ISSN No. 0976-5697

Volume 2, No. 4, July-August 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Scattered Context Grammars with Priority

Jakub Křoustek*
Department of Information Systems,
Faculty of Information Technology, Brno University of Technology,
Božetěchova 2, Brno 612 66, Czech Republic
ikroustek@fit.vutbr.cz

Stanislav Židek,
Department of Information Systems,
Faculty of Information Technology, Brno University of Technology,
Božetěchova 2, Brno 612 66, Czech Republic
izidek@fit.vutbr.cz

Dušan Kolář,
Department of Information Systems,
Faculty of Information Technology, Brno University of Technology,
Božetěchova 2, Brno 612 66, Czech Republic
kolar@fit.vutbr.cz

Alexander Meduna
Department of Information Systems,
Faculty of Information Technology, Brno University of Technology,
Božetěchova 2, Brno 612 66, Czech Republic
meduna@fit.vutbr.cz

Abstract: The scattered context grammars are based on application of n context-free productions in parallel to generate their sentences. We can find two basic versions of this grammar type – erasing and propagating grammars. Erasing productions are allowed in erasing grammar, while they are prohibited in propagating grammars. In this paper, we present regulated versions of these grammars, where productions are regulated by the production-priority function. The priority function guarantees that productions will be applied whenever it is possible according to their priority. We also provide formal proofs of generative power of these grammars.

Keywords: scattered context grammar, regulation, priority, generative power

I. INTRODUCTION

Scattered context grammars belong to the semiparallel rewriting systems [1], where each production consists of *n* context-free productions that are applied in parallel to the current sentential form.

According to [2], family of languages characterized by erasing scattered context grammars coincide with family of recursively enumerable languages, while the power of propagating scattered context grammars lies between families of context-free and context-sensitive grammars [3]. Definition of the exact position in the Chomsky hierarchy is an open problem.

In some situations, it is important to make sure that the specific production will be used before any other. For example, we need to assure that a blocking production will be used whenever it is possible. Because the process of a production selection is not regulated in any way, it is necessary to include this control into the grammar itself. However, this leads to a significant increase of productions number, which implies more complex and less effective sentence derivation/parsing.

Therefore, it will be useful to specify that some productions have a higher priority in the production selection than others. For this purpose, we introduce new regulated version of scattered context grammars – *scattered context grammars with priority*. With this approach, we are able to create the complete production selection hierarchy or just specify one production which is more important than others (e.g. blocking production). That is the main difference over the other regulation types (see [4], [5]). In the following sections, we will study generative power of erasing and propagating scattered context grammars with priority.

II. PRELIMINARIES

We assume a reader is familiar with the formal language theory (for further reference, see for example [6]).

For an alphabet V, V^* denotes the free monoid generated by V under the operation of concatenation, with the unit element ε. Set $V^+ = V^* - \{\varepsilon\}$. For $w \in V^*$, $|w|$ denotes the length of w and $\mathrm{alph}(w)$ denotes the set of symbols appearing in w. For $U \subseteq V$, $|w|_U$ denotes the number of occurrences of symbols from U in w.

A phrase-structure grammar is a quadruple $G = (V, T, P, S)$, where V is a total alphabet, $T \subset V$ is a finite set of terminal symbols (terminals), $S \in V - T$ is the starting symbol and P is a finite set of productions $p = x \to y$, $x \in V^*(V-T)V^*$, $y \in V^*$. We set $\mathrm{lhs}(p) = x$ and $\mathrm{rhs}(p) = y$, which represents the left-hand side and the right-hand side of the production p, respectively.

A context-sensitive grammar (CSG) is a phrase-structure grammar $G = (V, T, P, S)$, such that every production $p = x \to y \in P$ satisfies $|x| \le |y|$.

Let $G = (V, T, P, S)$ be a CSG, $y = w_1 \alpha w_2$, $z = w_1 \beta w_2$, $y, z \in V^*$, $p = \alpha \to \beta \in P$. Then y directly derives z in the CSG G according to the

© 2010, IJARCS All Rights Reserved

production p, $y \Rightarrow_G z[p]$ (or simply $y \Rightarrow_G z$). Let $\Rightarrow_G^+$ and $\Rightarrow_G^*$ denote the transitive and the reflexive-transitive closure of $\Rightarrow_G$, respectively. To express that G makes the derivation from u to v by using the sequence of productions $p_1, p_2, \ldots, p_n \in P$, we write $u \Rightarrow_G^* v[p_1 p_2 \ldots p_n]$ (or $u \Rightarrow_G^+ v[p_1 p_2 \ldots p_n]$ to emphasize that the sequence is non-empty). The language generated by G is denoted by $L(G)$ and defined as $L(G) = \{w : w \in T, S \Rightarrow_G^* w\}$. A context-sensitive language is language generated by a CSG. The family of context-sensitive languages is denoted by $\mathcal{L}(\mathrm{CS})$.

