GENERATION OF SENTENCES WITH THEIR PARSES BY SCATTERED CONTEXT GRAMMARS

Jiří TECHET, Master Degree Programme (4)
Dept. of Information Systems, FIT, BUT
E-mail: xteche00@stud.fit.vutbr.cz

Supervised by: Dr. Alexander Meduna

ABSTRACT

This paper uses the propagating scattered context grammars to generate their language's sentences together with their parses (the sequences of productions whose use lead to the generation of the corresponding sentences). It proves 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.

1 INTRODUCTION

Scattered context grammars generate their languages in a parallel ways, thus inspiring us to use them in parsing somehow. Indeed, parsing is inseparable form grammars, and as parallelism fulfils a crucial topic in its investigation today, the use of scattered context grammars in relation 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 productions whose use lead to the generation of the corresponding sentences (in the literature, derivations words and Szilard words are synonymous with parses). It demonstrates 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, the use of propagating scattered context grammars in this paper provides us with parses corresponding to the generated sentences, which obviously represent useful information, and they incease their power in this way.

2 PRELIMINARIES

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 $V^+ = V^* - \{\varepsilon\}$. For $w \in V^*$, $|w|$ and reversal(w) denotes 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 $v \in V^+$, rm(v) denotes the rightmost symbol of v. For $L \subseteq V^*$, alph(L) denotes the set of symbols appearing in a word of L. A homomorphism, ω, over V^*, represents an almost identity if there exists a symbol, $\# \in V$, such that $\omega(a) = a$ for every $a \in (\Sigma - \{\#\})$ and $\omega(\#) \in \{\#, \varepsilon\}$.

A *scattered context grammar*, 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 $(A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n)$, for some $n \geq 1$, where $A_i \in V - T$, $x_i \in V^*$, for $1 \leq i \leq n$. If every $(A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n) \in P$ satisfies $x_i \in V^+$ for all $1 \leq i \leq n$, G is a *propagating scattered context grammar*, a PSCG for short. If $(A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n) \in P$, $u = u_1 A_1 u_2 \ldots u_n A_n u_{n+1}$, and $v = u_1 x_1 u_2 \ldots u_n x_n u_{n+1}$, where $u_i \in V^*$, $1 \leq i \leq n$, then $u \Rightarrow v\,[(A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n)]$ 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 $L(G) = \{x \mid x \in T^*, S \Rightarrow^* x\}$.

3 DEFINITIONS

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 $(A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n)$, we write $p : (A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n)$. For every $p : (A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n) \in P$, lhs$(p)$ and rhs(p) denote $A_1 A_2 \ldots A_n$ and $x_1 x_2 \ldots x_n$, respectively. Furthermore, l-pos(p, j) and r-pos(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. The *language of generated sentences with their parses* is denoted by $L(G)_{parse}$ and defined as $L(G)_{parse} = \{x\rho \mid x \in T^*, \rho \in \text{lab}(G)^*, S \Rightarrow^* x\,[\rho]\}$; notice that $L(G)_{parse} \subseteq T^*\,\text{lab}(G)^*$. Let π be the weak identity from $(V \cup \text{lab}(G))^*$ to V^* defined as $\pi(a) = a$ for every $a \in V$ and $\pi(p) = \varepsilon$ for every $p \in$ lab(G). Observe that $L(G) = \pi(L(G)_{parse})$. Let $G = (V, P, S, T)$ be a SCG. For G, set ${}_\pi G = (\pi(V), {}_\pi P, S, \pi(T))$ with lab(G) = lab$({}_\pi G)$ and $p : (A_1, \ldots, A_n) \rightarrow (\pi(x_1), \ldots, \pi(x_n)) \in {}_\pi P$ iff $p : (A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n) \in P$. G is a *proper generator of its sentences with their parses* if $L(G) = L({}_\pi G)_{parse}$. Consequently, every $x \in L(G)$ is of the form $x = y\rho$, where $y \in (T - \text{lab}(G))^*$ and $\rho \in$ lab$(G)^*$, and $S \Rightarrow^* y[\rho]$ in ${}_\pi G$. Observe that alph$(L({}_\pi G)) \cap$ lab$({}_\pi G) = \emptyset$.

