \to \gamma p$, where $p, q \in Q$, $Z \in \Omega$, $a \in \Sigma \cup \{\varepsilon\}$, $\gamma \in \Omega^*$, $F \subseteq Q$ is a set of final states, and $Z_0 \in \Omega$ is the initial pushdown symbol. A configuration of M is any word from $\Omega^* Q \Sigma^*$. For any configuration $xAqay$, where $x \in \Omega^*$, $y \in \Sigma^*$, $q \in Q$, and any $Aqa \to \gamma p \in \delta$, M makes a move from configuration $xAqay$ to configuration $x\gamma py$ according to $Aqa \to \gamma p$, written as $xAqay \Rightarrow x\gamma py\, [Aqa \to \gamma p]$, or, simply, $xAqay \Rightarrow x\gamma py$. If $x, y \in \Omega^* Q \Sigma^*$ and $m > 0$, then $x \Rightarrow^m y$ if there exists a sequence $x_0 \Rightarrow x_1 \Rightarrow \cdots \Rightarrow x_m$, where $x_0 = x$ and $x_m = y$. Then, we say $x \Rightarrow^+ y$ if there exists $m > 0$ such that $x \Rightarrow^m y$, and $x \Rightarrow^* y$ if $x = y$ or $x \Rightarrow^+ y$. If $w \in \Sigma^*$ and $Z_0 q_0 w \Rightarrow^* f$, where $f \in F$, then w is accepted by M, and $Z_0 q_0 w \Rightarrow^* f$ is an acceptance of w in M. The language of M is defined as $\mathcal{L}(M) = \{w \in \Sigma^* : Z_0 q_0 w \Rightarrow^* f \text{ is an acceptance of } w\}$. Let **CF** denote the family of context-free languages and $CF - \varepsilon$ denote the family of languages generated by context-free grammars without ε-rules.

A *phrase structure grammar* is a quadruple $G = (N, T, S, P)$, where N and T are alphabets such that $N \cap T = \emptyset$, $S \in N$, and P is a finite set of productions of the form $\alpha \to \beta$, where $\alpha \in N^+$ and $\beta \in (N \cup T)^*$. If $\alpha \to \beta \in P$, $u = x_0\alpha x_1$, and $v = x_0\beta x_1$, where $x_0, x_1 \in V^*$, then $u \Rightarrow v$ $[\alpha \to \beta]$ in G or, simply, $u \Rightarrow v$. Let $\Rightarrow^+$ and $\Rightarrow^*$ denote the transitive closure of $\Rightarrow$ and the transitive-reflexive closure of $\Rightarrow$, respectively. The *language of G* is denoted by $\mathcal{L}(G)$ and defined as $\mathcal{L}(G) = \{w \in T^* : S \Rightarrow^* w\}$. Let **RE** denote the family of recursively enumerable languages.

A *programmed grammar* (see [6]) is a septuple $G = (N, T, S, P, \Lambda, \sigma, \phi)$, where N and T are alphabets such that $N \cap T = \emptyset$, $S \in N$, P is a finite set of productions of the form $\alpha \to \beta$ and Λ is a finite s et of labels for the productions in P. Λ can be interpreted as a function which outputs a production when being given a label. σ and ϕ are functions from Λ into the 2^Λ. For $(x, r_1), (y, r_2) \in V^* \times \Lambda$ and $\Lambda(r_1) = (\alpha \to \beta)$, we write $(x, r_1) \Rightarrow (y, r_2)$ iff either $x = x_1\alpha x_2$, $y = x_1\beta x_2$ and $r_2 \in \sigma(r_1)$, or $x = y$, and rule $\alpha \to \beta$ is not applicable to x, and $r_2 \in \phi(r_1)$. Let $\Rightarrow^+$ and $\Rightarrow^*$ denote the transitive closure of $\Rightarrow$ and the transitive-reflexive closure of $\Rightarrow$, respectively. The *language of G* is denoted by $\mathcal{L}(G)$ and defined as $\mathcal{L}(G) = \{w \in T^* : (S, r_1) \Rightarrow^* (w, r_2)$, for some $r_1, r_2 \in \Lambda\}$. Let $\mathbf{P}(\mathbf{CF}, ac)$ denote the family of languages generated by programmed grammars containing only context-free rules. If $\phi(r) = \emptyset$ for each $r \in \Lambda$, we are led to the family $\mathbf{P}(\mathbf{CF})$.