A scattered context grammar (SCG, see [3]) is a quadruple, $G = (V, T, P, S)$, where V is a total alphabet, $T \subset V$ is a finite set of terminal symbols (terminals; symbols from $V - T$ are called nonterminal symbols or nonterminals), $S \in V - T$ is the starting symbol and P is a finite set of productions of the form $(A_1, \ldots, A_n) \to (x_1, \ldots, x_n)$, where $A_i \in V - T$, $x_i \in V^*$ for all $i : 1 \le i \le n$. For $p = (A_1, \ldots, A_n) \to (x_1, \ldots, x_n) \in P$, $\mathrm{lhs}(p)$ and $\mathrm{rhs}(p)$ denote $A_1 A_2 \ldots A_n$ and $x_1 x_2 \ldots x_n$, respectively. A propagating SCG is a SCG $G = (V, T, P, S)$ in which every $(A_1, \ldots, A_n) \to (x_1, \ldots, x_n) \in P$ satisfies $x_i \in V^+$ for all $i : 1 \le i \le n$.

Let $G = (V, T, P, S)$ be a (propagating) SCG, $y = u_1 A_1 u_2 \ldots u_n A_n u_{n+1}$, $z = u_1 x_1 u_2 \ldots u_n x_n u_{n+1}$, $y, z \in V^*$, $p = (A_1, \ldots, A_n) \to (x_1, \ldots, x_n) \in P$. Then y directly derives z in the SCG G according to the production p, $y \Rightarrow_G z[p]$ (or simply $y \Rightarrow_G z$). Let $\Rightarrow_G^+$ and $\Rightarrow_G^*$ denote the transitive and the reflexive-transitive closure of $\Rightarrow_G$, respectively. To express that G makes the derivation from u to v by using the sequence of productions $p_1, p_2, \ldots, p_n \in P$, we write $u \Rightarrow_G^* v[p_1 p_2 \ldots p_n]$ (or $u \Rightarrow_G^+ v[p_1 p_2 \ldots p_n]$ to emphasize that the sequence is non-empty). The language generated by G is denoted by $L(G)$ and defined as $L(G) = \{w : w \in T, S \Rightarrow_G^* w\}$. A (propagating) scattered context language is language generated by (P)SCG. The family of (propagating) scattered context languages is denoted by $\mathcal{L}((\mathrm{P})\mathrm{SC})$.

We abbreviate $\Rightarrow_G$ to $\Rightarrow$ when it is clear which grammar we are referring to.

III. SCATTERED CONTEXT GRAMMARS WITH PRIORITY

Definition 1. A (propagating) scattered context grammar with priority ((P)SCGP) is a quintuple, $G = (V, T, P, S, \pi)$, where (V, T, P, S) is a (propagating) scattered context grammar and π is a function, $\pi : P \to \mathbb{N}$. A (propagating) scattered context language with priority is language generated by a (propagating) scattered context grammar with priority. The family of (propagating) scattered context languages is denoted by $\mathcal{L}((\mathrm{P})\mathrm{SCP})$.

Definition 2. Let $G = (V, T, P, S, \pi)$ be a (P)SCGP. We say that y directly derives z in (P)SCG G according to the production p, $y \Rightarrow_G z[p]$ (or simply $y \Rightarrow_G z$), if and only if:

- $y = u_1 A_1 u_2 \ldots u_n \quad A_n u_{n+1} \in V^*$,
- $z = u_1 x_1 u_2 \ldots u_n x_n u_{n+1} \in V^*$,
- $p = (A_1, \ldots, A_n) \to (x_1, \ldots, x_n) \in P$, and
- there is no $p' = (A_{1'}, \ldots, A_{n'}) \to (x_{1'}, \ldots, x_{n'}) \in P$, such that:
  1. $y = u_{1'} A_{1'} u_{2'} \ldots u_{n'} A_{n'} u_{n+1}' \in V^*$, and
  2. $\pi(p') > \pi(p)$.

Let $\Rightarrow_G^+$ and $\Rightarrow_G^*$ denote the transitive and the reflexive-transitive closure of $\Rightarrow_G$, respectively.

Lemma 1.

$$\mathcal{L}(RE) = \mathcal{L}(SCP)$$

Proof. The proof is trivial. We use the theorem $\mathcal{L}(RE) = \mathcal{L}(SC)$ [2]. For every SCG $G_{\mathrm{SC}} = (V, T, P, S)$, there exists a SCGP $G = (V, T, P, S, \pi)$, such that for every $p \in P : \pi(p) = 0$, and $L(G_{\mathrm{SC}}) = L(G)$. Therefore, $\mathcal{L}(SC) \subseteq \mathcal{L}(SCP)$.

