Parses and Their Generators in Scattered Context Grammars

Stanislav Židek

Department of Information Systems
Faculty of Information Technology
Brno University of Technology

Modern Theoretical Computer Science, 2009

Contents

1 Introduction

2 Terminology
- Scattered Context Grammars
- Language Generated by SCG
- Generators of Sentences with Their Parses

3 Result
- Theorem – $\mathcal{L}(RE)$ by PSCG-P
- Construction of SCG-P from a SCG (Idea)
- Formal Algorithm

4 Summary

Motivation

- parsing is important part of formal language theory
- parse
  - sequence of rules that generated a sentence
  - useful piece of information
- obtaining parses by using scattered context grammars

Previous Research

- characterization of RE languages L by propagating scattered context grammars using “stuffing terminals” [2]
- sentences followed by their parses in matrix grammars (Extended Szilard languages) [3]

Scattered Context Grammar (SCG)

Definition

SCG is a 4-tuple $G = (V, T, P, S)$, where

- V is a finite set of symbols
- T is a finite set of terminals, $T \subseteq V$
- S is a starting symbol, $S \in (V - T)$
- P is a finite set of productions of the form

$$r : (A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n),$$

where

- $A_1, \ldots, A_n \in (V - T)$ and $x_1, \ldots, x_n \in V^*$
- r is a unique production label.

Propagating Scattered Context Grammar (PSCG)

Definition

PSCG is a 4-tuple $G = (V, T, P, S)$, where

- V, T, S is the same as in SCG
- P is a finite set of productions of the form

$$r : (A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n),$$

where

- $A_1, \ldots, A_n \in (V - T)$ and $x_1, \ldots, x_n \in V^+$
- r is a unique production label.

Note

lab(G) – set of all production labels of grammar G

Relation of Direct Derivation ($\Rightarrow$)

Definition

Let $G = (V, T, P, S)$ be a SCG. Suppose that

- $r : (A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n) \in P$
- $u = w_0 A_1 w_1 \ldots w_{n-1} A_n w_n \in V^*$
- $v = w_0 x_1 w_1 \ldots w_{n-1} x_n w_n \in V^*$

G directly derives v from u according to production r

$$u \Rightarrow v \; [r] \qquad \text{(or simply } u \Rightarrow v\text{)}$$

Reflexive and Transitive Closure of $\Rightarrow$

$\Rightarrow^*$ – reflexive and transitive closure of $\Rightarrow$

Suppose that $v \Rightarrow w_1 \; [p_1] \Rightarrow \ldots \Rightarrow w_{n-1} \; [p_{n-1}] \Rightarrow u \; [p_n]$; we write

$$u \Rightarrow^* v \; [p_1 p_2 \ldots p_n]$$

Generated Language

Definition

Let $G = (V, T, P, S)$ be a SCG.

$$L(G) = \{w : w \in T^* \wedge S \Rightarrow^* w\ [p_1 p_2 \ldots p_n]\}$$

$p_1 p_2 \ldots p_n$ – parse (derivation word, Szilard word, control word)

Generative Power

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

SCG Generating Sentences with Their Parses

Definition

Let $G = (V, T, P, S)$ be a SCG. G is a proper generator of its sentences with their parses (SCG-P) iff

$$L(G) = \{w\rho : w\rho \in T^* \wedge S \Rightarrow^* w\rho\ [\rho]\}$$

First Example

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

$$\begin{aligned} P_1 = \{ & 1 : (S) \rightarrow (ABC), \\ & 2 : (A, B, C) \rightarrow (Aa, Bb, Cc), \\ & 3 : (A, B, C) \rightarrow (\varepsilon, \varepsilon, \varepsilon)\} \end{aligned}$$

Is G_1 SCG-P?

