Scattered Context Generators of Sentences With Their Parses

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 (SC grammar)

Scattered context grammar $G = (V, T, P, S)$

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 $(A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n)$; $A_1, \ldots, A_n \in (V - T)$; $x_1, \ldots, x_n \in V^*$

Propagating scattered context grammar (PSC grammar)

- special case of SC grammar
- every $(A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n)$ satisfies $x_1, \ldots, x_n \in V^+$

SC Grammars – Generated Language

Derivation step

If

- $(A_1, \dots, A_n) \to (x_1, \dots, x_n) \in P$
- $u = u_1 A_1 \dots u_n A_n u_{n+1}$
- $v = u_1 x_1 \dots u_n x_n u_{n+1}$

then $u \Rightarrow v \ [(A_1, \dots, A_n) \to (x_1, \dots, x_n)]$

Generated language

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

Generative power

- $\mathscr{L}(SCG) = \mathscr{L}(RE)$
- $\mathscr{L}(CF) \subset \mathscr{L}(PSCG) \subseteq \mathscr{L}(CS)$

SC Grammars – Example

Example

$$G_1 = (V_1, T_1, P_1, S),$$

where

$$V_1 = \{a, b, c, A, B, C, S\}, T_1 = \{a, b, c\},$$

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

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

G_1 is a SC grammar
G_1 is not a PSC grammar

Production Labels I

- for every grammar, G, there is a set of production labels
- we denote them $lab(G)$
- every $p \in lab(G)$ uniquely identifies one production
- we write $p : (A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n)$

Example

$G_2 = (\{S, A, B, C, a, b, c\}, \{a, b, c\}, P_2, S)$
$lab(G_2) = \{1, 2, 3\}$

$$\begin{aligned} P_2 = \{1 &: (S) \rightarrow (ABC), \\ 2 &: (A, B, C) \rightarrow (aA, bB, cC), \\ 3 &: (A, B, C) \rightarrow (\epsilon, \epsilon, \epsilon)\} \end{aligned}$$

$L(G_2) = \{a^n b^n c^n \ : \ n \geq 0\}$

Production Labels II

- to express that $x \Rightarrow y$ by $p : (A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n)$, we write $x \Rightarrow y\ [p]$

Example

$S \Rightarrow ABC\ [1] \Rightarrow aAbBcC\ [2] \Rightarrow aaAbbBccC\ [2] \Rightarrow aabbcc\ [3]$ in G_2

- to express that $x \Rightarrow^* y$ by productions labeled with $p_1, \ldots, p_n$, we write $x \Rightarrow^* y\ [p_1 \ldots p_n]$
- $p_1 \ldots p_n \in lab(G)^*$

Example

$S \Rightarrow^* aabbcc\ [1223]$ in G_2
$1223 \in lab(G_2)^*$

Proper Generator of its Sentences with Their Parses I

Parse

If $S \Rightarrow^* x\ [\rho], x \in T^*, \rho \in lab(G)^*$, then x is a sentence generated by G according to parse ρ

Example

aabbcc is a sentence generated according to parse 1223 in G_2

Proper generator of its sentences with their parses

- G is a proper generator of its sentences with their parses if $L(G) = \{x : x = y\rho, y \in (T - lab(G))^*, \rho \in lab(G)^*, S \Rightarrow^* x\ [\rho]\}$

Proper Generator of its Sentences with Their Parses II

Example

$G_3 = (\{S, A, B, C, a, b, c, 1, 2, 3, \$\}, \{a, b, c, 1, 2, 3\}, P_3, S)$
$lab(G_3) = \{1, 2, 3\}$

$$
\begin{aligned}
P_3 = \{1 &: (S) \rightarrow (ABC1\$) \\
2 &: (A, B, C, \$) \rightarrow (aA, bB, cC, 2\$) \\
3 &: (A, B, C, \$) \rightarrow (\epsilon, \epsilon, \epsilon, 3)\}
\end{aligned}
$$

$S \Rightarrow \ ABC1\$ \ [1] \Rightarrow aAbBcC12\$ \ [2] \Rightarrow aaAbbBccC122\$ \ [2] \Rightarrow$
$aabbcc1223 \ [3]$
$S \Rightarrow^* aabbcc1223 \ [1223]$