■

Lemma 2.

$$\mathcal{L}(CS) \subseteq \mathcal{L}(PSCP)$$

Proof. The proof is inspired by an another type of regulated grammar – the *ordered scattered context grammar* [4]. For every context-sensitive grammar $G_{\mathrm{CS}} = (V_{\mathrm{CS}}, T_{\mathrm{CS}}, P_{\mathrm{CS}}, S)$, there exists a PSCGP $G = (V, T, P, S_0, \pi)$, such that $L(G_{\mathrm{CS}}) = L(G)$.

Assume that G_{CS} is in Pentonnen normal form for context-sensitive grammars (see [7]). Let $T = T_{\mathrm{CS}}$, $N = V_{\mathrm{CS}} - T_{\mathrm{CS}}$, $N' = \{A' : A \in N\}$, $\Phi_0 = \{A_0 : A \in N\}$, $\Phi_1 = \{A_1 : A \in N \cup N'\}$, $\Phi_2 = \{A_2 : A \in N\}$, $\Phi = \Phi_0 \cup \Phi_1 \cup \Phi_2$, $\Phi' = N' \cup \{A_{1'} : A \in N\}$, $V = T \cup \Phi \cup \Phi' \cup \{\perp\}$.

Without loss of generality, we assume that N, N', Φ_0, Φ_1, and Φ_2 are pairwise disjoint.

Define P and π as follows:

© 2010, IJARCS All Rights Reserved

- For every production $q = A \rightarrow BC \in P_{\text{CS}}$ and $X \in V_{\text{CS}} - T_{\text{CS}}$, let
  - o $p_{0a}\langle q, X\rangle = (X_0, A) \rightarrow (X_0, BC) \in \Xi_0$, and
  - o $p_{0b}\langle q\rangle = (A_0) \rightarrow (B_0 C) \in \Xi_0$.

  Add contents of Ξ_0 to P.
- For every production $q = AB \rightarrow AC \in P_{\text{CS}}$, $X \in V_{\text{CS}} - T_{\text{CS}}$, and $Y \in V - T$, let
  - o $p_{1a}\langle q, X\rangle = (X_0, A, B) \rightarrow (X_1, A', B') \in \Xi_1$,
  - o $p_{1a}'\langle q, X\rangle = (X_1, A', B') \rightarrow (X_0, A, C) \in \Xi_1$,
  - o $p_{1b}\langle q, X\rangle = (A_0, B) \rightarrow (A'_1, B') \in \Xi_1$, and
  - o $p_{1b}'\langle q, X\rangle = (A'_1, B') \rightarrow (A_0, C) \in \Xi_1$.

  And let
  - o $p_{\perp a}\langle q, X, Y\rangle = (X_1, A', Y, B') \rightarrow (\perp, \perp, \perp, \perp) \in \Xi_\perp$, and
  - o $p_{\perp b}\langle q, Y\rangle = (A'_1, Y, B') \rightarrow (\perp, \perp, \perp) \in \Xi_\perp$.

  Add contents of Ξ_1 and $\Xi_\perp$ to P.
- For every $X \in V_{\text{CS}} - T_{\text{CS}}$, add $(X_0) \rightarrow (X_2)$ to P.
- For every production $q = A \rightarrow a \in P_{\text{CS}}$, let
  - o $p_{2a}\langle q, X\rangle = (X_2, A) \rightarrow (X_2, a) \in \Xi_2$, and
  - o $p_{2b}\langle q\rangle = (A_2) \rightarrow (a) \in \Xi_2$.

  Add contents of Ξ_2 to P.

Define function $\pi : P \rightarrow \{0,1\}$ as follows:

$$\pi(p) = \begin{cases} 1 & if \;\; p \in \Xi_\perp \\ 0 & otherwise \end{cases}$$

Proof Idea

We will demonstrate that G simulates every derivation $S \Rightarrow_{G_{\text{CS}}} w$ of G_{CS} in two phases – first it simulates the application of productions of the form $A \rightarrow BC$ and $AB \rightarrow AC$ (i.e. without terminals) and then it rewrites all nonterminals to terminals (i. e. simulating productions of the form $A \rightarrow a$); more precisely, every successful derivation of G proceeds as follows:

$$\begin{array}{llll} S_0 & \Rightarrow_G^* & X_0 v_1 & [\rho] \\ & \Rightarrow_G & X_2 v_1 & [X_0 \rightarrow X_2] \\ & \Rightarrow_G^* & X_2 v_2 & [\sigma] \\ & \Rightarrow_G & w & [X_2 \rightarrow a] \;, \end{array}$$

where $X_0, X_2 \in (V - T)$, $v_1 \in (V - T)^*$, $v_2, w \in T^*$, $\rho \in (\Xi_0 \cup \Xi_1)^*$, $\sigma \in \Xi_2^*$, and $a \in T$. In the first phase, productions of the form $A \rightarrow BC$ are simulated when the zero index occurs at the first nonterminal of a sentential form by simply rewriting corresponding nonterminal A to BC or A_0 to $B_0 C$, respectively. Simulation of context-sensitive productions ($AB \rightarrow AC$) proceeds in two steps; first, nonterminals being rewritten are marked with apostrophes (A' and B'), then they are rewritten to nonterminals on the right-hand side of the simulated production (A and C); the priority is very important in this step, because productions with priority 1 (from $\Xi_\perp$) guarantee that rewriting of A', B' to A, C *cannot* happen when there is some other nonterminal symbol between them – in such a case, the derivation is blocked by $\perp$ nonterminals, which cannot be rewritten any further.

