Acta Cybernetica **17** (2005) 11–20.

Generation of Sentences with Their Parses: the Case of Propagating Scattered Context Grammars

Alexander Meduna* and Jiří Techet*

Abstract

Propagating scattered context grammars are used to generate their sentences together with their parses—that is, the sequences of labels denoting productions whose use lead to the generation of the corresponding sentences. It is proved that for every recursively enumerable language L, there exists a propagating scattered context grammar whose language consists of L's sentences followed by their parses.

Keywords: parsing, propagating scattered context grammars

1 Introduction

Parallel parsing represents a vivid investigation area concerning compilers today (see [1, 2, 9, 10, 16]). As parsing is almost always based on suitable grammatical models, parallel grammars are important to this area. Since scattered context grammars generate their languages in a parallel way, their use related to parsing surely deserves our attention.

In this paper, we use the propagating scattered context grammars, which contain no erasing productions, to generate their language's sentences together with their parses—that is, the sequences of labels denoting productions whose use lead to the generation of the corresponding sentences (in the literature, derivations words and Szilard words are synonymous with parses). We demonstrate that for every recursively enumerable language L, there exists a propagating scattered context grammar whose language consists of L's sentences followed by their parses. That is, if we eliminate all the suffixes representing the parses, we obtain precisely L. This characterization of recursively enumerable languages is of some interest because it is based on propagating scattered context grammars whose languages are included in the family of context-sensitive languages, which is properly contained in the family of recursively enumerable languages. Simply stated, in this paper, we use the propagating scattered context grammars in such a way that this use provides us with the parses corresponding to the generated sentences and, in addition, increases the generative power of these grammars.

*Department of Information Systems, Faculty of Information Technology, Brno University of Technology, Božetěchova 2, Brno 61266, Czech Republic

2 Preliminaries

We assume that the reader is familiar with the language theory (see [6, 11, 12, 13]). For an alphabet V, $card(V)$ denotes the cardinality of V. V^* represents the free monoid generated by V under the operation of concatenation. The unit of V^* is denoted by ϵ. Set $V^+ = V^* - \{\epsilon\}$. For $w \in V^*$, $|w|$ and $rev(w)$ denote the length of w and the reversal of w, respectively. For $U \subseteq V$, $occur(w, U)$ denotes the number of occurrences of symbols from U in w. For $L \subseteq V^*$, $alph(L)$ denotes the set of symbols appearing in a word of L. Let L_1, L_2 be two languages. The *right quotient* of L_1 with respect to L_2, denoted by L_1/L_2, is defined as $L_1/L_2 = \{y \,|\, yx \in L_1, \text{for some } x \in L_2, y \in alph(L_1)^*\}$. The *left quotient* of L_1 with respect to L_2, denoted by $L_2\backslash L_1$, is defined as $L_2\backslash L_1 = \{y \,|\, xy \in L_1, \text{for some } x \in L_2, y \in alph(L_1)^*\}$.

A *scattered context grammar* (see [3, 4, 5, 7, 8, 14, 15] and pages 259–260 in [13]), a *SCG* for short, is a quadruple, $G = (V, P, S, T)$, where V is an alphabet, $T \subseteq V$, $S \in V - T$, and P is a finite set of productions such that each production has the form $(A_1, \ldots, A_n) \to (x_1, \ldots, x_n)$, for some $n \geq 1$, where $A_i \in V - T$, $x_i \in V^*$, for $1 \leq i \leq n$. If every $(A_1, \ldots, A_n) \to (x_1, \ldots, x_n) \in P$ satisfies $x_i \in V^+$ for all $1 \leq i \leq n$, G is a *propagating scattered context grammar*, a *PSCG* for short. If $(A_1, \ldots, A_n) \to (x_1, \ldots, x_n) \in P$, $u = u_1A_1u_2 \ldots u_nA_nu_{n+1}$, and $v = u_1x_1u_2 \ldots u_nx_nu_{n+1}$, where $u_i \in V^*$, $1 \leq i \leq n$, then $u \Rightarrow v\,[(A_1, \ldots, A_n) \to (x_1, \ldots, x_n)]$ 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 $L(G)$ and defined as $L(G) = \{x \,|\, x \in T^*, S \Rightarrow^* x\}$.

