POWER-DECREASING DERIVATION RESTRICTION IN GRAMMAR SYSTEMS

Martin Čermák
Doctoral Degree Programme (1), FIT BUT
E-mail: icermak@fit.vutbr.cz

Supervised by: Alexander Meduna
E-mail: meduna@fit.vutbr.cz

ABSTRACT

In this paper, we place a left restriction on derivations in CD grammar systems with phrase–structure grammars. This restriction requires that every production is always applied within the first k nonterminals in every sentential form, for some $k \geq 1$. Under the restriction, these systems generate only family of context-free languages.

1 INTRODUCTION

The formal language theory has investigated various left restrictions in grammars working in a context-free way. In ordinary context-free grammars, this restriction has no effect on the generative power. In terms of regulated context-free grammars, the formal language theory has introduced o broad variety of leftmost derivation restrictions, many of which effect their generative power (see [1, 2, 3, 4, 5, 6, 8, 9, 11, 12, 13, 14]). The present paper introduces new left restriction on CD grammars system which working in t-mode and its components are phrase-structure grammars. The restriction says that productions are applied within the first k nonterminals in every sentential form, for some $k \geq 1$. CD grammar system under this restriction can be simulated by push-down automaton.

2 PRELIMINARIES

In this paper, we assume that the reader is familiar with 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, $sub(w)$ 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, 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 $qa \rightarrow p$, 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 $q \in Q$, $y \in \Sigma^*$ and any $qa \rightarrow p \in \delta$, M makes a move from configuration qay to configuration py according to $qa \rightarrow p$, written as $qay \Rightarrow py\,[qa \rightarrow p]$, 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 $q_0w \Rightarrow^* f$, where $f \in F$, then w is accepted by M, and $q_0w \Rightarrow^* f$ is an acceptance of w in M. The language of M is defined as $\mathcal{L}(M) = \{w \in \Sigma^* : q_0w \Rightarrow^* f \text{ is an acceptance of } w\}$. 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 $Zqa \rightarrow \gamma p$, where $p, q \in Q$, $Z \in \Omega$, $a \in \Sigma \cup \{\varepsilon\}$, $\gamma \in \Omega^*$, F is a set of final states, and Z_0 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 $Aqa \rightarrow \gamma p \in \delta$, M makes a move from configuration $xAqay$ to configuration $x\gamma py$ according to $Aqa \rightarrow \gamma p$, written as $xAqay \Rightarrow x\gamma py\,[Aqa \rightarrow \gamma p]$, 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 $Z_0q_0w \Rightarrow^* f$, where $f \in F$, then w is accepted by M, and $Z_0q_0w \Rightarrow^* f$ is an acceptance of w in M. The language of M is defined as $\mathcal{L}(M) = \{w \in \Sigma^* : Z_0q_0w \Rightarrow^* f \text{ is an acceptance of } w\}$. Let **CF** denote the family of context-free languages.

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 $\alpha \rightarrow \beta$, where $\alpha \in N^+$ and $\beta \in (N \cup T)^*$. If $\alpha \rightarrow \beta \in P$, $u = x_0\alpha x_1$, and $v = x_0\beta x_1$, where $x_0, x_1 \in V^*$, then $u \Rightarrow v\ [\alpha \rightarrow \beta]$ 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.

3 DEFINITIONS

Now, we define derivation restriction 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 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 $\text{occur}(x_0\alpha, N) \leq l$, then $u\ {}_l\!\Leftrightarrow\!\Rightarrow v\ [\alpha \rightarrow \beta]$ in G or, simply, $u\ {}_l\!\Leftrightarrow\!\Rightarrow v$. Let ${}_l\!\Leftrightarrow\!\Rightarrow^k$ denote the k-fold product of ${}_l\!\Leftrightarrow\!\Rightarrow$, where $k \geq 0$. Furthermore, let ${}_l\!\Leftrightarrow\!\Rightarrow^*$ denote the transitive-reflexive closure of ${}_l\!\Leftrightarrow\!\Rightarrow$.

Restriction says that we can only derivate within first l nonterminals from left. In the next example we can see derivation sequence under phrase-structure grammar with this restriction.