Note the index at the first nonterminal of the sentential form. We use it to keep the state of the derivation, 0 denoting the normal mode of simulating productions that do not rewrite symbols to terminals, 1 denoting the auxiliary step needed for the simulation of context-sensitive production and 2 denoting the final phase of rewriting all nonterminal symbols to terminals.

Formal proof

We will establish the Lemma 2 by Claim 1 through Claim 3.

Claim 1. G generates every $w \in L(G)$ in the following way:

$$\begin{array}{llll} S_0 & \Rightarrow_G^* & w_1 & [\rho] \\ & \Rightarrow_G & w_2 & [X_0 \rightarrow X_2] \\ & \Rightarrow_G^* & w_3 & [\sigma] \\ & \Rightarrow_G & w & [X_2 \rightarrow a] \;, \end{array}$$

where $w_1, w_2, w_3 \in V^*$, $\rho \in \{\Xi_0 \cup \Xi_1\}^*$, $\sigma \in \Xi_2^*$.

Proof. First, let us make these observations:

- For every $u \in (V - T)^*$, such that $S_0 \Rightarrow_G^* u \Rightarrow_G^+ w$,
  $$u \in \Phi(V - \Phi)^*.$$
  This directly follows from the facts that for every $p \in P$,
  $$1 = |\text{lhs}(p)|_{\Phi_{0,1,2}} \geq |\text{rhs}(p)|_{\Phi_{0,1,2}}$$
  (where $\Phi_{0,1,2} = \Phi_0 \cup \Phi_1 \cup \Phi_2$) and that the single nonterminal from $\Phi_{0,1,2}$ is kept as the first symbol of the sentential form through the whole derivation.
- There is no production $p \in P$, such that $|\,\text{lhs}(p)|_{\Phi_2} = 1$ and $|\,\text{rhs}(p)|_{\Phi_0} = 1$, so once the production in the form $X_0 \rightarrow X_2$ is used, only productions from Ξ_2 remain applicable.
- It holds that $w_3 \in \Phi_2 N^*$. The only way to successfully terminate the derivation is by using production of the form $X_2 \rightarrow a$, where $X_2 \in \Xi_2$ and $a \in T$.

We see Claim 1 holds.

■

© 2010, IJARCS All Rights Reserved

Claim 2. Consider derivation introduced in Claim 1 and a derivation $S \Rightarrow^*_{G_{\mathrm{CS}}} r$, $r \in (V_{\mathrm{CS}} - T_{\mathrm{CS}})^*$. For every $s \in (V - T - \Phi')^*$,

$$S_0 \Rightarrow^*_G s \Rightarrow^*_G w_1 \text{ iff } S \Rightarrow^*_{G_{\mathrm{CS}}} r,$$

such that $s = (A_1)_0 A_2 \dots A_m, r = A_1 A_2 \dots A_m, (A_1)_0 \in \Phi_0$, $A_1, \dots, A_m \in V_{\mathrm{CS}} - T_{\mathrm{CS}}$.

Proof. There are two types of productions in G_{CS} that may be applied without introducing terminal symbol into the sentential form:

1. *Context-free productions* (of the form $A \to BC$) are simulated by the productions from Ξ_0 in a very straightforward way.
2. *Context-sensitive productions* (of the form $AB \to AC$) are simulated in two phases by the productions from Ξ_1 with higher priority productions in $\Xi_\perp$ preventing the rewriting of non-adjacent nonterminals. More precisely, application of the production $AB \to AC$ is simulated by derivation

$$\begin{aligned} & (A_1)_0 A_2 \dots AB \dots A_m \\ \Rightarrow_G \quad & (A_1)_1 A_2 \dots A'B' \dots A_m \\ \Rightarrow_G \quad & (A_1)_0 A_2 \dots AC \dots A_m \end{aligned}$$

where $A_2, \dots A_m, A, B, C \in V_{\mathrm{CS}} - T_{\mathrm{CS}}$, $A', B' \in \Phi'$, $(A_1)_0 \in \Phi_0$ and $(A_1)_1 \in \Phi_1$. The second step is possible only when there is no other nonterminal between rewritten nonterminals.

Note: We considered only the case when the first nonterminal in sentential form is not rewritten. Simulation of rewriting the first nonterminal by context-sensitive production is analogical.