$L(G_3) = \{a^n b^n c^n \rho \, : \, n \geq 0, S \Rightarrow^* a^n b^n c^n \rho \, [\rho], \rho = 12^n 3\}$

G_3 is a proper generator of its sentences with their parses

Theorem 1

- let $G = (V, T, P, S)$ be a proper generator of its sentences with their parses
- we define the weak identity π from V^* to $(V - lab(G))^*$ as
  - $\pi(a) = a$ for every $a \in (V - lab(G))$
  - $\pi(p) = \epsilon$ for every $p \in lab(G)$

Example

$\pi(aabbcc1223) = aabbcc$ in G_3

Theorem

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

Leftmost derivation step in SC grammars

Derivation step in SC grammars

If

$$\begin{aligned}&(A_1, \dots, A_n) \rightarrow (x_1, \dots, x_n) \in P,\\ &u = u_1 A_1 \dots u_n A_n u_{n+1},\\ &v = u_1 x_1 \dots u_n x_n u_{n+1},\end{aligned}$$

then $u \Rightarrow v \; [(A_1, \dots, A_n) \rightarrow (x_1, \dots, x_n)]$

- $alph(w)$ denotes the set of all symbols occuring in w

Example

$alph(bacaab) = \{a, b, c\}$

Leftmost derivation step in SC grammars

every $A_i \notin alph(u_i)$ for all $1 \leq i \leq n$

Language Generated in a Leftmost Way

Language generated in a leftmost way

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

- and every step in every generation of $x \in T^*$ is leftmost

Proper leftmost generator of its sentences with their parses

$L(G) = \{x : x = y\rho, y \in (T - lab(G))^*, \rho \in lab(G)^*, S \Rightarrow^* x\ [\rho]\}$

- and G generates $L(G)$ in a leftmost way

Language Generated in a Leftmost Way – Example

Example

$G_4 = (\{S, A, B, C, a, b, c, 1, 2, 3, 4, \$\}, P_4, S, \{a, b, c, 1, 2, 3, 4\})$
$lab(G_4) = \{1, 2, 3, 4\}$

$$\begin{aligned} P_4 = \{ & 1 : (S) \rightarrow (ABC1\$), \\ & 2 : (A, B, C, \$) \rightarrow (AA, BB, CC, 2\$), \\ & 3 : (A, B, C, \$) \rightarrow (a, b, c, 3\$), \\ & 4 : (A, B, C, \$) \rightarrow (\epsilon, \epsilon, \epsilon, 4)\} \end{aligned}$$

$S \Rightarrow ABC1\$ \ [1] \Rightarrow AABBCC12\$ \ [2] \Rightarrow AabBCc123\$ \ [3] \Rightarrow$
$AAabBBCCc1232\$ \ [2] \Rightarrow aAabBbcCc12323\$ \ [3] \Rightarrow$
$aabbcc123234\$ \ [4]$
$S \Rightarrow^* aabbcc123234 \ [123234]$
$L(G_4) = \{a^n b^n c^n \rho \,:\, n \geq 0, S \Rightarrow^* a^n b^n c^n \rho \, [\rho]\}$
G_4 is a proper generator of its sentences with their parses
G_4 is not a proper leftmost generator of its sentences with their parses

Theorem 2

- let $G = (V, T, P, S)$ be a proper generator of its sentences with their parses
- we define the weak identity π from V^* to $(V - lab(G))^*$ as
  - $\pi(a) = a$ for every $a \in (V - lab(G))$
  - $\pi(p) = \epsilon$ for every $p \in lab(G)$

Example

$\pi(aabbcc123234) = aabbcc$ in G_4

Theorem

For every recursively enumerable language, L, there exists a PSC grammar, G, such that G contains no more than six nonterminals, G is a proper leftmost generator of its sentences with their parses and $L = \pi(L(G))$.

Queue Grammar

- we represent the recursively enumerable language by a queue grammar

Queue Grammar $G = (V, T, W, F, s, P)$

V is a finite alphabet of symbols

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

W is a finite alphabet of states

F is a set of final states, $F \subset W$

s is a starting string, $s \in (V - T)(W - F)$

P is a finite set of productions of the form: (a, b, x, c)

- **a** $\in V$
- **b** $\in (W - F)$
- **x** $\in V^*$
- **c** $\in W$

Queue Grammar – Derivation Step

Derivation Step

If $u = arb$, $v = rxc$, $a \in V$, $r, x \in V^*$, $b, c \in W$, and $(a, b, x, c) \in P$, then $u \Rightarrow v\ [(a, b, x, c)]$.

