k-Limited Erasing Performed by Scattered Context Grammars

Jiří Techet Tomáš Masopust (Alexander Meduna)

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

Modern Formal Language Theory, 2007

Scattered Context Grammar

Scattered context grammar (SC grammar)

A SC grammar is a quadruple, $G = (V, T, P, S)$, where

V is a finite alphabet

T is a set of terminals, $T \subset V$

S is a starting symbol, $S \in (V - T)$

P is a finite set of productions of the form: $(A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n)$; $A_1, \ldots, A_n \in (V - T)$; $x_1, \ldots, x_n \in V^*$

- $\text{len}((A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n)) = n$
- $\text{pos}(a_1 \ldots a_i \ldots a_n, i) = a_i$

Propagating scattered context grammar (PSC grammar)

- every $(A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n)$ satisfies $x_1, \ldots, x_n \in V^+$

Generated Language

Derivation step

For $(A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n) \in P$ and

$$u = u_1 A_1 \ldots u_n A_n u_{n+1}$$
$$v = u_1 x_1 \ldots u_n x_n u_{n+1}$$

we write $u \Rightarrow v \; [(A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n)]$

Generated language

$L(G) = \{x \in T^* \,:\, S \Rightarrow^* x\}$

Generative power

- $\mathcal{L}_{SC} = \mathcal{L}_{RE}$
- $\mathcal{L}_{CF} \subset \mathcal{L}_{PSC} \subseteq \mathcal{L}_{CS}$

PSC Grammar—Example

Example

$G_1 = (\{A, B, C, S, a, b, c\}, \{a, b, c\}, P_1, S)$ with

$$\begin{aligned} P_1 = \{&(S) \rightarrow (ABC),\\ &(A, B, C) \rightarrow (aA, bB, cC),\\ &(A, B, C) \rightarrow (\varepsilon, \varepsilon, \varepsilon)\} \end{aligned}$$

$S \Rightarrow ABC \Rightarrow aAbBcC \Rightarrow aaAbbBccC \Rightarrow aabbcc$
$L(G_1) = \{a^n b^n c^n : n \geq 0\}$

Example

$G_2 = (\{A, B, C, S, a, b, c\}, \{a, b, c\}, P_2, S)$ with

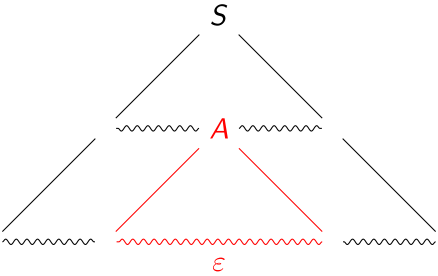
$$\begin{aligned} P_2 = \{&(S) \rightarrow (\varepsilon), (S) \rightarrow (ABC),\\ &(A, B, C) \rightarrow (aA, bB, cC),\\ &(A, B, C) \rightarrow (a, b, c)\} \end{aligned}$$

Symbols Erased During Derivation

Symbols erased during derivation

A symbol, A, is erased during a derivation if the frontier of the subtree rooted at A is ε.

- If the symbol A is erased, we write $\check{A}$;
- otherwise the symbol is not erased and we write $\hat{A}$.

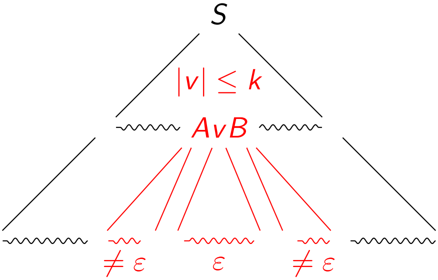


Nonterminals Erased in a k-Limited Way

Nonterminals erased in a k-limited way

For every $y \in L(G)$ there exists a derivation in which every sentential form x satisfies:

1. Every $x = uAvBw$, $\hat{A}$, $\hat{B}$, $\check{v}$, satisfies $|v| \leq k$.
2. Every $x = uAw$, $\hat{A}$, satisfies: if $\check{u}$ or $\check{w}$, then $|u| \leq k$ or $|w| \leq k$, respectively.



Results

Theorem

For every SC grammar, G, which erases its nonterminals in a k-limited way there exists a propagating SC grammar, $\bar{G}$, such that $L(G) = L(\bar{G})$.

