Scattered Context Generators of Sentences with Their Parses

Radim Kocman

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

Modern Theoretical Computer Science, 2014

Based on Scattered Context Grammars

Scattered Context Grammar (SCG)

$$G = (V, T, P, S)$$

V is finite alphabet

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

S is the start symbol, $S \in V - T$

P is a finite set of productions of the form

$$(A_1, ..., A_n) \to (x_1, ..., x_n)$$

where $A_1, \ldots, A_n \in V - T$, $x_1, \ldots, x_n \in V^*$

Propagating Scattered Context Grammar (PSCG)

where $A_1, \ldots, A_n \in V - T$, $x_1, \ldots, x_n \in V^+$

Generative Power of Scattered Context Grammars

Scattered Context Grammar

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

Propagating Scattered Context Grammar

$$\mathscr{L}(CF) \subset \mathscr{L}(PSC) \subseteq \mathscr{L}(CS)$$

Can we somehow simulate SCG with PSCG?

Answer

Yes

Basic Idea

Let G be SCG and PG be PSCG simulating G

1. Add new terminal symbol \$ to PG
2. Instead of ϵ generate \$
3. Move all \$ to the left (or right)

Generated Language

$$L(PG) = \{ \{\$\}^* x : x \in L(G) \}$$

Scattered Context Generators

Basic Idea

Instead of meaningless symbols $ add useful information - the parse.

What is the parse?

Assume that for every SCG $G = (V, T, P, S)$ there is a set of production labels, denoted by lab(G), such that $|lab(G)| = |P|$ and there is bijection from P to lab(G):

$$l : (A_1, ..., A_n) \rightarrow (x_1, ..., x_n) \text{ where } l \in lab(G) \text{ and } production \in P$$

Derivation step:

$$u \Rightarrow v\ [l] \text{ where } u, v \in V^* \text{ and } l \in lab(G)$$

Generated language:

$$L(G) = \{x \in T^* : S \Rightarrow^* x\ [p] \text{ and } p \in lab(G)^*\} \quad \text{p is the parse}$$

Scattered Context Generators

Generated Language

Let G be PSCG. Let $lab(G) \subseteq T$. G is a proper generator of its sentences with their parses if and only if

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

Theorem

For every recursively enumerable language L, there is a propagating scattered context grammar G such that G is a proper generator of its sentences with their parses, and $L = L(G)//lab(G)^+$.

Example Conversion of SCG to Proper Generator

Example SCG

$$G = (\{S, A, B, C, a, b, c\}, \{a, b, c\}, P, S)$$

where P:

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

Generated language:

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

Example derivations

$$\begin{aligned} &S \Rightarrow ABC \Rightarrow \epsilon \ [13] \\ &S \Rightarrow ABC \Rightarrow aAbBcC \Rightarrow abc \ [123] \end{aligned}$$

Basic Conversion Method

Input

$$G = (V, T, P, S)$$

General Generator

$$\overline{G} = (\overline{V}, \overline{T}, \overline{P}, \overline{S})$$

where:

- $\overline{V} = V \cup lab(\overline{G}) \cup \{\overline{S}, X, Y, Z, \$_1, \$_2, \$_3\} \cup \{\overline{a} : a \in T\}$
- $\overline{T} = T \cup lab(\overline{G})$

Basic Conversion Method

General Generator - Rules $\overline{P}$

Add these types of rules:

1. Start rules
2. Simulation rules
3. Stop rules
4. Shift rules
5. Erase rule
6. End rule

Start rules

Start rules

$$(\!|1|\!) : (\overline{S}) \rightarrow (X(\!|1|\!)\$_1 ZS)$$
$$(\!|1_\epsilon|\!) : (\overline{S}) \rightarrow ((\!|1_\epsilon|\!)\$_1 S)$$

Example derivation $S \Rightarrow ABC \Rightarrow \epsilon$

$$\overline{S} \Rightarrow (\!|1_\epsilon|\!)\$_1 S$$

Example derivation $S \Rightarrow ABC \Rightarrow aAbBcC \Rightarrow abc$

$$\overline{S} \Rightarrow X(\!|1|\!)\$_1 ZS$$

Simulation rules

- For each rule in P add simulation rule
- Convert terminals to nonterminals
- Convert ϵ to Y

$$\llparenthesis 1l \rrparenthesis : (\$_1, A_1, \ldots, A_n) \rightarrow (\llparenthesis 1l \rrparenthesis \$_1, x_1, \ldots, x_n)$$

