Scattered Context Grammars

A. Meduna J. Techet

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

2008

Based on

2008 Masopust T., Techet J.: Leftmost Derivations of Propagating Scattered Context Grammars: A New Proof, *Discrete Mathematics and Theoretical Computer Science*, 10, 39–46

2007 Meduna A., Techet J.: Canonical Scattered Context Generators of Sentences with Their Parses, *Theoretical Computer Science*, 389, 73–81

2007 Meduna A., Techet J.: Maximal and Minimal Scattered Context Rewriting, *FCT 2007 Proceedings*, Budapest, 412–423

2007 Meduna A., Techet J.: Reduction of Scattered Context Generators of Sentences Preceded by Their Leftmost Parses, *Proceedings of DCFS 2007*, High Tatras, 178–185

2007 Techet J.: k-Limited Erasing Performed by Scattered Context Grammars, *WFM 2007*, Hradec nad Moravicí, 227–234

2005 Meduna A., Techet J.: Generation of Sentences with Their Parses: the Case of Propagating Scattered Context Grammars, *Acta Cybernetica*, 17, 11–20

Outline

Outline

Scattered Information and Its Grammatical Formalization

- while context-sensitive grammars are suitable for modeling immediate context...

$$AAAABCAAAA \qquad BC \rightarrow AA$$

...they fail to describe scattered context dependencies efficiently

$$ABAAAAAACA \qquad BA \rightarrow AA', \; A'A \rightarrow AA', \; A'C \rightarrow AA$$

- scattered context dependencies are common in real world:

He is interested in football, isn't he?

```
... int i; int j = 10; for (i = 0; i < j; i++) { ...
```

- scattered context grammars (introduced by S. Greibach and J. Hopcroft in 1969) are convenient for describing this kind of dependencies

Scattered Context Grammars

Scattered Context Grammar

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

V is a 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

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

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

Propagating Scattered Context Grammar

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

Derivation Step

Derivation Step

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

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

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

- $\mathrm{alph}(x)$ denotes the set of all symbols appearing in x

Leftmost Derivation Step

- each A_i satisfies $A_i \notin \mathrm{alph}(u_i)$

Generated Language

Generated Language

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

Language Families

- $\mathscr{L}(SC)$ – scattered context languages
- $\mathscr{L}(PSC)$ – propagating scattered context languages

Theorem

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

Theorem

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

Example

Example

Scattered context grammar

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

where

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

Example of derivation

$$S \Rightarrow ABC \Rightarrow aAbBcC \Rightarrow aaAbbBccC \Rightarrow aabbcc$$

Generated language

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

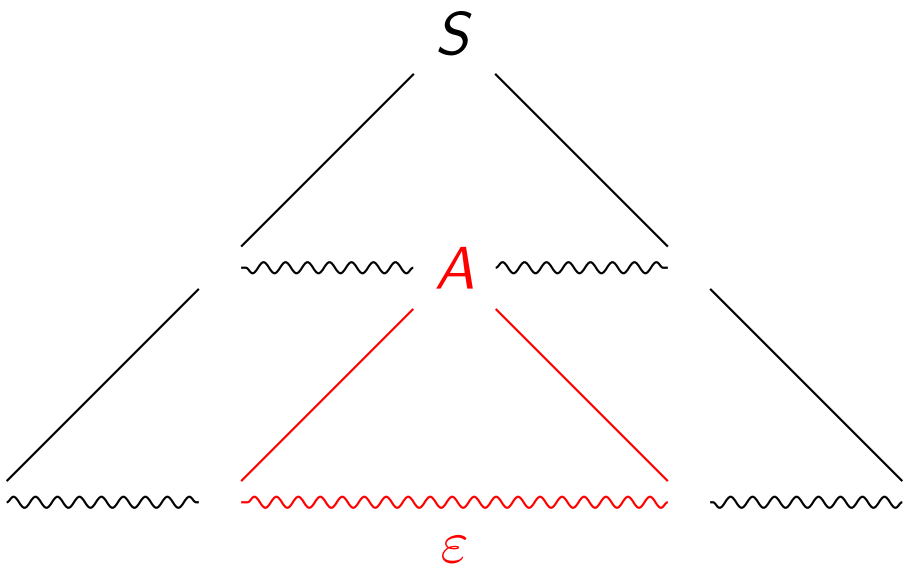
Outline

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;