Basic Idea—Demonstration

Example

$G_3 = (\{A, B, C, S, a, b, c\}, \{a, b, c\}, P_3, S)$ with

$$P_3 = \{(S) \rightarrow (ABC), (A, B, C) \rightarrow (\varepsilon, \varepsilon, \varepsilon),\\ (A) \rightarrow (aAA), (B) \rightarrow (bBB), (C) \rightarrow (cCC)\}$$

$$S \Rightarrow ABC \Rightarrow^3 aAAbBBcCC \Rightarrow aAbBcC \Rightarrow^3 aaAAbbBBccCC \Rightarrow^2 aabbcc$$

$$\begin{array}{lccccl}
 & & \langle S \rangle & & & (S) \rightarrow (ABC) \\
\Rightarrow & \langle A \rangle & \langle B \rangle & \langle C \rangle & & (A) \rightarrow (aAA), \ldots, (C) \rightarrow (cCC) \\
\Rightarrow^3 & \langle aAA \rangle & \langle bBB \rangle & \langle cCC \rangle & & (A, B, C) \rightarrow (\varepsilon, \varepsilon, \varepsilon) \\
\Rightarrow & \langle aA \rangle & \langle bB \rangle & \langle cC \rangle & & (A) \rightarrow (aAA), \ldots, (C) \rightarrow (cCC) \\
\Rightarrow^3 & \langle a \rangle\langle aAA \rangle & \langle b \rangle\langle bBB \rangle & \langle c \rangle\langle cCC \rangle & & (A, B, C) \rightarrow (\varepsilon, \varepsilon, \varepsilon) \\
\Rightarrow & \langle a \rangle\langle aA \rangle & \langle b \rangle\langle bB \rangle & \langle c \rangle\langle cC \rangle & & (A, B, C) \rightarrow (\varepsilon, \varepsilon, \varepsilon) \\
\Rightarrow & \langle a \rangle\langle a \rangle & \langle b \rangle\langle b \rangle & \langle c \rangle\langle c \rangle & & \\
\Rightarrow^6 & aa & bb & cc & &
\end{array}$$

Basic Idea

Let $G = (V, T, P, S)$ be a grammar which erases its nonterminals in a k-limited way. Every application of $p = (A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n)$ is simulated by a PSC grammar $\bar{G}$ as follows:

1. Every CF component of a SC production is simulated independently.

$$\langle z_{11} \lfloor p, 1 \rfloor z_{12} \rangle \langle z_{21} B_1 A_2 z_{22} A_3 z_{23} \rangle \ldots \langle z_{n1} A_n z_{n2} \rangle$$
$$\Rightarrow \langle z_{11} x_1 \lfloor p, 1 \rfloor' z_{12} \rangle \langle z_{21} B_1 A_2 z_{22} A_3 z_{23} \rangle \ldots \langle z_{n1} A_n z_{n2} \rangle$$

2. Transition to the next component of the SC production is performed.

$$\langle z_{11} x_1 \lfloor p, 1 \rfloor' z_{12} \rangle \langle z_{21} B_1 A_2 z_{22} A_3 z_{23} \rangle \ldots \langle z_{n1} A_n z_{n2} \rangle$$
$$\Rightarrow \langle z_{11} x_1 z_{12} \rangle \langle z_{21} B_1 \lfloor p, 2 \rfloor z_{22} A_3 z_{23} \rangle \ldots \langle z_{n1} A_n z_{n2} \rangle$$

3. After the last component is simulated, transition to the following SC production, $q = (B_1, \ldots, B_m) \rightarrow (y_1, \ldots, y_m)$, is performed.

$$\langle z_{11} x_1 z_{12} \rangle \langle z_{21} B_1 x_2 z_{22} x_3 z_{23} \rangle \ldots \langle z_{n1} x_n \lfloor p, n \rfloor' z_{n2} \rangle$$
$$\Rightarrow \langle z_{11} x_1 z_{12} \rangle \langle z_{21} \lfloor q, 1 \rfloor x_2 z_{22} x_3 z_{23} \rangle \ldots \langle z_{n1} x_n z_{n2} \rangle$$

Finally, we replace symbols of the form $\langle a \rangle$ with a, where $a \in T$.

Construction of Symbols

- $\Psi = \{\lfloor p, i \rfloor : p \in P, 1 \leq i \leq \text{len}(p)\}$
- $\Psi' = \{\lfloor p, i \rfloor' : \lfloor p, i \rfloor \in \Psi\}$
- $\bar{N}_1 = \{\langle x \rangle : x \in (V - T)^* \cup (V - T)^* T (V - T)^*, |x| \leq 2k + 1\}$

For every $\langle x \rangle \in \bar{N}_1$ and $\lfloor p, i \rfloor \in \Psi$, define
$\text{lhs-replace}(\langle x \rangle, \lfloor p, i \rfloor) = \{\langle x_1 \lfloor p, i \rfloor x_2 \rangle : x_1, x_2 \in V^*, x_1 \, \text{lhs}(\lfloor p, i \rfloor) x_2 = x\}$