SCG Generating Sentences with Their Parses

Definition

Let $G = (V, T, P, S)$ be a SCG. G is a proper generator of its sentences with their parses (SCG-P) iff

$$L(G) = \{w\rho : w\rho \in T^* \wedge S \Rightarrow^* w\rho\ [\rho]\}$$

First Example

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

$$\begin{aligned} P_1 = \{&1 : (S) \rightarrow (ABC), \\ &2 : (A, B, C) \rightarrow (Aa, Bb, Cc), \\ &3 : (A, B, C) \rightarrow (\varepsilon, \varepsilon, \varepsilon)\} \end{aligned}$$

Is G_1 SCG-P?

SCG Generating Sentences with Their Parses

Definition

Let $G = (V, T, P, S)$ be a SCG. G is a proper generator of its sentences with their parses (SCG-P) iff

$$L(G) = \{w\rho : w\rho \in T^* \wedge S \Rightarrow^* w\rho\ [\rho]\}$$

First Example

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

$$\begin{aligned} P_1 = \{&1 : (S) \rightarrow (ABC), \\ &2 : (A, B, C) \rightarrow (Aa, Bb, Cc), \\ &3 : (A, B, C) \rightarrow (\varepsilon, \varepsilon, \varepsilon)\} \end{aligned}$$

Is G_1 SCG-P? **NO!** $(\mathrm{lab}(G_1) \not\subseteq T)$

SCG-P Examples II

Second Example

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

$$\begin{aligned} P_2 = \{ & 1 : (S) \rightarrow (ABC1\$), \\ & 2 : (A, B, C, \$) \rightarrow (Aa, Bb, Cc, 2\$), \\ & 3 : (A, B, C, \$) \rightarrow (\varepsilon, \varepsilon, \varepsilon, 3)\} \end{aligned}$$

Is G_2 SCG-P?

SCG-P Examples II

Second Example

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

$$
\begin{aligned}
P_2 = \{ & 1 : (S) \rightarrow (ABC1\$), \\
& 2 : (A, B, C, \$) \rightarrow (Aa, Bb, Cc, 2\$), \\
& 3 : (A, B, C, \$) \rightarrow (\varepsilon, \varepsilon, \varepsilon, 3)\}
\end{aligned}
$$

Is G_2 SCG-P? **YES!**

SCG-P Examples II

Second Example

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

$$\begin{aligned} P_2 = \{ & 1 : (S) \rightarrow (ABC1\$), \\ & 2 : (A, B, C, \$) \rightarrow (Aa, Bb, Cc, 2\$), \\ & 3 : (A, B, C, \$) \rightarrow (\varepsilon, \varepsilon, \varepsilon, 3)\} \end{aligned}$$

Is G_2 SCG-P? **YES!**

Example G_2 derivation

$$\begin{aligned} S &\Rightarrow ABC1\$ \ [1] \Rightarrow AaBbCc12\$ \ [2] \\ &\Rightarrow AaaBbbCcc122\$ \ [2] \Rightarrow aabbcc1223 \ [3] \end{aligned}$$

SCG-P Examples II

Second Example

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

$$\begin{aligned} P_2 = \{ & 1 : (S) \rightarrow (ABC1\$), \\ & 2 : (A, B, C, \$) \rightarrow (Aa, Bb, Cc, 2\$), \\ & 3 : (A, B, C, \$) \rightarrow (\varepsilon, \varepsilon, \varepsilon, 3)\} \end{aligned}$$

Is G_2 SCG-P? **YES!**

Example G_2 derivation

$$\begin{aligned} S &\Rightarrow ABC1\$ \ [1] \Rightarrow AaBbCc12\$ \ [2] \\ &\Rightarrow AaaBbbCcc122\$ \ [2] \Rightarrow aabbcc1223 \ [3] \end{aligned}$$

SCG-P Examples II

Second Example

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

$$\begin{aligned} P_2 = \{ & 1 : (S) \rightarrow (ABC1\$), \\ & 2 : (A, B, C, \$) \rightarrow (Aa, Bb, Cc, 2\$), \\ & 3 : (A, B, C, \$) \rightarrow (\varepsilon, \varepsilon, \varepsilon, 3)\} \end{aligned}$$

Is G_2 SCG-P? **YES!**

Example G_2 derivation

$$S \Rightarrow ABC1\$ \ [1] \Rightarrow AaBbCc12\$ \ [2]$$
$$\Rightarrow AaaBbbCcc122\$ \ [2] \Rightarrow aabbcc1223 \ [3]$$