Let G be a programmed grammar. For a derivation $D : S = w_1 \Rightarrow w_2 \Rightarrow \ldots \Rightarrow w_n = w$, $w \in T^*$, of G, $ind(D, G) = max\{|w_i|_N \ : 1 \le i \le n\}$, and for $w \in T^*$, $ind(w, G) = min\{ind(D, G) \ : D$ is a derivation of w in $G\}$. The *index of G* is $ind(G) = sup\{ind(w, G) \ : w \in L(G)\}$. For a language L in the family $\mathbf{P}(X)$ of languages generated by programmed grammars with productions of type X, $ind_X(L) = inf\{ind(G) \ : L(G) = L$ and G has productions of type $X\}$. For a family $\mathbf{P}(X)$, $\mathbf{P}_n(X) = \{L \ : L \in \mathbf{P}(X)$ and $ind_X(L) \le n$ for $n \ge 1\}$ (see [6]).

3. Definitions

Now, we define the three derivation restrictions discussed in this paper. Let $G = (N, T, S, P)$ be a phrase structure grammar. Let $V = N \cup T$ be the total alphabet of G.

3.1. First restriction

Let $l \geq 1$. If there is $\alpha \rightarrow \beta \in P$, $u = x_0\alpha x_1$, and $v = x_0\beta x_1$, where $x_0 \in T^*N^*$, $x_1 \in V^*$, and $\mathrm{occur}(x_0\alpha, N) \leq l$, then $u\ {}_{l}\Leftrightarrow\ v\ [\alpha \rightarrow \beta]$ in G or, simply, $u\ {}_{l}\Leftrightarrow\ v$. Let ${}_{l}\Leftrightarrow^{k}$ denote the k-fold product of ${}_{l}\Leftrightarrow$, where $k \geq 0$. Furthermore, let ${}_{l}\Leftrightarrow^{*}$ denote the transitive-reflexive closure of ${}_{l}\Leftrightarrow$.

3.2. Second and third restrictions

Let $m, h \geq 1$. $W(m)$ denotes the set of all strings $x \in V^*$ satisfying 1 given next. $W(m, h)$ denotes the set of all strings $x \in V^*$ satisfying 1 and 2.

1. $x \in (T^*N^*)^mT^*$,

2. ($y \in \mathrm{sub}(x)$ and $|y| > h$) implies $\mathrm{alph}(y) \cap T \neq \emptyset$.

Let $u \in V^*N^+V^*$, $v \in V^*$ and $u \Rightarrow v$. $u\ {}^{h}_{m}\Leftrightarrow\ v$ if $u, v \in W(m, h)$, and $u\ {}_{m}\Leftrightarrow\ v$ if $u, v \in W(m)$. Let ${}^{h}_{m}\Leftrightarrow^{k}$ and ${}_{m}\Leftrightarrow^{k}$ denote the k-fold product of ${}^{h}_{m}\Leftrightarrow$ and ${}_{m}\Leftrightarrow$, respectively, where $k \geq 0$. Furthermore, let ${}^{h}_{m}\Leftrightarrow^{*}$ and ${}_{m}\Leftrightarrow^{*}$ denote the transitive-reflexive closure of ${}^{h}_{m}\Leftrightarrow$ and ${}_{m}\Leftrightarrow$, respectively.

3.3. Cooperating distributed grammar system

A *cooperating distributed grammar system* (a *CD grammar system* for short) is an $(n+3)$-tuple $\Gamma = (N, T, S, P_1, \ldots, P_n)$, where N, T are alphabets such that $N \cap T = \emptyset$, $V = N \cup T$, $S \in N$, and $G_i = (N, T, S, P_i), 1 \leq i \leq n$, is a phrase structure grammar.

Let $u \in V^*N^+V^*$, $v \in V^*$, $k \geq 0$. Then, we write $u\ {}_{l}\Leftrightarrow^{k}_{P_i}\ v$, $u\ {}^{h}_{m}\Leftrightarrow^{k}_{P_i}\ v$, and $u\ {}_{m}\Leftrightarrow^{k}_{P_i}\ v$ to denote that $u\ {}_{l}\Leftrightarrow^{k}\ v$, $u\ {}^{h}_{m}\Leftrightarrow^{k}\ v$, and $u\ {}_{m}\Leftrightarrow^{k}\ v$, respectively, was performed by P_i. Analogously, we write $u\ {}_{l}\Leftrightarrow^{*}_{P_i}\ v$, $u\ {}^{h}_{m}\Leftrightarrow^{*}_{P_i}\ v$, $u\ {}_{m}\Leftrightarrow^{*}_{P_i}\ v$, $u\ {}_{l}\Leftrightarrow^{+}_{P_i}\ v$, $u\ {}^{h}_{m}\Leftrightarrow^{+}_{P_i}\ v$, and $u\ {}_{m}\Leftrightarrow^{+}_{P_i}\ v$.