3 Definitions and examples

Throughout this paper, we assume that for every *SCG* $G = (V, P, S, T)$, there is a set of production labels denoted by $lab(G)$ such that $card(lab(G)) = card(P)$; as usual, $lab(G)^*$ denotes the set of all strings over $lab(G)$. Let us label each production in P uniquely with a label from $lab(G)$ so that this labeling represents a bijection from $lab(G)$ to P. To express that $p \in lab(G)$ labels a production $(A_1, \ldots, A_n) \to (x_1, \ldots, x_n)$, we write $p : (A_1, \ldots, A_n) \to (x_1, \ldots, x_n)$. For every $p : (A_1, \ldots, A_n) \to (x_1, \ldots, x_n) \in P$, $lhs(p)$ and $rhs(p)$ denote $A_1A_2 \ldots A_n$ and $x_1x_2 \ldots x_n$, respectively. Furthermore, ${}_lpos(p, j)$ and ${}_rpos(p, j)$ denote A_j and x_j, respectively. To express that G makes $x \Rightarrow^* y$ by using a sequence of productions labeled by $p_1, p_2, \ldots, p_n$, we write $x \Rightarrow^* y\,[\rho]$, where $x, y \in V^*$, $\rho = p_1 \ldots p_n \in lab(G)^*$. Let $S \Rightarrow^* x\,[\rho]$ in G, where $x \in T^*$ and $\rho \in lab(G)^*$; then, x is a *sentence generated by* G *according to parse* ρ. Let $G = (V, P, S, T)$ be a *SCG* with $lab(G) \subseteq T$. G is a *proper generator of its sentences with their parses* if $L(G) = \{x \,|\, x = y\rho, y \in (T - lab(G))^*, \rho \in lab(G)^*, S \Rightarrow^* x\,[\rho]\}$.

Next, we illustrate these definitions by three *SCG*s, each of which has its set of production labels equal to $\{1, 2, 3, 4\}$. First, consider *SCG* $G_1 = (\{S, A, B, C, a, b, c\}, P_1, S, \{a, b, c\})$ with P_1 containing $1 : (S) \to (\epsilon)$, $2 : (S) \to$

(ABC), $3 : (A, B, C) \rightarrow (aA, bB, cC)$, $4 : (A, B, C) \rightarrow (a, b, c)$. As $\{1, 2, 3, 4\} \not\subseteq \{a, b, c\}$, G_1 is no proper generator of its sentences with their parses. Second, consider $G_2 = (\{S, A, B, C, a, b, c, 1, 2, 3, 4\}, P_2, S, \{a, b, c, 1, 2, 3, 4\})$ with P_2 containing $1 : (S) \rightarrow (1)$, $2 : (S) \rightarrow (ABC2)$, $3 : (A, B, C) \rightarrow (aA, bB, cC3)$, $4 : (A, B, C) \rightarrow (a, b, c4)$. Notice that $\{1, 2, 3, 4\} \subseteq \{a, b, c, 1, 2, 3, 4\}$. However, $L(G_2) = \{a^n b^n c^n rev(\rho) \mid n \geq 0, S \Rightarrow^* a^n b^n c^n rev(\rho)\, [\rho]\} \neq \{a^n b^n c^n \rho \mid n \geq 0, S \Rightarrow^* a^n b^n c^n \rho\, [\rho]\}$, so G_2 is no proper generator of its sentences with their parses either. Third, consider $G_3 = (\{S, A, B, C, a, b, c, 1, 2, 3, 4\}, P_3, S, \{a, b, c, 1, 2, 3, 4\})$ with P_3 containing $1 : (S) \rightarrow (1)$, $2 : (S) \rightarrow (ABC2\$)$, $3 : (A, B, C, \$) \rightarrow (aA, bB, cC, 3\$)$, $4 : (A, B, C, \$) \rightarrow (a, b, c, 4)$. Observe that $L(G_3) = \{a^n b^n c^n \rho \mid n \geq 0, S \Rightarrow^* a^n b^n c^n \rho\, [\rho]\}$, so G_3 is a proper generator of its sentences with their parses.

4 Results

Next, we demonstrate that for every recursively enumerable language L, there is a *PSCG* $G = (V, P, S, T)$, which represents a proper generator of its sentences with their parses so that L results from $L(G)$ by eliminating all production labels in $L(G)$. To express this property formally, we introduce the weak identity π from V^* to $(V - lab(G))^*$ defined as $\pi(a) = a$ for every $a \in (V - lab(G))$ and $\pi(p) = \epsilon$ for every $p \in lab(G)$ and use π in the next main theorem of this paper.

Theorem 1. *For every recursively enumerable language L, there exists a PSCG G such that G is a proper generator of its sentences with their parses and $L = \pi(L(G))$.*