SCG-P Examples II

Second Example

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

$$\begin{aligned} P_2 = \{ & 1 : (S) \rightarrow (ABC1\$), \\ & 2 : (A, B, C, \$) \rightarrow (Aa, Bb, Cc, 2\$), \\ & 3 : (A, B, C, \$) \rightarrow (\varepsilon, \varepsilon, \varepsilon, 3)\} \end{aligned}$$

Is G_2 SCG-P? **YES!**

Example G_2 derivation

$$\begin{aligned} S &\Rightarrow ABC1\$ \ [1] \Rightarrow AaBbCc12\$ \ [2] \\ &\Rightarrow AaaBbbCcc122\$ \ [2] \Rightarrow aabbcc1223 \ [3] \end{aligned}$$

SCG-P Examples II

Second Example

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

$$\begin{aligned} P_2 = \{ & 1 : (S) \rightarrow (ABC1\$), \\ & 2 : (A, B, C, \$) \rightarrow (Aa, Bb, Cc, 2\$), \\ & 3 : (A, B, C, \$) \rightarrow (\varepsilon, \varepsilon, \varepsilon, 3)\} \end{aligned}$$

Is G_2 SCG-P? **YES!**

Example G_2 derivation

$$\begin{aligned} S &\Rightarrow ABC1\$ \ [1] \Rightarrow AaBbCc12\$ \ [2] \\ &\Rightarrow AaaBbbCcc122\$ \ [2] \Rightarrow aabbcc1223 \ [3] \end{aligned}$$

PSCG Generating Sentences with Their Parses

Theorem

For every recursively enumerable language L there exists a propagating SCG-P G such that

$$L(G) = \bigcup_{w \in L} w\rho : w\rho \in T^* \wedge S \Rightarrow^* w\rho\ [\rho]$$

Proof Idea

Basic Concept

1. for every RE language L there is SCG G such that L(G)=L
   - $\mathcal{L}(RE) = \mathcal{L}(SCG)$ (see references [1])
2. for every SCG G there is a propagating SCG-P G' generating sentences of *G* followed by their parses
   - construction will be shown

Phase I – Sentence Generation

Simultaneously generate sequence of productions by $\$_1$ and sequential forms with marked non-terminals (ε on the right hand side replaced by Y)

$$
\begin{aligned}
S' &\Rightarrow X1\$_1 ZS \\
&\Rightarrow X12_1\$_1 ZABC && [1 : (S) \rightarrow (ABC)] \\
&\Rightarrow X12_1 2_2\$_1 ZAa' Bb' Cc' && [2 : (A, B, C) \rightarrow (Aa, Bb, Cc)] \\
&\Rightarrow X12_1 2_2 2_3\$_1 ZYa' Yb' Yc' && [3 : (A, B, C) \rightarrow (\varepsilon, \varepsilon, \varepsilon)]
\end{aligned}
$$

Phase I – Sentence Generation

Simultaneously generate sequence of productions by $\$_1$ and sequential forms with marked non-terminals (ε on the right hand side replaced by Y)

$$S' \Rightarrow X1\$_1ZS$$

$$\Rightarrow X1{2_1}\$_1ZABC \qquad [1 : (S) \rightarrow (ABC)]$$

$$\Rightarrow X1{2_1}{2_2}\$_1ZAa'Bb'Cc' \qquad [2 : (A, B, C) \rightarrow (Aa, Bb, Cc)]$$

$$\Rightarrow X1{2_1}{2_2}{2_3}\$_1ZYa'Yb'Yc' \qquad [3 : (A, B, C) \rightarrow (\varepsilon, \varepsilon, \varepsilon)]$$