- $\bar{N}_2 = \{\langle x \rangle : \langle x \rangle = \text{lhs-replace}(\langle y \rangle, \lfloor p, i \rfloor), \langle y \rangle \in \bar{N}_1, \lfloor p, i \rfloor \in \Psi\}$

For every $\langle x \rangle \in \bar{N}_1$ and $\lfloor p, i \rfloor' \in \Psi'$, define
$\text{insert}(\langle x \rangle, \lfloor p, i \rfloor') = \{\langle x_1 \lfloor p, i \rfloor' x_2 \rangle : x_1, x_2 \in V^*, x_1 x_2 = x\}$

- $\bar{N}'_2 = \{\langle x \rangle : \langle x \rangle = \text{insert}(\langle y \rangle, \lfloor p, i \rfloor'), \langle y \rangle \in \bar{N}_1, \lfloor p, i \rfloor' \in \Psi'\}$

Define the PSC grammar,

$$\bar{G} = (T \cup \bar{N}_1 \cup \bar{N}_2 \cup \bar{N}'_2 \cup \{\bar{S}\}, T, \bar{P}, \bar{S})$$

Construction of Productions I

For every $x = \langle x_1 \rangle \langle x_2 \rangle \dots \langle x_n \rangle \in (\bar{N}_1 \cup \bar{N}_2 \cup \bar{N}'_2)^*$ for some $n \geq 1$, define

$$\text{join}(x) = x_1 x_2 \dots x_n$$

For every $x \in \bar{N}_1 \cup \bar{N}_2 \cup \bar{N}'_2$, define

$$\text{split}(x) = \{y \,:\, x = \text{join}(y)\}$$

1. Initialization
   For every $p = (S) \rightarrow (x) \in P$, add
   $(\bar{S}) \rightarrow (\langle \lfloor p, 1 \rfloor \rangle)$ to $\bar{P}$
2. Termination
   For every $a \in T$, add
   $(\langle a \rangle) \rightarrow (a)$ to $\bar{P}$

Construction of Productions II

3. Simulation of one SC production's CF component
   For every
   - $\langle x_1\lfloor p,i\rfloor x_2\rangle \in \text{lhs-replace}(\langle x\rangle, \lfloor p,i\rfloor)$, $\langle x\rangle \in \bar{N}_1$, $\lfloor p,i\rfloor \in \Psi$, $x_1, x_2 \in V^*$
   - $Y \in \text{split}(x_1 \,\text{rhs}(\lfloor p,i\rfloor)\lfloor p,i\rfloor' x_2)$

   add $(\langle x_1\lfloor p,i\rfloor x_2\rangle) \rightarrow (Y)$ to $\bar{P}$
4. Transition to the next SC production's CF component
   For every
   - $\langle x\rangle \in \bar{N}_1$
   - $X \in \text{insert}(\langle x\rangle, \lfloor p,i\rfloor')$, where $p \in P$, $i < \text{len}(p)$
   - $\langle y\rangle \in \bar{N}_1$
   - $Y \in \text{lhs-replace}(\langle y\rangle, \lfloor p,i+1\rfloor)$, where $q \in P$

   add
   1. $(X, \langle y\rangle) \rightarrow (\langle x\rangle, Y)$ to $\bar{P}$
   2. If
      - $\langle x\rangle = \langle y\rangle$
      - $\text{pos}(X,l) = \lfloor p,i\rfloor'$, $\text{pos}(Y,m) = \lfloor p,i+1\rfloor'$, $l < m$

      add $(X) \rightarrow (Y)$ to $\bar{P}$

Construction of Productions III

5. Transition to the next SC production
   For every
   - $\langle x \rangle \in \bar{N}_1$
   - $X \in \text{insert}(\langle x \rangle, \lfloor p, n \rfloor')$, where $p \in P$, $\text{len}(p) = n$
   - $\langle y \rangle \in \bar{N}_1$
   - $Y \in \text{lhs-replace}(\langle y \rangle, \lfloor q, 1 \rfloor)$, where $q \in P$

   add
   1. $(X, \langle y \rangle) \rightarrow (\langle x \rangle, Y)$ to $\bar{P}$
   2. $(\langle y \rangle, X) \rightarrow (Y, \langle x \rangle)$ to $\bar{P}$
   3. If $\langle x \rangle = \langle y \rangle$, add
      $(X) \rightarrow (Y)$ to $\bar{P}$
   4. Finishing the simulation
      $(X) \rightarrow (\langle x \rangle)$ to $\bar{P}$

Summary and Future Investigation

Summary

- In general, in SC grammars ε-productions cannot be removed
- This removal is, however, possible under some conditions

Future investigation

- There are modifications of SC grammars which contain ε-productions and characterize all CS languages
- Is it possible to convert them to equivalent grammars which delete their nonterminals in a *k*-limited way?