■

Claim 3. Consider derivation introduced in Claim 1. It holds that

$$w_2 \in \Phi_2 (V_{\mathrm{CS}} - T_{\mathrm{CS}})^*$$

and

$$w_3 \in \Phi_2 (T_{\mathrm{CS}})^*,$$

such that $w_2 = (A_1)_2 A_2 \dots A_m$, $w_3 = (A_1)_2 a_2 \dots a_m$, and $A_i \to a_i \in P_{\mathrm{CS}}$ for every $i : 2 \le i \le m$.

Proof. Observe that for every $p \in P$, if $|\,\mathrm{lhs}(p)\,|_{\Xi_2} \neq 0$, then $|\,\mathrm{rhs}(p)\,|_{\Xi_0 \cup \Xi_1} = 0$. Therefore, only productions from Ξ_2 can be applied in this phase. Furthermore, the production $p_{2b}\langle q \rangle$, $q \in P_{\mathrm{CS}}$, cannot be applied before the last step, because there would be no way to rewrite remaining nonterminals.

■

From Claim 1 through 3, we see Lemma 2 holds.

■

Lemma 3.

$$\mathcal{L}(PSCP) \subseteq \mathcal{L}(CS)$$

Proof. Let $G_{\mathrm{PSCP}} = (V_{\mathrm{PSCP}}, T_{\mathrm{PSCP}}, P_{\mathrm{PSCP}}, S_{\mathrm{PSCP}}, \pi)$ be a PSCGP. Then, there exists a context-sensitive grammar G such that $L(G) = L(G_{\mathrm{PSCP}})$. Set (for description of auxiliary sets, see the following proof idea)

$\Phi_R = \{R^{p_i}_{A_i} : p = (A_1, \dots, A_i, \dots, A_n) \to (x_1, \dots, x_i, \dots, x_n) \in P_{\mathrm{PSCP}}\} \cup \{R_T\}$,

$\Phi_L = \{L^{p_i}_{A_i} : p = (A_1, \dots, A_i, \dots, A_n) \to (x_1, \dots, x_i, \dots, x_n) \in P_{\mathrm{PSCP}}\} \cup \{L_T, L_F\}$,

$\Phi = \{\langle a \rangle : a \in V_{\mathrm{PSCP}}\} \cup \{\langle a \mid X \rangle : a \in V_{\mathrm{PSCP}}, X \in \Phi_R \cup \Phi_L\}$,

$\Phi_\triangleleft = \{\langle \triangleleft M a \rangle, \langle \blacktriangleleft M a \rangle : a \in V_{\mathrm{PSCP}}, M \subseteq P_{\mathrm{PSCP}}\} \cup \{\langle \triangleleft M a \mid X \rangle, \langle \blacktriangleleft M a \mid X \rangle : a \in V_{\mathrm{PSCP}}, M \subseteq P_{\mathrm{PSCP}}, X \in \Phi_R \cup \Phi_L\}$,

$\Phi_\triangleright = \{\langle a \triangleright \rangle : a \in V_{\mathrm{PSCP}}\} \cup \{\langle a \triangleright \mid X \rangle : a \in V_{\mathrm{PSCP}}, X \in \Phi_R \cup \Phi_L\}$,

$\Phi_{\triangleleft\triangleright} = \{\langle \triangleleft\, a \triangleright \rangle : a \in V_{\mathrm{PSCP}}\}$, $\Phi_\bullet = \{\langle \bullet a \rangle : a \in T_{\mathrm{PSCP}}\}$.

Define the grammar $G = (V, T, P, S)$, where $T = T_{\mathrm{PSCP}}$, $V = \Phi \cup \Phi_\triangleleft \cup \Phi_\triangleright \cup \Phi_{\triangleleft\triangleright} \cup T$, $S_{\triangleleft\triangleright} \in V - T$.

Define the P as follows (description is emphasized for better understanding):

1. *Sentential form of length one.*
   For each $A \in V_{\mathrm{PSCP}} - T_{\mathrm{PSCP}}$ and $p = (A) \to (x)$, $x = a_1 a_2 \dots a_n$ such that there is no $q = (A) \to (y) \in P_{\mathrm{PSCP}}$ with $\pi(q) > \pi(p)$, add production $\langle \triangleleft A \triangleright \rangle \to z$ to P, where

$$z = \begin{cases} \varepsilon & \text{if } |x| = 0 \\ \langle \triangleleft\, x \triangleright \rangle & \text{if } |x| = 1 \\ \langle \triangleleft\, a_1 \rangle \langle a_2 \rangle \dots \langle a_{n-1} \rangle \langle a_n \triangleright \rangle & \text{if } |x| = n > 1. \end{cases}$$