Example I:
Let $l = 3$ and $G = (\{S, A, B, C, D\}, \{a, b\}, P, S)$, where $P = \{\ S \rightarrow ABB,\ AB \rightarrow aC,\ AB \rightarrow aAC,\ CB \rightarrow BBC,\ BBC \rightarrow bD,\ DC \rightarrow bD,\ D \rightarrow \varepsilon\ \}$.

Then derivation may be:
$S\ {}_l\!\Leftrightarrow\!\Rightarrow ABB[1]\ {}_l\!\Leftrightarrow\!\Rightarrow aACB[3]\ {}_l\!\Leftrightarrow\!\Rightarrow aABBC[4]\ {}_l\!\Leftrightarrow\!\Rightarrow aaACBC[3]\ {}_l\!\Leftrightarrow\!\Rightarrow aaABBCC[4]\ {}_l\!\Leftrightarrow\!\Rightarrow aaaCBCC[2]$

${}_{l}\diamond\!\!\Rightarrow aaaBBCCC[4]\ {}_{l}\diamond\!\!\Rightarrow aaabDCC[5]\ {}_{l}\diamond\!\!\Rightarrow aaabbDC[6]\ {}_{l}\diamond\!\!\Rightarrow aaabbbD[6]\ {}_{l}\diamond\!\!\Rightarrow aaabbb[7].$

Language generated by G with restriction under $l = 3$ holds:
${}_{\text{3-left}}L(G) = \{a^n b^n \mid n \geq 1\}$.

In the following section we define cooperating distributed grammar system with restriction defined in this paper. The restriction will be applied on every derivation step through grammar system. Furthermore, switching among components of grammar system will generate a sequence of indexes. This sequence must be string from some control language.

3.2 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 $u\ {}_{l}\diamond\!\!\Rightarrow^{k}_{P_i} v$, $u\ {}^{h}_{m}\circ\!\!\Rightarrow^{k}_{P_i} v$, and $u\ {}_{m}\circ\!\!\Rightarrow^{k}_{P_i} v$ to denote that $u\ {}_{l}\diamond\!\!\Rightarrow^{k} v$, $u\ {}^{h}_{m}\circ\!\!\Rightarrow^{k} v$, and $u\ {}_{m}\circ\!\!\Rightarrow^{k} v$, respectively, was performed by P_i. Analogously, we write $u\ {}_{l}\diamond\!\!\Rightarrow^{*}_{P_i} v$, $u\ {}^{h}_{m}\circ\!\!\Rightarrow^{*}_{P_i} v$, $u\ {}_{m}\circ\!\!\Rightarrow^{*}_{P_i} v$, $u\ {}_{l}\diamond\!\!\Rightarrow^{+}_{P_i} v$, $u\ {}^{h}_{m}\circ\!\!\Rightarrow^{+}_{P_i} v$, and $u\ {}_{m}\circ\!\!\Rightarrow^{+}_{P_i} v$.

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

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

$$\begin{aligned} {}_{\text{l-left}}\mathcal{L}^{L}_{t}(\Gamma) \quad = \quad & \{w \in T^* : S\ {}_{l}\diamond\!\!\Rightarrow^{t}_{P_{i_1}} w_1\ {}_{l}\diamond\!\!\Rightarrow^{t}_{P_{i_2}} \cdots {}_{l}\diamond\!\!\Rightarrow^{t}_{P_{i_p}} w_p = w, \\ & p \geq 1, 1 \leq i_j \leq n, 1 \leq j \leq p,\ i_1 i_2 \ldots i_p \in L\} \end{aligned}$$

3.3 LANGUAGE FAMILY

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

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

4 RESULT

This section proves main result with this power-decreasing derivation restriction.

$$\mathbf{CF} = {}_{\text{l-left}}GS^{\mathbf{REG}}_{t}$$

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

Proof idea. M simulates t-mode derivations of Γ regulated by $\bar{M}$ in its state. States of M are composed of 4 elements. The first element represents first l symbols from the first continuous block of nonterminals. The second element describes a stage of a simulated derivation step. Third and fourth elements store a state of $\bar{M}$ and an index of an active component of Γ determined by $\bar{M}$.