Moreover, we write $u\ {}_{l}\Leftrightarrow^{t}_{P_i}\ v$ if $u\ {}_{l}\Leftrightarrow^{+}_{P_i}\ v$ and there is no w such that $v\ {}_{l}\Leftrightarrow_{P_i}\ w$. Analogously, we write $u\ {}^{h}_{m}\Leftrightarrow^{t}_{P_i}\ v$ and $u\ {}_{m}\Leftrightarrow^{t}_{P_i}\ v$.

For a CD grammar system $\Gamma = (N, T, S, P_1, \ldots, P_n)$ and a controll language L, we set

$$
\begin{aligned}
{}_{\mathrm{l}}\mathcal{L}^L(\Gamma) &= \{w \in T^* : S \; {}_{l}\!\Leftrightarrow\!\!\Rightarrow^{t}_{P_{i_1}} w_1 \; {}_{l}\!\Leftrightarrow\!\!\Rightarrow^{t}_{P_{i_2}} \dots {}_{l}\!\Leftrightarrow\!\!\Rightarrow^{t}_{P_{i_p}} w_p = w, \\
&\qquad p \geq 1, 1 \leq i_j \leq n, 1 \leq j \leq p, \; i_1 i_2 \dots i_p \in L\} \\
{}_{\mathrm{N}}\mathcal{L}^L(\Gamma, m, h) &= \{w \in T^* : S \; {}^{\;h}_{m}\!\circ\!\!\Rightarrow^{t}_{P_{i_1}} w_1 \; {}^{\;h}_{m}\!\circ\!\!\Rightarrow^{t}_{P_{i_2}} \dots {}^{\;h}_{m}\!\circ\!\!\Rightarrow^{t}_{P_{i_p}} w_p = w, \\
&\qquad p \geq 1, 1 \leq i_j \leq n, 1 \leq j \leq p, \; i_1 i_2 \dots i_p \in L\} \\
{}_{\mathrm{N}}\mathcal{L}^L(\Gamma, m) &= \{w \in T^* : S \; {}_{m}\!\circ\!\!\Rightarrow^{t}_{P_{i_1}} w_1 \; {}_{m}\!\circ\!\!\Rightarrow^{t}_{P_{i_2}} \dots {}_{m}\!\circ\!\!\Rightarrow^{t}_{P_{i_p}} w_p = w, \\
&\qquad p \geq 1, 1 \leq i_j \leq n, 1 \leq j \leq p, \; i_1 i_2 \dots i_p \in L\}.
\end{aligned}
$$

Let **GS**s denote the family of all CD grammar systems. Let $l, m, h \geq 1$. Define the following language families:

$$
{}_{\mathrm{l}}GS^{\mathbf{REG}} = \{{}_{\mathrm{l}}\mathcal{L}^L(\Gamma) : \Gamma \in \mathbf{GS}\mathrm{s}, L \in \mathbf{REG}\},
$$

$$
{}_{\mathrm{N}}GS^{\mathbf{REG}}(m, h) = \{{}_{\mathrm{N}}\mathcal{L}^L(\Gamma, m, h) : \Gamma \in \mathbf{GS}\mathrm{s}, L \in \mathbf{REG}\},
$$

$$
{}_{\mathrm{N}}GS^{\mathbf{REG}}(m) = \{{}_{\mathrm{N}}\mathcal{L}^L(\Gamma, m) : \Gamma \in \mathbf{GS}\mathrm{s}, L \in \mathbf{REG}\}.
$$

4. Results

This section proves the main results of this paper:

1. $\mathbf{CF} = {}_{\mathrm{l}}GS^{\mathbf{REG}}$,
2. $\mathbf{RE} = {}_{\mathrm{N}}GS^{\mathbf{REG}}(1)$,
3. $\mathbf{P}_m(\mathbf{CF} - \varepsilon) = {}_{\mathrm{N}}GS^{\mathbf{REG}}(m, h)$.