2. *Sentential form of length two.*
   For each $a, b \in V_{\mathrm{PSCP}}$:
   (a) *Rewriting the first nonterminal.*
   For each $(a) \to (a_1 a_2 \dots a_m) \in P_{\mathrm{PSCP}}$, such that there is no $(a,b) \to (x,y)$, or $(b) \to (y')$ in P_{PSCP} with higher priority, add production $\langle \triangleleft \emptyset\, a \rangle \langle b \triangleright \rangle \to \langle \triangleleft \emptyset\, a_1 \rangle \dots \langle a_m \rangle \langle b \triangleright \rangle$ to P.
   (b) *Rewriting the second nonterminal.*
   For each $(b) \to (b_1 b_2 \dots b_m) \in P_{\mathrm{PSCP}}$, such that there is no $(a,b) \to (x,y)$, or $(a) \to (x')$ in P_{PSCP} with higher priority, add production $\langle \triangleleft \emptyset\, a \rangle \langle b \triangleright \rangle \to \langle \triangleleft \emptyset\, a \rangle \langle b_1 \rangle \dots \langle b_m \triangleright \rangle$ to P.
   (c) *Rewriting both nonterminals.*
   For each $(a,b) \to (a_1 a_2 \dots a_{m_1}, b_1 b_2 \dots b_{m_2}) \in P_{\mathrm{PSCP}}$, such that there is no $(a,b) \to (x,y)$, $(a) \to (x')$

© 2010, IJARCS All Rights Reserved

or $(b) \to (y')$ in P_{PSCP} with higher priority, add following production to P:

$\langle \lhd \emptyset\ a\rangle\langle b\ \emptyset \rhd\rangle \to \langle \lhd a_1\rangle \ldots \langle a_{m_1}\rangle\langle b_1\rangle \ldots \langle b_{m_2} \rhd\rangle$

3. *Sentential form of length higher than two.*

For each $M \subseteq P_{\text{PSCP}}$,

$p = (A_1, \ldots, A_n) \to (x_1, \ldots, x_n) \in P_{\text{PSCP}}$, $i : 1 \le i < n$, $j : 1 < j \le n$, $a, a', b, c \in V_{\text{PSCP}}$, and $b \ne A_1$, $c \ne A_i$, $c \ne A_j$, add following productions to P:

(a) *Checking phase:*

i. *Start the application of production* p.

$\langle \lhd M\ a\rangle \to \langle \blacktriangleleft\ M\ a \mid R_{A_1}^{p_1}\rangle$ if there is no $q \in P_{\text{PSCP}} - M$, such that $\pi(q) > \pi(p)$.

ii. *First step, the searched nonterminal at the first position.*

$\langle \blacktriangleleft Mb \mid R_{A_1}^{p_1}\rangle\langle a\rangle \to \langle \blacktriangleleft Mb\rangle\langle a \mid R_{A_1}^{p_1}\rangle$.

iii. *First step, other nonterminal at the first position.*

$\langle \blacktriangleleft MA_1 \mid R_{A_1}^{p_1}\rangle\langle a\rangle \to \langle \blacktriangleleft MA_1\rangle\langle a \mid R_{A_2}^{p_2}\rangle$.

iv. *Move right, nonterminal not found.*

$\langle c \mid R_{A_i}^{p_i}\rangle\langle a\rangle \to \langle c\rangle\langle a \mid R_{A_i}^{p_i}\rangle$.

v. *Move right, nonterminal found.*

$\langle A_i \mid R_{A_i}^{p_i}\rangle\langle a\rangle \to \langle A_i\rangle\langle a \mid R_{A_{i+1}}^{p_{i+1}}\rangle$.

vi. *Move right, last nonterminal found.*

$\langle A_n \mid R_{A_n}^{p_n}\rangle\langle a\rangle \to \langle A_n\rangle\langle a \mid R_T\rangle$.

vii. *Move right to the last position, nonterminal not found.*

$\langle c \mid R_{A_i}^{p_i}\rangle\langle a \rhd\rangle \to \langle c\rangle\langle a \mid R_{A_i}^{p_i} \rhd\rangle$,

viii. Move right to the last position, nonterminal found.

$\langle A_i \mid R_{A_i}^{p_i}\rangle\langle a \rhd\rangle \to \langle A_i\rangle\langle a \mid R_{A_{i+1}}^{p_{i+1}} \rhd\rangle$.

ix. *Move right to the last position, last nonterminal found.*

$\langle A_n \mid R_{A_n}^{p_n}\rangle\langle a \rhd\rangle \to \langle A_n\rangle\langle a \mid R_T \rhd\rangle$.