Proof. Let L be a recursively enumerable language. Then, there is a *SCG* $\bar{G} = (\bar{V}, \bar{P}, \bar{S}, \bar{T})$ such that $L = L(\bar{G})$ (see [7]). Set $\Phi = \{\langle a \rangle \mid a \in \bar{T}\}$. Define the homomorhism γ from $\bar{V}$ to $(\Phi \cup (\bar{V} - \bar{T}) \cup \{Y\})^+$ as $\gamma(a) = \langle a \rangle$ for all $a \in \bar{T}$ and $\gamma(A) = A$ for all $A \in \bar{V} - \bar{T}$. Extend the domain of γ to $\bar{V}^+$ in the standard manner; non-standardly, however, define $\gamma(\epsilon) = Y$ rather than $\gamma(\epsilon) = \epsilon$. (Let us note that at this point γ does not, strictly speaking, represent a morphism on $\bar{V}^*$.) Next, we introduce a *PSCG* $G = (V, P, S, T)$ such that G is a proper generator of its sentences with their parses and $L(\bar{G}) = \pi(L(G))$. Finally, set $\Gamma = \{\$_1, \$_2, \$_3\}$. Define the *PSCG*

$$G = (\{S, X, Y, Z\} \cup \bar{V} \cup lab(G) \cup \Phi \cup \Gamma, P, S, \bar{T} \cup lab(G))$$

with $lab(G) = \{\lfloor 0 \rfloor, \lfloor 1 \rfloor, \lfloor 2 \rfloor, \lfloor 3 \rfloor, \lfloor 4 \rfloor\} \cup \Xi_1 \cup \Xi_2 \cup \Xi_3$, where $\Xi_1 = \{\lfloor p1 \rfloor \mid p \in lab(\bar{G})\}$, $\Xi_2 = \{\lfloor a2 \rfloor \mid a \in \bar{T}\}$, $\Xi_3 = \{\lfloor a3 \rfloor \mid a \in \bar{T}\}$; without any loss of generality, assume $lab(G) \cap alph(L) = \emptyset$. P is constructed as follows:

1. Add
   $\lfloor 1 \rfloor : (S) \rightarrow (X \lfloor 1 \rfloor \$_1 Z \bar{S})$ to P;
   $\lfloor 1_\epsilon \rfloor : (S) \rightarrow (\lfloor 1_\epsilon \rfloor \$_1 \bar{S})$ to P;

2. For every $p : (A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n) \in \bar{P}$ add
$\lfloor p1 \rfloor : (\$_1, A_1, \ldots, A_n) \rightarrow (\lfloor p1 \rfloor \$_1, \gamma(x_1), \ldots, \gamma(x_n))$ to P;
in addition, add
$\lfloor 2 \rfloor : (\$_1) \rightarrow (\lfloor 2 \rfloor \$_2)$ to P;
$\lfloor 2_\epsilon \rfloor : (\$_1) \rightarrow (\lfloor 2_\epsilon \rfloor \$_3)$ to P;

3. For every $a \in \bar{T}$, add
$\lfloor a2 \rfloor : (X, \$_2, Z, \langle a \rangle) \rightarrow (aX, \lfloor a2 \rfloor \$_2, Y, Z)$ to P;
$\lfloor a3 \rfloor : (X, \$_2, Z, \langle a \rangle) \rightarrow (a, \lfloor a3 \rfloor \$_3, Y, Y)$ to P;

4. Add $\lfloor 3 \rfloor : (\$_3, Y) \rightarrow (\lfloor 3 \rfloor, \$_3)$ to P;

5. Add $\lfloor 4 \rfloor : (\$_3) \rightarrow (\lfloor 4 \rfloor)$ to P.

Basic Idea:

First, we explain how G makes the generation of a nonempty sentence followed by its parse; then, we explain the generation of the empty sentence followed by its parse.

G makes the generation of $a_1 a_2 \ldots a_n \rho$, where $n \geq 1$, each $a_i \in \bar{T}$ and ρ is the corresponding parse, by productions introduced in steps 1 through 5 in this order. After starting this generation by using the production from 1, it applies productions introduced in 2, which simulate the applications of productions from $\bar{P}$. More precisely, it simulates the use of $p : (A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n) \in \bar{P}$ by using $\lfloor p1 \rfloor : (\$_1, A_1, \ldots, A_n) \rightarrow (\lfloor p1 \rfloor \$_1, \gamma(x_1), \ldots, \gamma(x_n)) \in P$ so that it places its own label, $\lfloor p1 \rfloor$, right behind the previously generated production labels; this substring of labels occurs between the leftmost symbol, X, and $\$_1$, in the sentential form. Otherwise, $\lfloor p1 \rfloor : (\$_1, A_1, \ldots, A_n) \rightarrow (\lfloor p1 \rfloor \$_1, \gamma(x_1), \ldots, \gamma(x_n))$ is analogical to $p : (A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n)$ except that (i) the former has the fill-in symbol Y where the latter has ϵ and (ii) the former has $\langle a_i \rangle$ where the latter has terminal a_i. After using productions introduced in 2, G has its current sentential form of the form $X\tau\$_2 Z u_0 \langle a_1 \rangle u_1 \langle a_2 \rangle u_2 \ldots u_{n-1} \langle a_n \rangle u_n$, where τ is a prefix of ρ and $u_i \in \{Y\}^*$. By using productions from 3, it places $a_1 \ldots a_n$ at the beginning of the sentential form while replacing each $\langle a_i \rangle$ with Y and generating the production labels. By using productions labeled $\lfloor 3 \rfloor$ (see step 4), G replaces each Y with $\lfloor 3 \rfloor$ while shifting $\$_3$ to the right. Finally, the application of the production labeled with $\lfloor 4 \rfloor$ completes the generation of $a_1 a_2 \ldots a_n \rho$ (see step 5). Finally, let us explain how G makes the generation of the empty sentence ϵ followed by its parse. By use of productions labeled with $\lfloor 1_\epsilon \rfloor$ and $\lfloor 2_\epsilon \rfloor$ instead of $\lfloor 1 \rfloor$ and $\lfloor 2 \rfloor$, respectively, the process of placing terminal symbols at the beginning of the sentential form (by productions from step 3) is skipped; otherwise, the derivation proceeds as above.