Example simulation rules

$$\llparenthesis 11 \rrparenthesis : (\$_1, S) \rightarrow (\llparenthesis 11 \rrparenthesis \$_1, ABC)$$
$$\llparenthesis 12 \rrparenthesis : (\$_1, A, B, C) \rightarrow (\llparenthesis 12 \rrparenthesis \$_1, \overline{a}A, \overline{b}B, \overline{c}C)$$
$$\llparenthesis 13 \rrparenthesis : (\$_1, A, B, C) \rightarrow (\llparenthesis 13 \rrparenthesis \$_1, Y, Y, Y)$$

Simulation rules

Example derivation $S \Rightarrow ABC \Rightarrow \epsilon$

$$\begin{aligned} (\!|1_\epsilon|\!)\$_1 S &\Rightarrow (\!|1_\epsilon|\!)(\!|11|\!)\$_1 ABC \\ &\Rightarrow (\!|1_\epsilon|\!)(\!|11|\!)(\!|13|\!)\$_1 YYY \end{aligned}$$

Example derivation $S \Rightarrow ABC \Rightarrow aAbBcC \Rightarrow abc$

$$\begin{aligned} X(\!|1|\!)\$_1 ZS &\Rightarrow X(\!|1|\!)(\!|11|\!)\$_1 ZABC \\ &\Rightarrow X(\!|1|\!)(\!|11|\!)(\!|12|\!)\$_1 Z\overline{a}A\overline{b}B\overline{c}C \\ &\Rightarrow X(\!|1|\!)(\!|11|\!)(\!|12|\!)(\!|13|\!)\$_1 Z\overline{a}Y\overline{b}Y\overline{c}Y \end{aligned}$$

Stop rules

$$(\!|2|\!) : (\$_1) \rightarrow ((\!|2|\!)\$_2)$$
$$(\!|2_\epsilon|\!) : (\$_1) \rightarrow ((\!|2_\epsilon|\!)\$_3)$$

Example derivation $S \Rightarrow ABC \Rightarrow \epsilon$

$$(\!|\ldots|\!)\$_1 YYY \Rightarrow (\!|\ldots|\!)\$_3 YYY$$

Example derivation $S \Rightarrow ABC \Rightarrow aAbBcC \Rightarrow abc$

$$X(\!|\ldots|\!)\$_1 Z\overline{a}Y\overline{b}Y\overline{c}Y \Rightarrow X(\!|\ldots|\!)\$_2 Z\overline{a}Y\overline{b}Y\overline{c}Y$$

Shift rules

Shift rules

For each $a \in T$, add

$$\llparenthesis 2a \rrparenthesis : (X, \$_2, Z, \bar{a}) \rightarrow (aX, \llparenthesis 2a \rrparenthesis \$_2, Y, Z)$$
$$\llparenthesis 3a \rrparenthesis : (X, \$_2, Z, \bar{a}) \rightarrow (a, \llparenthesis 3a \rrparenthesis \$_3, Y, Y)$$

Example simulation rules

$$\llparenthesis 2a \rrparenthesis : (X, \$_2, Z, \bar{a}) \rightarrow (aX, \llparenthesis 2a \rrparenthesis \$_2, Y, Z)$$
$$\llparenthesis 2b \rrparenthesis : (X, \$_2, Z, \bar{b}) \rightarrow (bX, \llparenthesis 2b \rrparenthesis \$_2, Y, Z)$$
$$\llparenthesis 2c \rrparenthesis : (X, \$_2, Z, \bar{c}) \rightarrow (cX, \llparenthesis 2c \rrparenthesis \$_2, Y, Z)$$
$$\llparenthesis 3a \rrparenthesis : (X, \$_2, Z, \bar{a}) \rightarrow (a, \llparenthesis 3a \rrparenthesis \$_3, Y, Y)$$
$$\llparenthesis 3b \rrparenthesis : (X, \$_2, Z, \bar{b}) \rightarrow (b, \llparenthesis 3b \rrparenthesis \$_3, Y, Y)$$
$$\llparenthesis 3c \rrparenthesis : (X, \$_2, Z, \bar{c}) \rightarrow (c, \llparenthesis 3c \rrparenthesis \$_3, Y, Y)$$

Shift rules

Example derivation $S \Rightarrow ABC \Rightarrow aAbBcC \Rightarrow abc$

$$\begin{aligned} X(\!|\ldots|\!)\$_2 Z\bar{a}Y\bar{b}Y\bar{c}Y &\Rightarrow aX(\!|\ldots|\!)\$_2 YZY\bar{b}Y\bar{c}Y \\ &\Rightarrow abX(\!|\ldots|\!)\$_2 YYYZY\bar{c}Y \\ &\Rightarrow abc(\!|\ldots|\!)\$_3 YYYYYYY \end{aligned}$$