First, we show that for any language L from ${}_{\mathrm{l}}GS^{\mathbf{REG}}$ there exists a pushdown automaton M, such that $L = \mathcal{L}(M)$ and for every pushdown automaton M' language $\mathcal{L}(M')$ is in ${}_{\mathrm{l}}GS^{\mathbf{REG}}$. That is $\mathbf{CF} = {}_{\mathrm{l}}GS^{\mathbf{REG}}$.

Lemma 1. *For every CD grammar system* $\Gamma = (N, T, S, P_1, \dots, P_n)$, *every finite automaton* $\bar{M}$ *and every* $l \geq 1$, *there is a pushdown automaton* M, *such that* $\mathcal{L}(M) = {}_{l}\mathcal{L}^{\mathcal{L}(\bar{M})}(\Gamma)$.

Proof of Lemma 1. Let $\Gamma = (N, T, S, P_1, \dots, P_n)$, $\bar{M} = (\bar{Q}, \bar{\Sigma}, \bar{\delta}, \bar{s_0}, \bar{F})$, $l \geq 1$ and $N_{left}(P) = \{\alpha \mid \alpha \to \beta \in P\}$. Consider the following pushdown automaton $M = (\{s_0, f\} \cup \{[\gamma, s, \bar{s}, i] : \gamma \in N^*, |\gamma| \leq l, s \in \{q, r, e\}, \bar{s} \in \bar{Q}, i \in \{1, \dots, n\}\}, T, T \cup N \cup \{Z\}, \delta, s_0, Z, \{f\})$, where $Z \notin T \cup N$ and δ contains rules of the following forms:

1.	$s_0 \to [S, q, \bar{s_0}, i]$	$i \in \{1, \dots, n\}$
2.	$[\gamma, q, s, i] \to (\gamma')^R[\varepsilon, r, s, i]$	if $\gamma \in N^*, \|\gamma\| \leq l$ s.t. $\gamma \;{}_l\!\Leftrightarrow\!\Rightarrow^1_{P_i} \gamma'$
3.	$a[\varepsilon, r, s, i]a \to [\varepsilon, r, s, i]$	$i \in \{1, \dots, n\}$
4.	$Z[\varepsilon, r, s, i] \to f$	if $si \to s' \in \bar{\delta}$ for some $s' \in \bar{F}$
5.	$A[A_1 \dots A_o, r, s, i] \to [A_1 \dots A_o A, r, s, i]$	if $A \in N, o < l$
6.	$[A_1 \dots A_l, r, s, i] \to [A_1 \dots A_l, e, s, i]$	$i \in \{1, \dots, n\}$
7.	$a[A_1 \dots A_o, r, s, i] \to a[A_1 \dots A_o, e, s, i]$	if $o < l, a \in T$
8.	$Z[A_1 \dots A_o, r, s, i] \to Z[A_1 \dots A_o, e, s, i]$	if $o < l$
9.	$[\gamma, e, s, i] \to [\gamma, q, s', i']$	if $sub(\gamma) \cap N_{\text{left}}(P_i) = \emptyset$, $si \to s' \in \bar{\delta}$ $i' \in \{1, \dots, n\}$
10.	$[\gamma, e, s, i] \to [\gamma, q, s, i]$	if $sub(\gamma) \cap N_{\text{left}}(P_i) \neq \emptyset$

We prove that $\mathcal{L}(M) = {}_l\mathcal{L}_f^{\mathcal{L}(\bar{M})}(\Gamma)$.

($\subseteq$:) First, we prove the following claim.

Claim A. If $Z\delta^R[\gamma, q, s, i_1]w \Rightarrow^* f$ in M, then $\gamma\delta \;{}_l\!\Leftrightarrow\!\Rightarrow^t_{P_{i_1}} w_1 \;{}_l\!\Leftrightarrow\!\Rightarrow^t_{P_{i_2}} w_2 \dots {}_l\!\Leftrightarrow\!\Rightarrow^t_{P_{i_p}} w_p = w$, $p \geq 0$ in Γ and $i_1 \dots i_p \in suf(\mathcal{L}(\bar{M}))$.

Proof of Claim A. By induction on the number of rules constructed in 2 used in a sequence of moves.