Phase I – Sentence Generation

Simultaneously generate sequence of productions by $\$_1$ and sequential forms with marked non-terminals (ε on the right hand side replaced by Y)

$$
\begin{aligned}
S' &\Rightarrow X1\$_1 ZS \\
&\Rightarrow X12_1\$_1 ZABC && [1 : (S) \rightarrow (ABC)] \\
&\Rightarrow X12_1 2_2\$_1 ZAa' Bb' Cc' && [2 : (A, B, C) \rightarrow (Aa, Bb, Cc)] \\
&\Rightarrow X12_1 2_2 2_3\$_1 ZYa' Yb' Yc' && [3 : (A, B, C) \rightarrow (\varepsilon, \varepsilon, \varepsilon)]
\end{aligned}
$$

Phase I – Sentence Generation

Simultaneously generate sequence of productions by $\$_1$ and sequential forms with marked non-terminals (ε on the right hand side replaced by Y)

$$
\begin{aligned}
S' &\Rightarrow X1\$_1 ZS \\
&\Rightarrow X12_1\$_1 ZABC && [1 : (S) \rightarrow (ABC)] \\
&\Rightarrow X12_1 2_2\$_1 ZAa'Bb'Cc' && [2 : (A, B, C) \rightarrow (Aa, Bb, Cc)] \\
&\Rightarrow X12_1 2_2 2_3\$_1 ZYa'Yb'Yc' && [3 : (A, B, C) \rightarrow (\varepsilon, \varepsilon, \varepsilon)]
\end{aligned}
$$

Phase I – Sentence Generation

Simultaneously generate sequence of productions by $\$_1$ and sequential forms with marked non-terminals (ε on the right hand side replaced by Y)

$$
\begin{aligned}
S' &\Rightarrow X1\$_1 ZS \\
&\Rightarrow X12_1\$_1 ZABC && [1 : (S) \rightarrow (ABC)] \\
&\Rightarrow X12_1 2_2\$_1 ZAa' Bb' Cc' && [2 : (A, B, C) \rightarrow (Aa, Bb, Cc)] \\
&\Rightarrow X12_1 2_2 2_3\$_1 ZYa' Yb' Yc' && [3 : (A, B, C) \rightarrow (\varepsilon, \varepsilon, \varepsilon)]
\end{aligned}
$$

SCG-P Construction Idea II

Phase II

Phase II – Generating Terminals

Move marked non-terminals before the parse (replace them by Y)

$$
\begin{aligned}
& X12_1 2_2 2_3 \$_1 ZYa' Yb' Yc' \\
\Rightarrow\ & X12_1 2_2 2_3 2\$_2 ZYa' Yb' Yc' \\
\Rightarrow\ & aX12_1 2_2 2_3 23_a \$_2 YYZYb' Yc' \\
\Rightarrow\ & abX12_1 2_2 2_3 23_a 3_b \$_2 YYYYZYc' \\
\Rightarrow\ & abc12_1 2_2 2_3 23_a 3_b 3_c \$_3 YYYYYYY
\end{aligned}
$$

SCG-P Construction Idea II

Phase II

Phase II – Generating Terminals

Move marked non-terminals before the parse (replace them by Y)

$$
\begin{aligned}
& X12_1 2_2 2_3 \$_1 ZYa' Yb' Yc' \\
\Rightarrow\ & X12_1 2_2 2_3 2\$_2 ZY a' Yb' Yc' \\
\Rightarrow\ & aX12_1 2_2 2_3 23_a \$_2 YYZYb' Yc' \\
\Rightarrow\ & abX12_1 2_2 2_3 23_a 3_b \$_2 YYYYZYc' \\
\Rightarrow\ & abc12_1 2_2 2_3 23_a 3_b 3_c \$_3 YYYYYYY
\end{aligned}
$$

SCG-P Construction Idea II

Phase II

Phase II – Generating Terminals

Move marked non-terminals before the parse (replace them by Y)

$$
\begin{aligned}
& X12_1 2_2 2_3 \$_1 ZYa' Yb' Yc' \\
\Rightarrow\ & X12_1 2_2 2_3 2\$_2 ZY a' Yb' Yc' \\
\Rightarrow\ & aX12_1 2_2 2_3 23_a \$_2 YYZYb' Yc' \\
\Rightarrow\ & abX12_1 2_2 2_3 23_a 3_b \$_2 YYYYZYc' \\
\Rightarrow\ & abc12_1 2_2 2_3 23_a 3_b 3_c \$_3 YYYYYYY
\end{aligned}
$$