Rigorous proof (Sketch):

Claim 1. *G generates every $w \in L(G) - lab(G)^+$ in the following way*

$$\begin{array}{llll} S & \Rightarrow & X\lfloor 1\rfloor\$_1 Z\bar{S} & [\lfloor 1\rfloor] \\ & \Rightarrow^+ & x & [\rho] \\ & \Rightarrow & y & [\lfloor 2\rfloor] \\ & \Rightarrow^* & z & [\sigma] \\ & \Rightarrow & u & [\lfloor a3\rfloor] \\ & \Rightarrow^+ & v & [\tau] \\ & \Rightarrow & w & [\lfloor 4\rfloor] \end{array} \tag{1}$$

where $\lfloor a3\rfloor \in \Xi_3$, ρ, σ and τ are sequences consisting from Ξ_1, Ξ_2 and $\{\lfloor 3\rfloor\}$, respectively.

Proof. First, let us make these four observations:

1. Since the only productions with S on its left-hand side are productions introduced in step 1 of the construction, $S \Rightarrow^+ w$ surely starts with a step made by one of these productions. Notice that $alph(\{w\}) \cap \bar{T} \neq \emptyset$ and only productions labeled with $p \in \Xi_2 \cup \Xi_3$ satisfy $a \in alph(\{rhs(p)\})$, $a \in \bar{T}$. As $X = {}_l pos(p, 1)$, $a \in alph(\{{}_r pos(p, 1)\})$, and only production labeled with $p \in \lfloor 1\rfloor$ satisfies $X \in alph(\{rhs(p)\})$, the derivation starts with a step made by this production. This derivation ends by applying production labeled with $\lfloor 4\rfloor$ because it is the only production with its right-hand side over T^*. Thus, $S \Rightarrow^+ w$ can be expressed as

$$\begin{array}{llll} S & \Rightarrow & X\lfloor 1\rfloor\$_1 Z\bar{S} & [\lfloor 1\rfloor] \\ & \Rightarrow^+ & v & \\ & \Rightarrow & w & [\lfloor 4\rfloor] \end{array}$$

2. Let p be the label of any production introduced in steps 2 through 4 of the construction; then, $occur(lhs(p), \Gamma) = occur(rhs(p), \Gamma) = 1$. In greater detail, for every $\lfloor p1\rfloor \in \Xi_1$, $\lfloor a2\rfloor \in \Xi_2$, $\lfloor a3\rfloor \in \Xi_3$, productions introduced in step 2 satisfy $occur(lhs(\lfloor p1\rfloor), \{\$_1\}) = occur(rhs(\lfloor p1\rfloor), \{\$_1\}) = 1$, $occur(lhs(\lfloor 2\rfloor), \{\$_1\}) = 1$, $occur(rhs(\lfloor 2\rfloor), \{\$_2\}) = 1$, $occur(lhs(\lfloor 2_\epsilon\rfloor), \{\$_1\}) = 1$, $occur(rhs(\lfloor 2_\epsilon\rfloor), \{\$_3\}) = 1$. Similarly, productions introduced in step 3 satisfy $occur(lhs(\lfloor a2\rfloor), \{\$_2\}) = occur(rhs(\lfloor a2\rfloor), \{\$_2\}) = 1$, $occur(lhs(\lfloor a3\rfloor), \{\$_2\}) = 1$, $occur(rhs(\lfloor a3\rfloor), \{\$_3\}) = 1$. Finally, production introduced in step 4 satisfies $occur(lhs(\lfloor 3\rfloor), \{\$_3\}) = occur(rhs(\lfloor 3\rfloor), \{\$_3\}) = 1$.

3. Because $X \in alph(\{x\})$ and only productions labeled with $p \in \Xi_3$ satisfy $X \in alph(\{lhs(p)\})$ and $X \notin alph(\{rhs(p)\})$, production labeled with $\lfloor 2_\epsilon\rfloor$ cannot be used.

4. Let p be the label of any production introduced in steps 1 through 5; then, $alph(\{rhs(p)\}) \cap lab(G) = \{p\}$ and $occur(rhs(p), \{p\}) = 1$.