Basis: Only one rule constructed in 2 is used. Then,

$$Z\delta^R[\gamma, q, s, i_0]w \Rightarrow Z(\gamma'\delta)^R[\varepsilon, r, s, i_0]w \Rightarrow^{|\gamma'\delta|} Z[\varepsilon, r, s, i_0] \Rightarrow f,$$

where $\gamma = \gamma_0\alpha\gamma_1$, $\gamma' = \gamma_0\beta\gamma_1$, $\alpha \to \beta \in P_{i_0}$, $\gamma \in N^+, \gamma'\delta \in T^*$. Therefore, $\gamma_0 = \gamma_1 = \varepsilon$, $\gamma'\delta = w$. Then,

$$\gamma\delta \;{}_l\!\Leftrightarrow\!\Rightarrow_{P_{i_0}} w.$$

By a rule constructed in 4 $i_0 \in suf(\mathcal{L}(\bar{M}))$ and the basis holds.

Induction hypothesis: Suppose that the claim holds for all sequences of moves containing no more than j rules constructed in 2.

Induction step: Consider a sequence of moves containing $j + 1$ rules constructed in 2:

$$\begin{array}{lll} & Z\delta^R[\gamma,q,s,i_0]w & \\ \Rightarrow & Z(\gamma'\delta)^R[\varepsilon,r,s,i_0]w & \text{(by a prod. constructed in 2)} \\ \Rightarrow^* & Z(\delta')^R[\varepsilon,r,s,i_0]w' & \text{(by prod. constructed in 3)} \\ \Rightarrow^* & Z(\delta'')^R[\gamma'',r,s,i_0]w' & \text{(by prod. constructed in 5)} \\ \Rightarrow & Z(\delta'')^R[\gamma'',e,s,i_0]w' & \text{(by a prod. constructed in 6, 7 or 8)} \\ \Rightarrow & Z(\delta'')^R[\gamma'',q,s',i_1]w' & \text{(by a prod. constructed in 9 or 10)} \\ \Rightarrow^* & f & \end{array}$$

where $\gamma = \gamma_0\alpha\gamma_1$, $\gamma' = \gamma_0\beta\gamma_1$, $\alpha \to \beta \in P_{i_0}$, $\delta' \in NV^* \cup \{\varepsilon\}$, $v \in T^*$, $\gamma'\delta = v\delta'$, $vw' = w$, $\delta' = \gamma''\delta''$, either $si \to s'$ or $s = s'$, and one of the following holds:

- $|\gamma''| = l$, or
- $|\gamma''| < l$ and $\delta'' \in TV^* \cup \{\varepsilon\}$.

Then, by the rule $\alpha \to \beta$,

$$\gamma_0\alpha\gamma_1\delta \ {}_l\!\Leftrightarrow\!\!\!\Rightarrow_{P_{i_0}} \gamma_0\beta\gamma_1\delta,$$

where $|\gamma_0\alpha\gamma_1| \leq l$, $\gamma_0\beta\gamma_1\delta = v\delta' = v\gamma''\delta''$ and, by the induction hypothesis,

$$v\gamma''\delta'' \ {}_l\!\Leftrightarrow\!\!\!\Rightarrow^t_{P_{i_1}} vw_1 \ {}_l\!\Leftrightarrow\!\!\!\Rightarrow^t_{P_{i_2}} vw_2 \ldots {}_l\!\Leftrightarrow\!\!\!\Rightarrow^t_{P_{i_p}} vw_p = vw \text{ and}$$

$$i_1 \ldots i_p \in suf(\mathcal{L}(\bar{M})),$$

where $p \geq 0$.

If a rule constructed in 9 was used, $\gamma_0\alpha\gamma_1\delta \ {}_l\!\Leftrightarrow\!\!\!\Rightarrow^t_{P_{i_0}} \gamma_0\beta\gamma_1\delta$ is a t-mode derivation, $i_0i_1i_2 \ldots i_p \in suf(\mathcal{L}(\bar{M}))$ and the claim holds.