SCG-P Construction Idea II

Phase II

Phase II – Generating Terminals

Move marked non-terminals before the parse (replace them by Y)

$$
\begin{aligned}
& X12_1 2_2 2_3 \$_1 ZYa' Yb' Yc' \\
\Rightarrow\ & X12_1 2_2 2_3 2\$_2 ZY a' Yb' Yc' \\
\Rightarrow\ & aX12_1 2_2 2_3 23_a \$_2 YYZYb' Yc' \\
\Rightarrow\ & abX12_1 2_2 2_3 23_a 3_b \$_2 YYYYZYc' \\
\Rightarrow\ & abc12_1 2_2 2_3 23_a 3_b 3_c \$_3 YYYYYYY
\end{aligned}
$$

Phase II – Generating Terminals

Move marked non-terminals before the parse (replace them by Y)

$$
\begin{aligned}
& X12_1 2_2 2_3 \$_1 ZYa' Yb' Yc' \\
\Rightarrow\ & X12_1 2_2 2_3 2\$_2 ZY a' Yb' Yc' \\
\Rightarrow\ & aX12_1 2_2 2_3 23_a \$_2 YY Z Y b' Yc' \\
\Rightarrow\ & abX12_1 2_2 2_3 23_a 3_b \$_2 YYYY Z Y c' \\
\Rightarrow\ & abc12_1 2_2 2_3 23_a 3_b 3_c \$_3 YYYYYY Y
\end{aligned}
$$

Phase III

Removing Ys Replace Ys by labels

$$
\begin{aligned}
& abc12_1 2_2 2_3 2\$_3 YYYYYYY \\
\Rightarrow\ & abc12_1 2_2 2_3 23_a 3_b 3_c 3\$_3 YYYYYY \\
\Rightarrow\ & abc12_1 2_2 2_3 23_a 3_b 3_c 33\$_3 YYYYY \\
\Rightarrow\ & \ldots \\
\Rightarrow\ & abc12_1 2_2 2_3 23_a 3_b 3_c 3333333\$_3 \\
\Rightarrow\ & \underbrace{abc}_{\text{sentence}}\ \underbrace{12_1 2_2 2_3 23_a 3_b 3_c 33333334}_{\text{parse}}
\end{aligned}
$$

Phase III

Removing Ys Replace Ys by labels

$$
\begin{aligned}
& abc12_1 2_2 2_3 2\$_3 YYYYYYY \\
\Rightarrow\ & abc12_1 2_2 2_3 23_a 3_b 3_c 3\$_3 YYYYYY \\
\Rightarrow\ & abc12_1 2_2 2_3 23_a 3_b 3_c 33\$_3 YYYYY \\
\Rightarrow\ & \ldots \\
\Rightarrow\ & abc12_1 2_2 2_3 23_a 3_b 3_c 3333333\$_3 \\
\Rightarrow\ & \underbrace{abc}_{\text{sentence}}\ \underbrace{12_1 2_2 2_3 23_a 3_b 3_c 33333334}_{\text{parse}}
\end{aligned}
$$

Phase III

Removing Ys Replace Ys by labels

$$
\begin{aligned}
& abc12_12_22_32\$_3 YYYYYYY \\
\Rightarrow\ & abc12_12_22_323_a3_b3_c3\$_3 YYYYYY \\
\Rightarrow\ & abc12_12_22_323_a3_b3_c33\$_3 YYYY \\
\Rightarrow\ & \ldots \\
\Rightarrow\ & abc12_12_22_323_a3_b3_c3333333\$_3 \\
\Rightarrow\ & \underbrace{abc}_{\text{sentence}}\ \underbrace{12_12_22_323_a3_b3_c33333334}_{\text{parse}}
\end{aligned}
$$