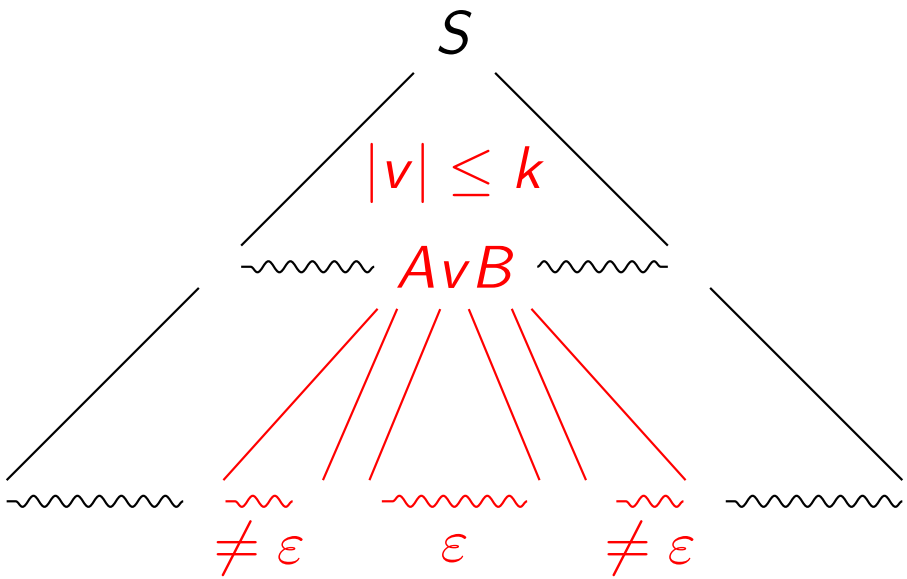
- if the symbol A is erased, we write $\grave{A}$,
- if the symbol A is not erased, we write $\acute{A}$



Nonterminals Erased in a Generalized k-Limited Way

For $y \in L(G)$, every sentential form x in $S \Rightarrow_G^* y$ satisfies:

1. every $x = uAvBw$, where $\acute{A}$, $\acute{B}$, $\grave{v}$, satisfies $|v| \leq k$,
2. every $x = uAw$, where $\acute{A}$, satisfies: if $\grave{u}$ or $\grave{w}$, then $|u| \leq k$ or $|w| \leq k$, respectively



Results

Nonterminals Erased in Generalized k-Limited Way by SC Grammars

A scattered context grammar G erases its nonterminals in a generalized k-limited way if $L(G) = L(G, \varepsilon, k)$, where

$$L(G, \varepsilon, k) = \{x \in T^* : S \Rightarrow_G^* x, \text{ and } G \text{ erases nonterminals in a generalized } k\text{-limited way in } S \Rightarrow_G^* x\}$$

Theorem

For each $k \geq 0$ and every scattered context grammar G, there is a propagating scattered context grammar $\bar{G}$ such that $L(G, \varepsilon, k) = L(\bar{G})$.

Corollary

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

Outline

Linear Scattered Context Grammars

Linear Scattered Context Grammar

- is a scattered context grammar $G = (V, T, P, S)$
- P is a finite set of productions of the following two forms:
  1. $(S) \rightarrow (x_1 A_1 \dots x_k A_k x_{k+1})$, where $A_i \in (V - T) - \{S\}$, $x_j \in T^*$,
  2. $(A_1, \dots, A_k) \rightarrow (z_1, \dots, z_k)$, where $A_i \in (V - T) - \{S\}$, and either
     - $z_i = x_i B_i y_i$, where $x_i, y_i \in T^*$, $B_i \in (V - T) - \{S\}$, or
     - $z_i \in T^*$

Linear Scattered Context Grammar of Degree n

- $(S) \rightarrow (x_1 A_1 \dots x_n A_n x_{n+1}) \in P$ satisfies $n \geq m$ for all $(S) \rightarrow (y_1 A_1 \dots y_m A_m y_{m+1}) \in P$

Right-Linear Scattered Context Grammars

Right-Linear Scattered Context Grammar

- is a linear scattered context grammar $G = (V, T, P, S)$
- P is a finite set of productions of the following two forms:
  1. $(S) \rightarrow (x_1 A_1 \dots x_k A_k)$, where $A_i \in (V - T) - \{S\}$, $x_i \in T^*$,
  2. $(A_1, \dots, A_k) \rightarrow (z_1, \dots, z_k)$, where $A_i \in (V - T) - \{S\}$, and either
     - $z_i = x_i B_i$, where $x_i \in T^*$, $B_i \in (V - T) - \{S\}$, or
     - $z_i \in T^*$