If a rule constructed in 10 was used, $i_0 = i_1$, $\gamma_0\alpha\gamma_1\delta \ {}_l\!\Leftrightarrow\!\!\!\Rightarrow^t_{P_{i_1}} vw_1$, $i_1i_2 \ldots i_p \in suf(\mathcal{L}(\bar{M}))$ and the claim holds. □

Let $Zs_0w \Rightarrow Z[S,q,\bar{s_0},i_1]w$, by a rule constructed in 1. By the previous claim, $Z[S,q,\bar{s_0},i_1]w \Rightarrow^* f$ implies $S \ {}_l\!\Leftrightarrow\!\!\!\Rightarrow^t_{P_{i_1}} w_1 \ {}_l\!\Leftrightarrow\!\!\!\Rightarrow^t_{P_{i_2}} w_2 \ldots {}_l\!\Leftrightarrow\!\!\!\Rightarrow^t_{P_{i_p}} w_p = w, p \geq 0$ in Γ and $i_1 \ldots i_p \in suf(\mathcal{L}(\bar{M}))$. □

($\supseteq$:) First, we prove the following claim.

Claim B. If $\tau_0x_0 \ {}_l\!\Leftrightarrow\!\!\!\Rightarrow^t_{P_{i_1}} w_1 \ {}_l\!\Leftrightarrow\!\!\!\Rightarrow^t_{P_{i_2}} w_2 \ldots {}_l\!\Leftrightarrow\!\!\!\Rightarrow^t_{P_{i_p}} w_p = w$ in Γ, where $p \geq 0$, $\tau_0 \in N^+$, $x_0 \in TV^* \cup \{\varepsilon\}$, $w_i \in V^*$, $i \in \{1,\ldots,p-1\}$, $w_p \in T^*$ and $i_1 \ldots i_p \in suf(\mathcal{L}(\bar{M}))$, then $Z(\tau_0^2x_0)^R[\tau_0^1,q,s,i_1]w \Rightarrow^* f$, for some $s \in \bar{Q}$, where $\tau_0 = \tau_0^1\tau_0^2$, $|\tau_0| \leq l$ implies $\tau_0^1 = \tau_0$, and $|\tau_0| > l$ implies $|\tau_0^1| = l$.

Proof of Claim B. By induction on the length of derivations.

Basis: Let $\tau_0 x_0 \ {}_l\!\Leftrightarrow_{P_{i_0}} \tau_0' x_0 = w$, where $\tau_0^1 = \gamma_0 \alpha \gamma_1$, $\tau_0' = \gamma_0 \beta \gamma_1 \tau_0^2$, $\alpha \to \beta \in P_{i_0}$, $\tau_0' x_0 \in {}_1\mathcal{L}_f^{\mathcal{L}(\bar{M})}(\Gamma)$. Therefore, $\gamma_0 = \gamma_1 = \tau_0^2 = \varepsilon$ and for some $s \in \bar{Q}$, $si_0 \to s' \in \bar{\delta}$, where $s' \in \bar{F}$. M simulates this derivation step in the following way:

$$
\begin{array}{lll}
 & Z(\tau_0^2 x_0)^R[\tau_0^1, q, s, i_0]w & \\
\Rightarrow & Z(\tau_0' x_0)^R[\varepsilon, r, s, i_0]w & \text{(by a prod. constructed in 2)} \\
\Rightarrow^{|\tau_0' x_0|} & Z[\varepsilon, r, s, i_0] & \text{(by prod. constructed in 3)} \\
\Rightarrow & f & \text{(by a prod. constructed in 4).}
\end{array}
$$

Therefore, the basis holds.

Induction hypothesis: Suppose that the claim holds for all derivations of length j or less.

Induction step: Consider a derivation of length $j + 1$:

$$\tau_0 x_0 \ {}_l\!\Leftrightarrow_{P_{i_0}} \tau_0' x_0 = v_1 \tau_1 x_1 \ {}_l\!\Leftrightarrow^t_{P_{i_1}} v_1 w_1 \ {}_l\!\Leftrightarrow^t_{P_{i_2}} w_2 \ldots {}_l\!\Leftrightarrow^t_{P_{i_p}} w_p = w = v_1 w',$$

where $p \geq 0$, $v_1 \in T^*$, $\tau_0, \tau_1 \in N^+$, $\tau_0' \in V^*$, $x_0, x_1 \in TV^* \cup \{\varepsilon\}$, $w_i \in V^*, i \in \{1, \ldots, p-1\}$, $w_p, w' \in T^*$. Then, M simulates this derivation as follows:

$$
\begin{array}{lll}
 & Z(\tau_0^2 x_0)^R[\tau_0^1, q, s, i_0]w & \\
\Rightarrow & Z(\tau_0' x_0)^R[\varepsilon, r, s, i_0]w & \text{(by a prod. constructed in 2)} \\
= & Z(v_1 \tau_1 x_1)^R[\varepsilon, r, s, i_0]v_1 w' & \\
\Rightarrow^{|v_1|} & Z(\tau_1 x_1)^R[\varepsilon, r, s, i_0]w' & \text{(by prod. constructed in 3)} \\
\Rightarrow^{|\tau_1^1|} & Z(\tau_1^2 x_1)^R[\tau_1^1, r, s, i_0]w' & \text{(by prod. constructed in 5)} \\
\Rightarrow & Z(\tau_1^2 x_1)^R[\tau_1^1, e, s, i_0]w' & \text{(by a prod. constructed in 6, 7, or 8)} \\
\Rightarrow & Z(\tau_1^2 x_1)^R[\tau_1^1, q, s', i_1]w' & \text{(by a prod. constructed in 9 or 10)} \\
\Rightarrow^* & f & \text{(by the induction hypothesis)}
\end{array}
$$

If $\tau_0 x_0 \ {}_l\!\Leftrightarrow_{P_{i_0}} \tau_0' x_0$ is a t-mode derivation, a rule of type 9 is used during the simulation. Otherwise, a rule of type 10 is used (and therefore $i_0 = i_1$). Hence, the claim holds. □

Let $S \ {}_l\!\Leftrightarrow_{P_{i_0}} u \tau_0 x_0 \ {}_l\!\Leftrightarrow^t_{P_{i_1}} \ldots {}_l\!\Leftrightarrow^t_{P_{i_p}} uw$, where $p \geq 0$, $u, w \in T^*$, $\tau_0 \in N^+ \cup \{\varepsilon\}$, $x_0 \in TV^* \cup \{\varepsilon\}$ and $i_1 \ldots i_p \in \mathcal{L}(\bar{M})$. If $u \tau_0 x_0 \notin T^*$, M simulates this derivation in the following way:

$$
\begin{array}{lll}
 & Zs_0uw & \\
\Rightarrow & Z[S,q,s,i_0] & \text{(by a prod. constructed in 1)} \\
\Rightarrow & Z(u\tau_0x_0)^R[\varepsilon,r,s,i_0]uw & \text{(by a prod. constructed in 2)} \\
\Rightarrow^{|u|} & Z(\tau_0x_0)^R[\varepsilon,r,s,i_0]w & \text{(by prod. constructed in 3)} \\
\Rightarrow^{|\tau_0^1|} & Z(\tau_0^2x_0)^R[\tau_0^1,r,s,i_0]w & \text{(by prod. constructed in 5)} \\
\Rightarrow & Z(\tau_0^2x_0)^R[\tau_0^1,e,s,i_0]w & \text{(by a prod. constructed in 6, 7, or 8)} \\
\Rightarrow & Z(\tau_0^2x_0)^R[\tau_0^1,q,s',i_1]w & \text{(by a prod. constructed in 9 or 10)} \\
\Rightarrow^* & f & \text{(by the previous claim)}
\end{array}
$$

If $u\tau_0x_0 \in T^*$, M simulates this derivation in the following way:

$$
\begin{array}{lll}
 & Zs_0uw & \\
\Rightarrow & Z[S,q,s,i_0] & \text{(by a prod. constructed in 1)} \\
\Rightarrow & Z(u\tau_0x_0)^R[\varepsilon,r,s,i_0]uw & \text{(by a prod. constructed in 2)} \\
\Rightarrow^{|u\tau_0x_0|} & Z[\varepsilon,r,s,i_0]w & \text{(by prod. constructed in 3)} \\
\Rightarrow & f & \text{(by a prod. constructed in 4)}
\end{array}
$$

From the previous claims, it follows that the lemma holds. □

By the previous lemma, we have the following result.

Theorem 1. Let l be a positive integer. Then, $\mathbf{CF} = {}_{\mathrm{l}}GS^{\mathbf{REG}}$.

Proof of Theorem 1. One inclusion is clear, the other follows from Lemma 1. □

The Theorem 1 says, that grammar systems under the first restriction are much weaker than grammar systems without this restriction. Now, we prove that the second restriction in this paper has no efect on the generative power.

Theorem 2. $\mathbf{RE} = {}_{\mathrm{N}}GS^{\mathbf{REG}}(1)$.