Generated Language

$L(G) = \{w : s \Rightarrow^* wf, w \in T^*, f \in F\}$

Generative Power

$\mathscr{L}(QG) = \mathscr{L}(RE)$

Lemma

For every QG there exists an equivalent QG which generates every string so that it first uses only productions rewriting symbols over $(V - T)^*$, and then only symbols over T^*.

Queue Grammar – Example

Example

$G_5 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_5)$

$$
\begin{aligned}
P_5 = \{ \ & 1 : (A, \bar{e}, bAa, \bar{e}), \\
& 2 : (A, \bar{e}, \varepsilon, \bar{f}), \\
& 3 : (a, \bar{e}, a, \bar{e}), \\
& 4 : (b, \bar{e}, b, \bar{e})\}
\end{aligned}
$$

$A\bar{e} \Rightarrow bAa\bar{e}$ [1] $\Rightarrow Aab\bar{e}$ [4] $\Rightarrow abbAa\bar{e}$ [1] $\Rightarrow bbAaa\bar{e}$ [3] $\Rightarrow bAaab\bar{e}$ [4] $\Rightarrow Aaabb\bar{e}$ [4] $\Rightarrow aabb\bar{f}$ [2]

$L(G_5) = \{a^n b^n \,:\, n \geq 0\}$

Proof Sketch

Basic idea

1. represent the recursively enumerable language by a QG
2. initiate the derivation
3. simulate QG by PSC grammar
   1. simulate generation of words from $(V - T)^*$
   2. simulate generation of words from T^+
4. check if the simulation was correct
5. complete the derivation

- every production has to add its label to the sentential form to create the parse in the correct order
- generated sentence has to precede this parse

QG Simulation – Example

Example

$G_5 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_5)$

$$
\begin{aligned}
P_5 = \{\ & 1 : (A, \bar{e}, bAa, \bar{e}), \\
& 2 : (A, \bar{e}, \varepsilon, \bar{f}), \\
& 3 : (a, \bar{e}, a, \bar{e}), \\
& 4 : (b, \bar{e}, b, \bar{e})\}
\end{aligned}
$$

Queue	A
States	$\bar{e}$
Productions	

QG Simulation – Example

Example

$G_5 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_5)$

$$
\begin{aligned}
P_5 = \{\ & 1 : (A, \bar{e}, bAa, \bar{e}), \\
& 2 : (A, \bar{e}, \varepsilon, \bar{f}), \\
& 3 : (a, \bar{e}, a, \bar{e}), \\
& 4 : (b, \bar{e}, b, \bar{e})\}
\end{aligned}
$$

Queue	A	b	A	a
States	$\bar{e}$	$\bar{e}$		
Productions	1			

QG Simulation – Example

Example

$G_5 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_5)$

$$
\begin{aligned}
P_5 = \{\ & 1 : (A, \bar{e}, bAa, \bar{e}), \\
& 2 : (A, \bar{e}, \varepsilon, \bar{f}), \\
& 3 : (a, \bar{e}, a, \bar{e}), \\
& 4 : (b, \bar{e}, b, \bar{e})\}
\end{aligned}
$$

Queue	A	b	A	a	b
States	$\bar{e}$	$\bar{e}$	$\bar{e}$		
Productions	1	4			

QG Simulation – Example

Example

$G_5 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_5)$

$$
\begin{aligned}
P_5 = \{\ & 1 : (A, \bar{e}, bAa, \bar{e}), \\
& 2 : (A, \bar{e}, \varepsilon, \bar{f}), \\
& 3 : (a, \bar{e}, a, \bar{e}), \\
& 4 : (b, \bar{e}, b, \bar{e})\}
\end{aligned}
$$

Queue	A	b	A	a	b	b	A	a
States	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$				
Productions	1	4	1					