Based on these observations, notice that G generates every $w \in L(G) - \{\lfloor 0 \rfloor\}$ in the way described in the formulation of Claim 1. $\square$

Claim 2. *Consider derivation* (1). *In its beginning*

$$\begin{array}{lll} S & \Rightarrow & X\lfloor 1\rfloor \$_1 Z\bar{S} & [\lfloor 1\rfloor] \\ & \Rightarrow^+ & x & [\rho] \\ & \Rightarrow & y & [\lfloor 2\rfloor] \end{array}$$

every sentential form s *in* $X\lfloor 1\rfloor\$_1Z\bar{S} \Rightarrow^+ x$ *satisfies* $s \in \{X\}lab(G)^+\{\$_1\}\{Z\}(\Phi \cup (\bar{V} - \bar{T}) \cup \{Y\})^+$ *and* $y \in \{X\}lab(G)^+\{\$_2\}\{Z\}(\Phi \cup \{Y\})^+$.

Proof. By the definition of homomorphism γ, productions labeled with $\lfloor p1\rfloor$ rewrite symbols over $\Phi \cup (\bar{V} - \bar{T}) \cup \{Y\}$ and change $\$_1$ to $\lfloor p1\rfloor\$_1$. Since $\bar{V} \cap \{X, \$_1, Z\} = \emptyset$, every sentential form s in $X\lfloor 1\rfloor\$_1Z\bar{S} \Rightarrow^+ x$ satisfies $s \in \{X\}lab(G)^+\{\$_1\}\{Z\}(\Phi \cup (\bar{V} - \bar{T}) \cup \{Y\})^+$. Only Ξ_1 contains production labels p satisfying $alph(\{lhs(p)\}) \cap (\bar{V} - \bar{T}) \neq \emptyset$. Therefore, to generate $w \in T^*$, productions labeled with $\lfloor p1\rfloor$ have to be applied until $s \in \{X\}lab(G)^+\{\$_1\}\{Z\}(\Phi \cup \{Y\})^+$. Finally, a production labeled with $\lfloor 2\rfloor$ is used, so $y \in \{X\}lab(G)^+\{\$_2\}\{Z\}(\Phi \cup \{Y\})^+$ and the claim holds. $\square$

Claim 3. *In*

$$\begin{array}{llll} y & \Rightarrow^* & z & [\lfloor\sigma\rfloor] \\ & \Rightarrow & u & [\lfloor a3\rfloor] \end{array}$$

of derivation (1), *every sentential form* o *in* $y \Rightarrow^* z$ *can be expressed as* $o \in \bar{T}^*\{X\}lab(G)^+\{\$_2\}\{Y\}^*\{Z\}(\Phi \cup \{Y\})^+$ *and* $u \in \bar{T}^+lab(G)^+\{\$_3\}\{Y\}^+$. *In greater detail,*

$$\begin{array}{lll} & X\lfloor p_1\rfloor \dots \lfloor p_n\rfloor \$_2 Z Y^{i_0}\langle b_1\rangle Y^{i_1}\langle b_2\rangle Y^{i_2}\dots\langle b_m\rangle Y^{i_m} & \\ \Rightarrow & b_1X\lfloor p_1\rfloor \dots \lfloor p_n\rfloor\lfloor b_1 2\rfloor \$_2 Y^{i_0+1} Z Y^{i_1}\langle b_2\rangle Y^{i_2}\dots\langle b_m\rangle Y^{i_m} & [\lfloor b_1 2\rfloor] \\ \Rightarrow & b_1b_2X\lfloor p_1\rfloor \dots \lfloor p_n\rfloor\lfloor b_1 2\rfloor\lfloor b_2 2\rfloor \$_2 Y^{i_0+1}Y^{i_1+1} Z Y^{i_2}\dots\langle b_m\rangle Y^{i_m} & [\lfloor b_2 2\rfloor] \\ \Rightarrow^{m-3} & b_1b_2\dots b_{m-1}X\lfloor p_1\rfloor \dots \lfloor p_n\rfloor\lfloor b_1 2\rfloor\dots\lfloor b_{m-1}2\rfloor \$_2 Y^{i_0+1}Y^{i_1+1}\dots & \\ & \dots Y^{i_{m-2}} Z Y^{i_{m-1}}\langle b_m\rangle Y^{i_m} & [\bar{\sigma}] \\ \Rightarrow & b_1b_2\dots b_m\lfloor p_1\rfloor\dots\lfloor p_n\rfloor\lfloor b_1 2\rfloor\dots\lfloor b_{m-1}2\rfloor\lfloor b_m 3\rfloor \$_3 Y^{i_0+1}Y^{i_1+1}\dots Y^{i_m+1} & [\lfloor b_m 3\rfloor] \end{array}$$

where $\lfloor p_1\rfloor, \dots, \lfloor p_n\rfloor \in lab(G)$ *are labels that denote productions introduced in 1–2,* $\langle b_1\rangle, \dots, \langle b_m\rangle \in \Phi$, $b_1, \dots, b_m \in \bar{T}$, $\bar{\sigma} = \lfloor b_3 2\rfloor \dots \lfloor b_{m-1}2\rfloor$, $i_0, i_1, \dots, i_m \geq 0$, $m = |s|$, *where* $s \in L(\bar{G})$ *is a corresponding sentence of the SCG* $\bar{G}$.