First, M moves from the start state s_0 to state $[S, q, s_0, i]$, where S is the start nonterminal of Γ, q - means M is ready for an application of a production of the active component with index i, s_0 is the start state of automaton $\bar{M}$.

Next step simulates a derivation by rule $S \rightarrow \alpha \in P_i$ such that M puts α^R on the stack and automaton moves from $[S, q, s_0, i]$ to $[\varepsilon, r, s_0, i]$. Generally, if M is in a state $[\gamma, q, s, i]$, where $|\gamma| \leq l$, $\gamma \in N^*$ and $\gamma\ {}_l\!\!\Leftrightarrow^{t}_{P_i} \gamma'$ then M uses rule $[\gamma, q, s, i] \rightarrow (\gamma)^R[\varepsilon, r, s, i]$.

Current state of the automaton M says automaton has to remove terminals from the top of the stack, that is $\alpha B a_n \dots a_1 [\varepsilon, r, s, i] a_1 \dots a_n \omega \Rightarrow^* \alpha B[\varepsilon, r, s, i]\omega$, where $B \notin T$.

If $B \notin T \cup N$, M moves to the finish state and stops of computing. Otherwise, M must remove first l non-terminals (or fewer if there are not l non-terminals) from the top of the stack. For example, for $\alpha B[A_1 \dots A_o, r, s, i]\omega \Rightarrow \alpha[A_1 \dots A_o B, r, s, i]\omega$. By this way, $\alpha B A_o \dots A_1 [\varepsilon, r, s, i] \Rightarrow \alpha B[A_1 \dots A_o, r, s, i] \Rightarrow \alpha B[A_1 \dots A_o, e, s, i]$, where $B \in T$ or $B \notin N \cup T$ or $o = l$.

Next, $\alpha B[A_1 \dots A_o, e, s, i]\omega \Rightarrow \alpha B[A_1 \dots A_o, e, s', i']\omega$ if $s = s'$, $i = i'$ and $sub(A_1 \dots A_o) \cap N_{left}(P_i) \neq \emptyset$, or $s_i \rightarrow i \in \bar{\bar{\delta}}$.

Now, the automaton M is ready for simulation of next derivation step of Γ.

By this way, push-down automaton can simulate every production of Γ. Proof of this statement is clear, bat excessively long for the scope of this paper.

5 CONCLUSION

In this paper we defined the derivation restriction on a CD grammar system controlled by some control language. If we add some type of restriction or control language to GS, generative power of GS will usually be the same as generative power of origin GS. In spite of components of grammar systems are phrase-structure grammars, GS with this new restriction and with REG control language is able to generate CF languages only.

REFERENCES

[1] B, S, Baker. Context-sensitive grammars generating context-free languages. In M. Nivat, editor, *Automata, Languages and Programming*, pages 501–506. North-Holland, Amsterdam, 1972.

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

[3] J. Dassow, H. Fernau, and Gh. Păun. On the leftmost derivation in matrix grammars. *International Journal of Foundations of Computer Science*, 10(1):61–80, 1999.

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

[5] H. Fernau. Regulated grammars under leftmost derivation. *Grammars*, 3(1):37–62, 2000.

[6] M. Frazier and C. D. Page. Prefix grammars: An alternative characterization of the regular languages. *Information Processing Letters*, 51(2):67–71, 1994.

[7] V. Geffert. Context-free-like forms for the phrase-structure grammars. In *Proceedings of the Mathematical Foundations of Computer Science 1988*, pages 309–317, New York, 1988. Springer-Verlag.

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

[9] T. N. Hibbard. Context-limited grammars. *Journal of the ACM*, 21:446–453, 1974.

[10] Takumi Kasai. An hierarchy between context-free and context-sensitive languages. *J. Comput. Syst. Sci.*, 4(5):492–508, 1970.

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

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

[13] A. Meduna. Matrix grammars under leftmost and rightmost restrictions. In Gh. Păun, editor, *Mathematical Linguistics and Related Topics*, pages 243–257. Romanian Academy of Sciences, Bucharest, 1994.

[14] A. Meduna. On the number of nonterminals in matrix grammars with leftmost derivations. In *New Trends in Formal Languages: Control, Cooperation, and Combinatorics*, pages 27–39. Springer-Verlag, 1997.

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

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