Erase rule

Erase rule

$$(\!|3|\!) : (\$_3, Y) \rightarrow ((\!|3|\!), \$_3)$$

Example derivation $S \Rightarrow ABC \Rightarrow \epsilon$

$$\begin{aligned} (\!|\ldots|\!)\$_3 YYY &\Rightarrow (\!|\ldots|\!)\$_3 YY \\ &\Rightarrow (\!|\ldots|\!)\$_3 Y \\ &\Rightarrow (\!|\ldots|\!)\$_3 \end{aligned}$$

Example derivation $S \Rightarrow ABC \Rightarrow aAbBcC \Rightarrow abc$

$$\begin{aligned} abc(\!|\ldots|\!)\$_3 YYYYYYY &\Rightarrow abc(\!|\ldots|\!)\$_3 YYYYYY \\ &\ldots \\ &\Rightarrow abc(\!|\ldots|\!)\$_3 \end{aligned}$$

End rule

End rule

$$(\!|4|\!) : (\$_3) \rightarrow ((\!|4|\!))$$

Example derivation $S \Rightarrow ABC \Rightarrow \epsilon$

$$(\!|\ldots|\!)\$_3 \Rightarrow (\!|\ldots|\!)(\!|4|\!)$$

Example derivation $S \Rightarrow ABC \Rightarrow aAbBcC \Rightarrow abc$

$$abc(\!|\ldots|\!)\$_3 \Rightarrow abc(\!|\ldots|\!)(\!|4|\!)$$

Result for derivation $S \Rightarrow ABC \Rightarrow \epsilon$

Result

$$(\!|1_\epsilon|\!)(\!|11|\!)(\!|13|\!)(\!|2_\epsilon|\!)(\!|3|\!)(\!|3|\!)(\!|3|\!)(\!|4|\!)$$

Used rules

1. Start rules - $(\!|1_\epsilon|\!)$
2. Simulation rules - $(\!|11|\!)(\!|13|\!)$
3. Stop rules - $(\!|2_\epsilon|\!)$
4. Shift rules - None
5. Erase rule - $(\!|3|\!)(\!|3|\!)(\!|3|\!)$
6. End rule - $(\!|4|\!)$

Result for derivation $S \Rightarrow ABC \Rightarrow aAbBcC \Rightarrow abc$

Result

$$abc(\!|1|\!)(\!|11|\!)(\!|12|\!)(\!|13|\!)(\!|2|\!)(\!|2a|\!)(\!|2b|\!)(\!|3c|\!)(\!|3|\!)(\!|3|\!)(\!|3|\!)(\!|3|\!)(\!|3|\!)(\!|3|\!)(\!|3|\!)(\!|4|\!)$$

Used rules

1. Start rules - $(\!|1|\!)$
2. Simulation rules - $(\!|11|\!)(\!|12|\!)(\!|13|\!)$
3. Stop rules - $(\!|2|\!)$
4. Shift rules - $(\!|2a|\!)(\!|2b|\!)(\!|3c|\!)$
5. Erase rule - $(\!|3|\!)(\!|3|\!)(\!|3|\!)(\!|3|\!)(\!|3|\!)(\!|3|\!)(\!|3|\!)$
6. End rule - $(\!|4|\!)$

Canonical Generators

Theorem - Leftmost Generator

For every recursively enumerable language L, there is a propagating scattered context grammar G such that G is a proper leftmost generator of its sentences with their parses, and $L = L(G)//lab(G)^{+}$.

Theorem - Rightmost Generator

For every recursively enumerable language L, there is a propagating scattered context grammar G such that G is a proper rightmost generator of its sentences with their parses, and $L = L(G)//lab(G)^{+}$.

Reduced Generators

Theorem - Minimal number of nonterminals

For every recursively enumerable language L, there is a propagating scattered context grammar $G = (V, T, P, S)$ such that G is a proper leftmost generator of its sentences preceded by their parses, $|V - T| \leq 6$, $mcs(G) = 3$, and $L = lab(G)^+ \backslash\backslash L(G)$.

Theorem - Minimal context-dependency

For every recursively enumerable language L, there is a propagating scattered context grammar $G = (V, T, P, S)$ such that G is a proper leftmost generator of its sentences preceded by their parses, $|V - T| \leq 9$, $mcs(G) = 1$, and $L = lab(G)^+ \backslash\backslash L(G)$.

Conclusion

Main advantages

Scattered Context Generators can

- simulate any SCG with PSCG, which is less powerful
- generate additional useful information

Bibliography

A. Meduna and J. Techet.
Canonical scattered context generators of sentences with their parses.
Theoretical Computer Science, 389:73–81, 2007.

A. Meduna and J. Techet.
Scattered Context Grammars and their Applications.
WIT Press, Southampton, Boston, 2010.