QG Simulation – Example

Example

$G_5 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_5)$

$$
\begin{aligned}
P_5 = \{\ & 1 : (A, \bar{e}, bAa, \bar{e}), \\
& 2 : (A, \bar{e}, \varepsilon, \bar{f}), \\
& 3 : (a, \bar{e}, a, \bar{e}), \\
& 4 : (b, \bar{e}, b, \bar{e})\}
\end{aligned}
$$

Queue	A	b	A	a	b	b	A	a	a
States	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$				
Productions	1	4	1	3					

QG Simulation – Example

Example

$G_5 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_5)$

$$P_5 = \{\ 1 : (A, \bar{e}, bAa, \bar{e}),\\ 2 : (A, \bar{e}, \varepsilon, \bar{f}),\\ 3 : (a, \bar{e}, a, \bar{e}),\\ 4 : (b, \bar{e}, b, \bar{e})\}$$

Queue	A	b	A	a	b	b	A	a	a	b
States	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$				
Productions	1	4	1	3	4					

QG Simulation – Example

Example

$G_5 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_5)$

$$
\begin{aligned}
P_5 = \{ \ & 1 : (A, \bar{e}, bAa, \bar{e}), \\
& 2 : (A, \bar{e}, \varepsilon, \bar{f}), \\
& 3 : (a, \bar{e}, a, \bar{e}), \\
& 4 : (b, \bar{e}, b, \bar{e})\}
\end{aligned}
$$

Queue	A	b	A	a	b	b	A	a	a	b	b
States	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$				
Productions	1	4	1	3	4	4					

QG Simulation – Example

Example

$G_5 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_5)$

$$
\begin{aligned}
P_5 = \{\ & 1 : (A, \bar{e}, bAa, \bar{e}),\\
& 2 : (A, \bar{e}, \varepsilon, \bar{f}),\\
& 3 : (a, \bar{e}, a, \bar{e}),\\
& 4 : (b, \bar{e}, b, \bar{e})\}
\end{aligned}
$$

Queue	A	b	A	a	b	b	A	a	a	b	b
States	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{f}$			
Productions	1	4	1	3	4	4	2				

QG Simulation – Example

Example

$G_5 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_5)$

$$
\begin{aligned}
P_5 = \{\ & 1 : (A, \bar{e}, bAa, \bar{e}),\\
& 2 : (A, \bar{e}, \varepsilon, \bar{f}),\\
& 3 : (a, \bar{e}, a, \bar{e}),\\
& 4 : (b, \bar{e}, b, \bar{e})\}
\end{aligned}
$$

Queue	A	b	A	a	b	b	A	a	a	b	b
States	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{f}$			
Productions	1	4	1	3	4	4	2				
Prod. (queue)	1,2	4	1,2	3	4	4	1,2				
Prod. (state)	1-4	1-4	1-4	1-3,4	1-4	1-4	1,2-4				
Simulated pr.	1	4	1	3	4	4	2				

QG Simulation – Example

Example

$G_5 = (\{A, a, b\}, \{a, b\}, \{\bar{e}, \bar{f}\}, \{\bar{f}\}, A\bar{e}, P_5)$

$$
\begin{aligned}
P_5 = \{\ & 1 : (A, \bar{e}, bAa, \bar{e}),\\
& 2 : (A, \bar{e}, \varepsilon, \bar{f}),\\
& 3 : (a, \bar{e}, a, \bar{e}),\\
& 4 : (b, \bar{e}, b, \bar{e})\}
\end{aligned}
$$

Queue	A	b	A	a	b	b	A	a	a	b	b
States	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{e}$	$\bar{f}$			
Productions	1	4	1	3	4	4	2				
Prod. (queue)	1,2	4	1,2	3	4	4	1,2				
Prod. (state)	1-4	1-4	1-4	1-3,4	1-4	1-4	1,2-4				
Simulated pr.	1	4	1	3	4	4	2				

Construction I

- $Q = (V, T, W, F, s, R)$, $L(Q) = L$
- α: injection from $lab(Q)$ to $\{\bar{0}\}^*\{\bar{1}\}$
- $f(a) = \{\alpha(r) : r : (a, b, x, c) \in R\}$ for all $a \in V$
- $g(b) = \{\alpha(r) : r : (a, b, x, c) \in R\}$ for all $b \in W$