Proof. Notice that $occur(lhs(\lfloor a2\rfloor), \{X\}) = occur(rhs(\lfloor a2\rfloor), \{X\}) = 1$ and $occur(lhs(\lfloor a2\rfloor), \{Y\}) = occur(rhs(\lfloor a2\rfloor), \{Y\}) = 1$. In every derivation step of $y \Rightarrow^* z$, the the first symbol $\langle b\rangle \in \Phi$, following Z is replaced with Z, X is changed to bX, and $\$_2$ is changed to $l\$_2$, where $l \in lab(G)$. As $\lfloor a2\rfloor$ and $\lfloor a3\rfloor$ are the only production labels p satisfying $alph(\{lhs(p)\}) \cap \Phi \neq \emptyset$, $alph(\{rhs(p)\}) \cap \Phi = \emptyset$ and ${}_lpos(\lfloor a2\rfloor, 3) = Z$, ${}_rpos(\lfloor a2\rfloor, 4) = Z$, Z can replace only the first occurance of

$\langle b\rangle \in \Phi$ behind Z to generate $w \in T^*$. Productions labeled with $\lfloor a2\rfloor$ are used $m-1$ times. Thus, $y \Rightarrow^* z$ has the form

$$\begin{array}{lll} & X\lfloor p_1\rfloor \dots \lfloor p_n\rfloor \$_2 Z Y^{i_0}\langle b_1\rangle Y^{i_1}\langle b_2\rangle Y^{i_2}\dots\langle b_m\rangle Y^{i_m} & \\ \Rightarrow & b_1 X\lfloor p_1\rfloor \dots \lfloor p_n\rfloor \lfloor b_1 2\rfloor \$_2 Y^{i_0+1} Z Y^{i_1}\langle b_2\rangle Y^{i_2}\dots\langle b_m\rangle Y^{i_m} & [\lfloor b_1 2\rfloor] \\ \Rightarrow & b_1 b_2 X\lfloor p_1\rfloor \dots \lfloor p_n\rfloor \lfloor b_1 2\rfloor\lfloor b_2 2\rfloor \$_2 Y^{i_0+1} Y^{i_1+1} Z Y^{i_2}\dots\langle b_m\rangle Y^{i_m} & [\lfloor b_2 2\rfloor] \\ \Rightarrow^{m-3} & b_1 b_2\dots b_{m-1} X\lfloor p_1\rfloor \dots \lfloor p_n\rfloor \lfloor b_1 2\rfloor \lfloor b_{m-1}2\rfloor \$_2 Y^{i_0+1} Y^{i_1+1}\dots & \\ & \dots Y^{i_{m-2}} Z Y^{i_{m-1}}\langle b_m\rangle Y^{i_m} & [\bar{\sigma}] \end{array}$$

where every sentential form satisfies $\bar{T}^*\{X\}lab(G)^+\{\$_2\}\{Y\}^*\{Z\}(\Phi\cup\{Y\})^+$.

Finally, some production labeled with $\lfloor a3\rfloor$ is applied; therefore, $z \Rightarrow u$ can be expressed as

$$\begin{array}{lll} & b_1 b_2\dots b_{m-1} X\lfloor p_1\rfloor \dots \lfloor p_n\rfloor \lfloor b_1 2\rfloor \dots \lfloor b_{m-1}2\rfloor \$_2 Y^{i_0+1} Y^{i_1+1}\dots & \\ & \dots Y^{i_{m-2}} Z Y^{i_{m-1}}\langle b_m\rangle Y^{i_m} & \\ \Rightarrow & b_1 b_2\dots b_m \lfloor p_1\rfloor \dots \lfloor p_n\rfloor \lfloor b_1 2\rfloor \dots \lfloor b_{m-1}2\rfloor \lfloor b_m 3\rfloor \$_3 Y^{i_0+1} Y^{i_1+1}\dots Y^{i_m+1} & [\lfloor b_m 3\rfloor] \end{array}$$

with $u \in \bar{T}^+lab(G)^+\{\$_3\}\{Y\}^+$.