Proof of Theorem 2. It is well-known (see [8]) that any recursively enumerable language L is generated by a grammar G in the Geffert normal form, i.e., by a grammar of the form

$$G = (\{S, A, B, C\}, T, P \cup \{ABC \to \varepsilon\}, S),$$

where P contains context-free productions of the form

$$
\begin{array}{l}
S \to uSa \\
S \to uSv \\
S \to uv,
\end{array}
$$

where $u \in \{A, AB\}^*$, $v \in \{BC, C\}^*$, and $a \in T$. In addition, any terminal derivation in G is of the form $S \Rightarrow {}^*w_1w_2w$ by productions from P, where $w_1 \in \{A, AB\}^*$, $w_2 \in \{BC, C\}^*$, $w \in T^*$, and $w_1w_2w \Rightarrow {}^*w$ is derived by $ABC \rightarrow \varepsilon$.

Clearly, G is a CD grammar system with only one component. Set the control language to be $\{1\}^*$. Then, the theorem holds. □

The last Theorem says that generative power of grammar systems under the third restriction is less than generative power of grammar systems without any restriction.

PROPOSITION. For any m, $h \geq 1$, $\mathbf{P}_m(\mathbf{CF} - \varepsilon) = {}_{\mathrm{N}}GS^{\mathbf{REG}}(m, h)$.

PROOF OF PROPOSITION. All strings in the derivation contain no more than m blocks of nonterminals and these blocks are also of length no more than h. Hence, it is possible to represent each possible block by a single nonterminal and create an equivalent grammar system, which contains only context-free productions. From this and from Theorem 7.10 in [6], the proposition holds.

5. Acknowledgements

This work was supported by the MSM0021630528 and FIT-S-10-2 grants.

6. References

[1] Baker B.S.; *Context-sesitive grammars generating context-free languages*, in: *Automata, Languages and Programming*, M. Nivat (ed.), North-Holland, Amsterdam 1972, pp. 501–506.

[2] Book R.V.; *Terminal context in context-sensitive grammars*, SIAM Journal of Computing 1, 1972, pp. 20–30.

[3] Dassow J., Fernau H., Păun Gh.; *On the Leftmost Derivation in Matrix Grammars*, International Journal of Foundations of Computer Science 10(1), 1999, pp. 61–80.

[4] Dassow J., Păun Gh.; *Regulated Rewriting in Formal Language Theory*, Springer-Verlag, Berlin 1989.

[5] Fernau H.; *Regulated Grammars under Leftmost Derivation*, Grammars 3(1), 2000, pp. 37–62.

[6] Fernau H., Holzer M., Freund R.; *External Versus Internal Hybridization for Cooperating Distributed Grammar Systems*, Technical Raport TR 185–2/FR–1/96, Technische Universität Wien (Austria), 1996.

[7] Frazier M., Page C.D.; *Prefix Grammars: An Alternative Characterization of the Regular Languages*, Information Processing Letters 51(2), 1994, pp. 67–71.

[8] Geffert V.; *Context-Free-Like Forms for the Phrase-Structure Grammars*, Proceedings of the Mathematical Foundations of Computer Science, Springer-Verlag, New York 1988, pp. 309–317.

[9] Ginsburg S., Greibach S.; *Mappings which preserve context-sensitive languages*, Information and Control 9, 1966, pp. 563–582.

[10] Hibbard T.N.; *Context-Limited Grammars*, Journal of the ACM 21, 1974, pp. 446–453.

[11] Matthews G.; *A note on symmetry in phrase structure grammars*, Information and Control 7, 1964, pp. 360–365.

[12] Matthews G.; *Two-way languages*, Information and Control 10, 1967, pp. 111–119.

[13] Meduna A.; *Matrix Grammars under Leftmost and Rightmost Restrictions*, Mathematical Linguistics and Related Topics, Romanian Academy of Sciences, Bucharest 1994, pp. 243–257.

[14] Meduna A.; *On the Number of Nonterminals in Matrix Grammars with Leftmost Derivations*, New Trends in Formal Languages: Control, Cooperation, and Combinatorics, Springer-Verlag, New York 1997, pp. 27–39.

[15] Meduna A.; *Automata and Languages: Theory and Applications*, Springer-Verlag, London 2000.

[16] Rozenberg G., Salomaa A.; *Handbook of Formal Languages*, Vol. 1, Springer-Verlag, Berlin 1997.

Received March 15, 2010