Phase III

Removing Ys Replace Ys by labels

$$
\begin{aligned}
& abc12_12_22_32\$_3 YYYYYYY \\
\Rightarrow\ & abc12_12_22_323_a3_b3_c3\$_3 YYYYYY \\
\Rightarrow\ & abc12_12_22_323_a3_b3_c33\$_3 YYYY \\
\Rightarrow\ & \ldots \\
\Rightarrow\ & abc12_12_22_323_a3_b3_c3333333\$_3 \\
\Rightarrow\ & \underbrace{abc}_{\text{sentence}}\ \underbrace{12_12_22_323_a3_b3_c33333334}_{\text{parse}}
\end{aligned}
$$

Phase III

Removing Ys Replace Ys by labels

$$
\begin{aligned}
& abc12_12_22_32\$_3 YYYYYYY \\
\Rightarrow\ & abc12_12_22_323_a3_b3_c3\$_3 YYYYYY \\
\Rightarrow\ & abc12_12_22_323_a3_b3_c33\$_3 YYYY \\
\Rightarrow\ & \ldots \\
\Rightarrow\ & abc12_12_22_323_a3_b3_c3333333\$_3 \\
\Rightarrow\ & \underbrace{abc}_{\text{sentence}}\ \underbrace{12_12_22_323_a3_b3_c33333334}_{\text{parse}}
\end{aligned}
$$

SCG-P Formal Algorithm

Input and Output

Input: SCG $G = (V, T, P, S)$
Output: PSCG-P $G' = (V', T', P', S')$ such that

$$L(G') = \bigcup_{w \in L} w\rho : S \Rightarrow^* w\rho\ [\rho]$$

Preliminaries

Let $\Phi = \{a' : a \in T\}$. Define γ' and γ as follows:

$$\gamma'(a) = \begin{cases} a' & \text{if } a \in T \\ a & \text{if } a \in (V - T) \end{cases} \qquad \gamma(x) = \begin{cases} Y & \text{if } x = \varepsilon \\ \gamma'(x) & \text{otherwise} \end{cases}$$

SCG-P Formal Algorithm II

Symbols

Terminals

$$T' = T \cup \mathrm{lab}(G')$$

All Symbols

$$V' = T' \cup \Phi \cup \{S', X, Y, Z, \$_1, \$_2, \$_3\}$$

Note

Assume that $\mathrm{lab}(G') \cap \mathrm{alph}(L) = \emptyset$.

Productions

Construct P' as follows:

1. Add $1 : (S) \rightarrow (X1\$_1ZS)$ and $1_\varepsilon : (S) \rightarrow (1\$_1S)$ to P'.
2. For every $r : (A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n) \in P$ add $1_r : (\$_1, A_1, \ldots, A_n) \rightarrow (1_r\$_1, \gamma(x_1), \ldots, \gamma(x_n))$ to P'.
3. Add
   1. $2 : (\$_1) \rightarrow (2\$_2)$ and
   2. $2_\varepsilon : (\$_1) \rightarrow (2_\varepsilon\$_3)$ to P' .
4. For each $a \in T$ add
   1. $2_a : (X, \$_2, Z, a') \rightarrow (aX, 2_a\$_2, Y, Z)$ and
   2. $3_a : (X, \$_2, Z, a') \rightarrow (a, 3_a\$_2, Y, Y)$ to P'.
5. Add $3 : (\$_3, Y) \rightarrow (3, \$_3)$ to P'.
6. Add $4 : (\$_3) \rightarrow (4)$.

Summary

Result

Every RE language can be characterized by propagating SCG-P.

Future Research

Propagating SCG (or other grammars) generating/accepting parse trees.

Summary

Result

Every RE language can be characterized by propagating SCG-P.

Future Research

Propagating SCG (or other grammars) generating/accepting parse trees.

References

A. Meduna, J. Techet
Scattered Context Grammars and Their Applications.
WIT Press, 2009.

A. Ehrenfeucht, G. Rozenberg
An observation on scattered context grammars.
Information Processing Letters, pp. 84–85, 1979.

J. Dassow, G. Păun
Regulated Rewriting in Formal Languages Theory.
Akademie-Verlag, 1989.