Constructed PSC grammar

$G = (\{S, A, B, \#, \bar{0}, \bar{1}\} \cup T \cup \mathrm{lab}(G), T \cup \mathrm{lab}(G), P, S)$

- the construction of P and $lab(G)$ is demonstrated on the following slides

Construction II

Step 1 (initialization)

For every $\bar{a}_0 \in f(a_0)$, $\bar{q}_0 \in g(q_0)$ such that $s = a_0 q_0$, add

$\lfloor 1\bar{a}_0\bar{q}_0 \rfloor : (S) \rightarrow (A\lfloor 1\bar{a}_0\bar{q}_0 \rfloor AA\bar{q}_0 A\bar{a}_0 AB)$

Step 2 (simulation of Q's productions generating words over V-T)

For every $r : (a, b, x, d) \in R$, $x \in (V - T)^*$ and $d \in (W - F)$, $\bar{x} \in f(x)$, $\bar{d} \in g(d)$, add

$\lfloor 2r\bar{x}\bar{d} \rfloor : (A, A, A, A, A, B) \rightarrow (A, \lfloor 2r\bar{x}\bar{d} \rfloor A, \alpha(r)A, \bar{d}A, \bar{x}A, B)$

Step 3 (separation of steps 2 and 4)

Add

$\lfloor 3 \rfloor : (A, A, A, A, A, B) \rightarrow (A, \lfloor 3 \rfloor A, A, A, B, A)$

Construction III

Step 4 (simulation of Q's productions generating words over T)

For every $r : (a, b, c, d) \in R$, $c \in T$ and $d \in (W - F)$, $\bar{d} \in g(d)$, add

$\lfloor 4r\bar{d} \rfloor : (A, A, A, A, B, A) \rightarrow (cA, \lfloor 4r\bar{d} \rfloor A, \alpha(r)A, \bar{d}A, B, A)$

Step 5 (simulation of Q's final step)

For every $r : (a, b, c, d) \in R$, $c \in T$ and $d \in F$, add

$\lfloor 5r \rfloor : (A, A, A, A, B, A) \rightarrow (c, \lfloor 5r \rfloor A, \alpha(r)A, A, B, AA)$

Construction IV

Step 6 (simulation verification)

Add

$$\lfloor 6 \rfloor : (A, \bar{0}, A, \bar{0}, A, \bar{0}, B, A, A) \rightarrow (\lfloor 6 \rfloor, A, \#, A, \#, A, B, A, A),$$
$$\lfloor 7 \rfloor : (A, \bar{1}, A, \bar{1}, A, \bar{1}, B, A, A) \rightarrow (\lfloor 7 \rfloor, A, \#, A, \#, A, B, A, A)$$

Step 7 (finishing the derivation)

Add

$$\lfloor 8 \rfloor : (A, A, A, B, A, A) \rightarrow (\lfloor 8 \rfloor B, \#, \#, \#, \#, \#),$$
$$\lfloor 9 \rfloor : (B, \#) \rightarrow (\lfloor 9 \rfloor, B),$$
$$\lfloor 10 \rfloor : (B) \rightarrow (\lfloor 10 \rfloor)$$

Theorem 3

- $\rho((A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n)) = n$
- $\rho_{\max}(G) = \rho(p)$, $p \in P$, such that $\rho(p) \geq \rho(r)$ for all $r \in P$

Theorem
For every recursively enumerable language, L, there exists a PSC grammar, G, such that G is a proper leftmost generator of its sentences preceded by their parses, G contains no more than six nonterminals, $\rho_{\max}(G) = 4$, and $L = \pi(L(G))$.

Theorem
For every recursively enumerable language, L, there exists a PSC grammar, G, such that G is a proper leftmost generator of its sentences preceded by their parses, G contains no more than nine nonterminals, $\rho_{\max}(G) = 2$, and $L = \pi(L(G))$.

Conclusion

We have proved that

- for every RE there is a PSC grammar which generates its sentences with their parses
- there are canonical versions of these generators
- the number of needed nonterminals can be reduced

Future investigation

- which other grammars can be used as proper generators of their sentences with their parses?
  - grammar systems seem to be appropriate candidates
- is it possible to generate sentences together with other useful information (e.g. derivation trees)?