4 RESULTS

Theorem. 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 (Sketch). Let L be a recursively enumerable language. Then, there is a SCG $G = (V, P, S, T)$ such that $L = L(G)$. Set $\Phi = \{\langle a \rangle \mid a \in T\}$. Define the homomorhism γ from V to $(\Phi \cup (V - T) \cup \{Y\})^+$ as $\gamma(a) = \langle a \rangle$ for all $a \in T$ and $\gamma(A) = A$ for all $A \in V - T$. Extend the domain of γ to V^+ in the standard manner; non-standardly, however, define $\gamma(\varepsilon) = Y$ rather than $\gamma(\varepsilon) = \varepsilon$. Next, we introduce a PSCG, $\underline{G} = (\underline{V}, \underline{P}, \underline{S}, \underline{T})$, such that $\underline{G}$ is a proper generator of its sentences with their parses and $L(G) = \pi(L(\underline{G}))$. Finally, set $\Gamma = \{\$_1, \$_2, \$_3\}$. Define the PSCG

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

with $\mathrm{lab}(\bar{G}) = \{\lfloor 0 \rfloor, \lfloor 1 \rfloor, \lfloor 2 \rfloor, \lfloor 3 \rfloor, \lfloor 4 \rfloor\} \cup \Xi_1 \cup \Xi_2 \cup \Xi_3$, where $\Xi_1 = \{\lfloor p1 \rfloor \mid p \in \mathrm{lab}(G)\}$, $\Xi_2 = \{\lfloor a2 \rfloor \mid a \in T\}$, $\Xi_3 = \{\lfloor a3 \rfloor \mid a \in T\}$, and $\bar{P}$ constructed as follows.

0. If $\varepsilon \in L(G)$, add $\lfloor 0 \rfloor : (\bar{S}) \to (\lfloor 0 \rfloor)$ to $\bar{P}$;
1. Add $\lfloor 1 \rfloor : (\bar{S}) \to (X \lfloor 1 \rfloor \$_1 Z S)$ to $\bar{P}$;
2. For every $p : (A_1, \ldots, A_n) \to (x_1, \ldots, x_n) \in \bar{P}$ add
   $\lfloor p1 \rfloor : (\$_1, A_1, \ldots, A_n) \to (\lfloor p1 \rfloor \$_1, \gamma(x_1), \ldots, \gamma(x_n))$ to $\bar{P}$;
   in addition, add $\lfloor 2 \rfloor : (\$_1) \to (\lfloor 2 \rfloor \$_2)$ to $\bar{P}$;
3. For every $a \in T$, add $\lfloor a2 \rfloor : (X, \$_2, Z, \langle a \rangle) \to (aX, \lfloor a2 \rfloor \$_2, Y, Z)$ to $\bar{P}$;
   $\lfloor a3 \rfloor : (X, \$_2, Z, \langle a \rangle) \to (a, \lfloor a3 \rfloor \$_3, Y, Y)$ to $\bar{P}$;
4. Add $\lfloor 3 \rfloor : (\$_3, Y) \to (\lfloor 3 \rfloor, \$_3)$ to $\bar{P}$;
5. Add $\lfloor 4 \rfloor : (\$_3) \to (\lfloor 4 \rfloor)$ to $\bar{P}$.

Then, if $\varepsilon \in L(G)$, $\bar{S} \Rightarrow \lfloor 0 \rfloor [\lfloor 0 \rfloor]$ in $\bar{G}$, whereas every $x \in L(\bar{G}) - \{\lfloor 0 \rfloor\}$ is generated by $\bar{G}$ in this way:

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

where $a \in T$, ρ, σ and τ are sequences consisting from Ξ_1, Ξ_2 and Ξ_3, respectively.

REFERENCES

[1] Meduna, A.: Coincidental extension of scattered context languages, Acta Informatica 39, 307-314 (2003), Springer, ISSN 0236-0112

[2] Meduna, A.: Syntactic complexity of scattered context grammars, Acta Informatica 32, 285-298 (1995), Springer, ISSN 0001-5903