Putting together the previous parts of derivation, we obtain the formulation of Claim 3. Thus, Claim 3 holds. □ □

Claim 4. *In*

$$\begin{array}{llll} u & \Rightarrow^+ & v & [\tau] \\ & \Rightarrow & w & [\lfloor 4\rfloor] \end{array}$$

of derivation (1), *every sentential form* s *of* $u \Rightarrow^+ v$ *satisfies* $s \in \bar{T}^+lab(G)^+\{\$_3\}\{Y\}^*$ *and* $w \in \bar{T}^+lab(G)^+$. *In greater detail, this derivation can be expressed as*

$$\begin{array}{lll} & b_1\dots b_m\lfloor p_1\rfloor\dots\lfloor p_n\rfloor\{\$_3\}Y^i & \\ \Rightarrow & b_1\dots b_m\lfloor p_1\rfloor\dots\lfloor p_n\rfloor\lfloor 3\rfloor\{\$_3\}Y^{i-1} & [\lfloor 3\rfloor] \\ \Rightarrow & b_1\dots b_m\lfloor p_1\rfloor\dots\lfloor p_n\rfloor\lfloor 3\rfloor\lfloor 3\rfloor\{\$_3\}Y^{i-2} & [\lfloor 3\rfloor] \\ \Rightarrow^{i-3} & b_1\dots b_m\lfloor p_1\rfloor\dots\lfloor p_n\rfloor\lfloor 3\rfloor^{i-1}\{\$_3\}Y & [\bar{\tau}] \\ \Rightarrow & b_1\dots b_m\lfloor p_1\rfloor\dots\lfloor p_n\rfloor\lfloor 3\rfloor^{i}\{\$_3\} & [\lfloor 3\rfloor] \\ \Rightarrow & b_1\dots b_m\lfloor p_1\rfloor\dots\lfloor p_n\rfloor\lfloor 3\rfloor^{i}\lfloor 4\rfloor & [\lfloor 4\rfloor] \end{array}$$

where all $b_j \in \bar{T}$, $1 \le j \le m$ *and* $\lfloor p_k\rfloor \in lab(G)$, $1 \le k \le n$ *are labels that denote productions introduced in steps 1 through 3 of the construction,* $\bar{\tau}$ *is a sequence of production labels* $\lfloor 3\rfloor$.

Proof. Notice that ${}_lpos(\lfloor 3\rfloor, 1) = {}_rpos(\lfloor 3\rfloor, 2) = \$_3$. Observe, that in order to generate $w \in T^*$ the first occurrence of Y following $\$_3$ has to be taken by $\lfloor 3\rfloor$ in each derivation step. Finally, $\lfloor 4\rfloor$ is applied. At this moment, w satisfies $w \in T^*$ and $w \in \bar{T}^+lab(G)^+$. □

The next claim formally demonstrates how G generates the empty sentence ϵ followed by its parse.

Claim 5. *G generates every* $w \in L(G) \cap lab(G)^+$ *in the following way*

$$\begin{array}{llll} S & \Rightarrow & \lfloor 1_\epsilon \rfloor \$_1 \bar{S} & [\lfloor 1_\epsilon \rfloor] \\ & \Rightarrow^+ & x & [\rho] \\ & \Rightarrow & y & [\lfloor 2_\epsilon \rfloor] \\ & \Rightarrow^+ & v & [\tau] \\ & \Rightarrow & w & [\lfloor 4 \rfloor] \end{array} \tag{1}$$

where ρ *and* τ *are sequences consisting from* Ξ_1 *and* $\{\lfloor 3 \rfloor\}$*, respectively.*

Proof. Notice that $alph(\{w\}) \cap \bar{T} = \emptyset$ and only productions labeled with $p \in \Xi_3$ satisfy $X \in alph(\{lhs(p)\})$, $X \notin alph(\{rhs(p)\})$ and $X = {}_lpos(p, 1)$, $a = {}_rpos(p, 1)$, $a \in \bar{T}$. Therefore, X cannot appear in any sentential form of $S \Rightarrow^* w$, and the derivation starts with a step made by $\lfloor 1_\epsilon \rfloor$. As $X \notin alph(\{x\})$ and for $p \in \Xi_2 \cup \Xi_3$, $X \in alph(\{lhs(p)\})$, the production labeled with $\lfloor 2_\epsilon \rfloor$ has to be used. Observe that other derivation steps are made in the way described in Claim 2 and Claim 4. □

From Claims 4 and 5, it follows that for every recursively enumerable language L, there exists a *PSCG* G such that G is a proper generator of its sentences with their parses and $L = \pi(L(G))$. □

From Theorem 1, we obtain:

Corollary 1. *For every recursively enumerable language L, there exists a PSCG G such that G is a proper generator of its sentences with their parses and* $L = L(G)/lab(G)^* \cap alph(L)^*$.

Alternatively, we can introduce a *SCG* $G = (V, P, S, T)$, as a *proper generator of its sentences preceded by their parses* so that $L(G) = \{x \mid x = \rho y, y \in (T - lab(G))^*, \rho \in lab(G)^*, S \Rightarrow^* x\,[\rho]\}$.