(b) *End of check:*

i. *Start returning, production cannot be applied.*

$\langle a\rangle\langle a' \mid R_{A_i}^{p_i} \rhd\rangle \to \langle a \mid L_F^p\rangle\langle a' \rhd\rangle$.

ii. *Start returning, production can be applied.*

$\langle a\rangle\langle a' \mid R_T \rhd\rangle \to \langle a \mid L_{A_n}^{p_n}\rangle\langle a' \rhd\rangle$.

iii. *Start returning, production can be applied, with applying to the last nonterminal.*

$\langle a\rangle\langle A_n \mid R_T \rhd\rangle \to \langle a \mid L_{A_{n-1}}^{p_{n-1}}\rangle\langle x_{n_1}\rangle\langle x_{n_2}\rangle \ldots \langle x_{n_{m_n}} \rhd\rangle$.

iv. *Start returning, nonterminal found at the last position, with applying.*

$\langle a\rangle\langle A_n \mid R_{A_n}^{p_n} \rhd\rangle \to \langle a \mid L_{A_{n-1}}^{p_{n-1}}\rangle\langle x_{n_1}\rangle\langle x_{n_2}\rangle \ldots \langle x_{n_{m_n}} \rhd\rangle$.

v. *Start returning, nonterminal found at the last position, without applying.*

$\langle a\rangle\langle A_n \mid R_{A_n}^{p_n} \rhd\rangle \to \langle a \mid L_{A_n}^{p_n}\rangle\langle A_n \rhd\rangle$.

(c) *Going left, possibly applying:*

i. *Move left, without applying.*

$\langle a\rangle\langle a' \mid L_{A_j}^{p_j}\rangle \to \langle a \mid L_{A_j}^{p_j}\rangle\langle a'\rangle$.

ii. *Move left, with applying.*

$\langle a\rangle\langle A_j \mid L_{A_j}^{p_j}\rangle \to \langle a \mid L_{A_{j-1}}^{p_{j-1}}\rangle\langle x_{j_1}\rangle\langle x_{j_2}\rangle \ldots \langle x_{j_{m_j}}\rangle$.

iii. *Move left to the first position, without applying.*

$\langle \blacktriangleleft Ma\rangle\langle a' \mid L_{A_j}^{p_j}\rangle \to \langle \blacktriangleleft Ma \mid L_{A_j}^{p_j}\rangle\langle a'\rangle$.

iv. *Move left to the first position, with applying.*

$\langle \blacktriangleleft Ma\rangle\langle A_j \mid L_{A_j}^{p_j}\rangle \to \langle \blacktriangleleft Ma \mid L_{A_{j-1}}^{p_{j-1}}\rangle\langle x_{j_1}\rangle\langle x_{j_2}\rangle \ldots \langle x_{j_{m_j}}\rangle$.

v. *Move left, whole production applied.*

$\langle a\rangle\langle A_1 \mid L_{A_1}^{p_1}\rangle \to \langle a \mid L_T^{p_1}\rangle\langle x_{1_1}\rangle\langle x_{1_2}\rangle \ldots \langle x_{1_{m_1}}\rangle$.

vi. *Move left to the first position, whole production applied.*

$\langle \blacktriangleleft Ma\rangle\langle A_1 \mid L_{A_1}^{p_1}\rangle \to \langle \blacktriangleleft Ma \mid L_T^{p_1}\rangle\langle x_{1_1}\rangle\langle x_{1_2}\rangle \ldots \langle x_{1_{m_1}}\rangle$.

vii. *Move left, production applied.*

$\langle a\rangle\langle a' \mid L_T^p\rangle \to \langle a \mid L_T^p\rangle\langle a'\rangle$.

viii. *Move left to the first position, production applied.*

$\langle \blacktriangleleft Ma\rangle\langle a' \mid L_T^p\rangle \to \langle \blacktriangleleft Ma \mid L_T^p\rangle\langle a'\rangle$.

ix. *Move left, production cannot be applied.*

$\langle a\rangle\langle a' \mid L_F^p\rangle \to \langle a \mid L_F^p\rangle\langle a'\rangle$.

x. *Move left to the first position, production cannot beapplied.*

$\langle \blacktriangleleft Ma\rangle\langle a' \mid L_F^p\rangle \to \langle \blacktriangleleft Ma \mid L_F^p\rangle\langle a'\rangle$.

xi. Production applied.

$\langle \blacktriangleleft Ma \mid L_T^p\rangle \to \langle \lhd \emptyset\ a\rangle$.

xii. *Production applied, the first nonterminal at the first position.*

$\langle \blacktriangleleft MA_1 \mid L_{A_1}^{p_1}\rangle \to \langle \lhd \emptyset x_{1_1}\rangle\langle x_{1_2}\rangle \ldots \langle x_{1_{m_1}}\rangle$.

xiii. *Production cannot be applied.*

$\langle \blacktriangleleft Ma \mid L_F^p\rangle \to \langle \lhd (M \cup \{p\})\ a\rangle$.