Theorem 2. *For every recursively enumerable language L, there exists a PSCG G such that G is a proper generator of its sentences preceded by their parses and* $L = \pi(L(G))$.

Proof. This theorem can be proved by a straightforward modification of Theorem 1. A detailed version of this proof is left to the reader. □

Corollary 2. *For every recursively enumerable language L, there exists a PSCG G such that G is a proper generator of its sentences preceded by their parses and* $L = lab(G)^* \backslash L(G) \cap alph(L)^*$.

5 Conclusion

In this concluding section, we make some final notes and suggestions regarding the future investigation.

First, notice that all the above results can be also established so that the generated sentences are followed by the reversals of their parses.

Second, consider the unordered scattered context grammars (see page 260 in [13]). In essence, in this version of scattered context grammars, we apply a production of the form $(A_1 \rightarrow x_1, \ldots, A_n \rightarrow x_n)$ so we simultaneously replace A_i with x_i, for all $i = 1, \ldots, n$, no matter in what order the nonterminals A_i appear in the rewritten word. Naturally, we are tempted to use the construction given in the proof of Theorem 1 for these grammars in order to obtain analogical results to the above results. Unfortunately, this construction does not work for the unordered versions of scattered context grammmars. Specifically, steps 3 and 4 of the construction require the prescribed order of rewritten nonterminals; otherwise, the result is not guaranteed. Can we prove the results of this paper in terms of unordered scattered context grammars by using some other methods?

Finally, let us recall that we have demonstrated that for every recursively enumerable language, there exists a propagating scattered context grammar that generate the language's sentences followed by their parses. From a broader perspective, we could naturally reformulate this generation of sentences with their parses in terms of other propagating rewriting mechanisms that define the language family contained in the family of context-sensitive languages. Probably, some propagating parallel rewriting mechanisms, such as propagating PC grammar systems (see Chapter 4 in Volume 2 of [12]), can be used in this way. Furthermore, some propagating regulated grammars, such as propagating matrix grammars (see Chapter 3 in Volume 3 of [12]), seems to be suitable for this generation as well. On the other hand, we can hardly base the generation of sentences with their parses upon classical sequential rewriting mechanisms, such as context-free grammars. The authors suggest these problem areas as the topics of future investigation that continues with the discussion opened in the present paper.

Acknowledgements We thank the anonymous referee for useful comments concerning the first version of this paper. The first author gladly acknowledges support of GACR grant 201/04/0441.

References

[1] Chatterjee, S. (eds.): *Languages and Compilers for Parallel Computing*, Springer-Verlag, London, 1999.

[2] Darte, A. et al. (eds.): *Compilers for Parallel Computers*, World Scientific, Singapore, 2000.

[3] Fernau, H.: Scattered Context Grammars with Regulation, *Annals of Bucharest Univ., Math.-Informatics Series* 45(1) (1996), 41–49.

[4] Gonczarowski, J. and Warmuth, M. K.: Scattered Versus Context-Sensitive Rewriting, *Acta Informatica* 27 (1989), 81–95.

[5] Greibach, S. and Hopcroft, J. E.: Scattered Context Grammars, *Journal of Computer and System Sciences* 3 (1969), 233–247.

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

[7] Meduna, A.: A Trivial Method of Characterizing the Family of Recursively Enumerable Languages by Scattered Context Grammars, *EATCS Bulletin* 56 (1995), 104–106.

[8] Meduna, A.: Generative Power of Three-Nonterminal Scattered Context Grammars, *Theoretical Computer Science* 237 (2000), 625–631.

[9] Midkiff, S. P. et al. (eds.): *Languages and Compilers for Parallel Computing* (13th International Workshop on Languages and Compilers for Parallel Computing, 2000, Yorktown Heights, N.Y.), Springer, London, 2001.

[10] Rauchwerger, L. (eds.): *Languages and Compilers for Parallel Computing* (16th International Workshop, October 2003, Colledge Station, Texas), Springer, London, 2004.

[11] Revesz, G. E.: *Introduction to Formal Language Theory*, McGraw-Hill, New York, 1983.

[12] Rozenberg, G. and Salomaa, A.(eds.): *Handbook of Formal Languages*, Volume 1 through 3, Springer-Verlag, 1997.

[13] Salomaa, A.: *Formal Languages*, Academic Press, London, 1973.

[14] Vaszil, G.: On the Number of Conditional Rules in Simple Semi-conditional Grammars, *Theoretical Computer Science*, 2004 (in press).

[15] Virkkunen, V.: On Scattered Context Grammars, *Acta Universitatis Ouluensis*, Series A, Mathematica 6 (1973), 75–82.

[16] Wolfe, M. J.: *High Performance Compilers for Parallel Computing*, Addison-Wesley, Redwood City, 1996.

Received May, 2004