4. *Final steps, rewriting to terminals.*

For each $t, t' \in T$, add productions

(a) $\langle \lhd \emptyset t \rhd\rangle \to t$,

(b) $\langle \lhd \emptyset t\rangle \to \langle \bullet t\rangle$,

(c) $\langle \bullet t\rangle\langle t'\rangle \to t\langle \bullet t'\rangle$,

(d) $\langle \bullet t\rangle\langle t' \rhd\rangle \to tt'$ to P.

© 2010, IJARCS All Rights Reserved

As we can see, a construction of necessary context-sensitive grammar is quite complex, so we will concentrate on the basic idea behind it and we will omit the formal proof, which would be very tedious.

Proof Idea

The constructed grammar simulates the application of productions of G_{PSCP}. It uses specific nonterminals for storing the needed additional information of finite nature. For the sake of clarity, all nonterminals of V are enclosed in angle brackets. The auxiliary sets Φ, $\Phi_{\triangleleft}$, $\Phi_{\triangleright}$, $\Phi_{\triangleleft\triangleright}$, $\Phi_{\bullet}$ contain special nonterminals that comprise the nonterminal of grammar G_{PSCP} with additional information (attribute, beginning, end mark, etc.). Symbol $\triangleleft$ marks the first nonterminal before starting the applicability check, $\blacktriangleleft$ also marks the first nonterminal when the check is in progress; when both symbols are present, the sentential form contains just one symbol. Also note that the first nonterminal (in a case of more than two symbols in sentential form) contains the set of productions of original grammar G_{PSCP} whose application we cannot simulate.

It always tries to check (and then possibly apply) productions of P_{PSCP}, one by one from the highest priority to the lowest, i.e., only after checking that all productions with higher priority cannot be used, it is possible to start checking productions with lower priority (see production descriptions).

Checking if given scattered context production can be applied is done via auxiliary nonterminal "attribute" R, which goes through the sentential form and checks the applicability of the individual context-free parts of scattered context productions. See that we must be absolutely sure whether the production could be simulated, otherwise it would be possible to skip higher priority productions and therefore use lower priority production, which is not possible with PSCGP.

After the check, there are two situations. Either the production is applicable, in which case it must be applied, or it is not applicable and we are free to check other productions. Anyway, the attribute L is used to go through the sentential form back to the beginning, possibly apply the production (variants with current nonterminal to be rewritten in index) or bring the negative check result (denoted by F index) to the "attribute set" in the first nonterminal.

Note the basic difference in two phases of going through the sentential form. While we are checking if the production is applicable (going right), it is mandatory to change attribute $L_{A_i}^{p_i}$ to $L_{A_{i+1}}^{pi+1}$ when we encounter the nonterminal $\langle A_i \rangle$. When we are applying the production after the successful check (going left), rewriting the nonterminal is *optional*, i.e. it is possible to skip the nonterminal we need to rewrite without the rewriting. This behavior is necessary in order to being able to rewrite other than last occurrence of particular nonterminal. ■

Theorem 1.

$$\mathcal{L}(PSCP) = \mathcal{L}(CS)$$

Proof. Directly follows from Lemma 2 and Lemma 3. ■

IV. CONCLUSION

In this paper, we have introduced new variant of the SCG which is regulated by the priority function. The priority function affects the production selection, where a production can be used only if it is applicable in the current sentence and if there is no other applicable production with the higher priority. It is possible to regulate both erasing and propagating SCG.

Formal proofs of generative power of those regulated grammars has been presented too. It is obvious that family of languages generated by the SCGP is equivalent to family of recursively enumerable languages, and that family of languages generated by the PSCGP is equivalent to family of context-sensitive languages.

ACKNOWLEDGMENT

This work was supported by the research fundings MPO ČR, No. FR-TI1/038, TAČR, No. TA01010667, BUT FIT grant FIT-S-11-2, by the Research Plan No. MSM 0021630528, and by the SMECY European project.

REFERENCES

[1] G. Rozenberg, A. Salomaa, "Handbook of Formal Languages Volume 1–3," Springer, Berlin, 1997.

[2] A. Meduna, J. Techet, "Scattered Context Grammars and Their Applications," WIT Press, 2009, pp. 137–155.

[3] S. Greibach, J. Hopcroft, "Scattered context grammars," Journal of Computer and System Sciences, vol. 3, 1969, pp. 233–247.

[4] J. Dassow, G. Paun, "Regulated Rewriting in Formal Language Theory," EATCS Monographs in Theoretical Computer Science, vol. 18, Springer-Verlag, Berlin, 1989.

[5] H. Fernau, "Scattered context grammars with regulation," Annals of Bucharest University, Mathematics-Informatics Series, vol. 45(1), 1996, pp. 41–49.

[6] A. Meduna, "Automata and Languages: Theory and Applications," Springer-Verlag, 2000.

[7] M. Pentonnen, "One-Sided and Two-Sided Context in Formal Grammars," Information and Control, vol. 25, 1974, pp. 371–392.

© 2010, IJARCS All Rights Reserved