Language Families

- $\mathscr{L}(SC, LIN, n)$ – linear scattered context grammars of degree n
- $\mathscr{L}(SC, RLIN, n)$ – right-linear scattered context grammars of degree n

Results

Theorem

For each $n \geq 1$,

$$\begin{aligned} \mathscr{L}(SC, LIN, n) &\subset \mathscr{L}(SC, LIN, n+1), \\ \mathscr{L}(SC, RLIN, n) &\subset \mathscr{L}(SC, RLIN, n+1), \\ \mathscr{L}(SC, RLIN, n) &\subset \mathscr{L}(SC, LIN, n). \end{aligned}$$

- $\mathscr{L}(SC, LIN) = \bigcup_{n=1}^{\infty} \mathscr{L}(SC, LIN, n)$
- $\mathscr{L}(SC, RLIN) = \bigcup_{n=1}^{\infty} \mathscr{L}(SC, RLIN, n)$

Theorem

$$\mathscr{L}(CF) - \mathscr{L}(SC, LIN) \neq \emptyset, \ \mathscr{L}(CF) - \mathscr{L}(SC, RLIN) \neq \emptyset,$$

$$\mathscr{L}(SC, RLIN) \subset \mathscr{L}(SC, LIN) \subset \mathscr{L}(PSC).$$

Outline

n-Limited Derivations

- $|x|_A$ denotes the number of occurrences of symbols from A in x

n-Limited Derivation Step

If $(A_1, \ldots, A_k) \rightarrow (x_1, \ldots, x_k) \in P$,

$$u = u_1 A_1 u_2 \ldots u_k A_k u_{k+1},$$
$$v = u_1 x_1 u_2 \ldots u_k x_k u_{k+1},$$

and u satisfies

$$|u_1 A_1 \ldots u_k A_k|_{V-T} \leq n,$$

then $u \overset{n}{\underset{\text{lim}}{\Rightarrow}}_G v$

n-Limited Derivation

- derivation $x \overset{n}{\underset{\text{lim}}{\Rightarrow}}_G^* y$ in which every derivation step $u \overset{j}{\underset{\text{lim}}{\Rightarrow}}_G v$ satisfies $j \leq n$

Results

Language of Order n

- $L(G, \lim, n) = \{x \in T^* : S \overset{n}{\underset{\lim}{\Rightarrow}}_G^* x\}$

Language Families

- $\mathscr{L}(PSC, \lim, n)$
- $\mathscr{L}(PSC, \lim, \infty) = \bigcup_{i=1}^{\infty} \mathscr{L}(PSC, \lim, i)$

Theorem

$\mathscr{L}(CF) = \mathscr{L}(PSC, \lim, 1) \subset \ldots \subset \mathscr{L}(PSC, \lim, \infty) \subset \mathscr{L}(CS)$.

Outline

Leftmost Derivations

- much simplified proof of the result proved by V. Virkkunen in 1973

Propagating Scattered Context Grammar which Uses Leftmost Derivations

- propagating scattered context grammar $G = (V, T, P, S)$ whose language is defined as

$$L(G, \mathrm{lm}) = \{x \in T^* : S \ {}_{\mathrm{lm}}\Rightarrow_G^* x\}$$

Language Family

- $\mathscr{L}(PSC, \mathrm{lm})$

Theorem

$\mathscr{L}(PSC, \mathrm{lm}) = \mathscr{L}(CS)$.

Outline

Maximal and Minimal Derivation

- $\text{len}((A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n)) = |A_1 \ldots A_n| = n$

Maximal Derivation Step

If

1. $u \Rightarrow v\ [p]$,
2. there is no $r \in P$, $\text{len}(r) > \text{len}(p)$, such that $u \Rightarrow w\ [r]$,

where $p \in P$, then $u \ {}_{\max}\!\Rightarrow v\ [p]$

Minimal Derivation Step

If

1. $u \Rightarrow v\ [p]$,
2. there is no $r \in P$, $\text{len}(r) < \text{len}(p)$, such that $u \Rightarrow w\ [r]$,

where $p \in P$, then $u \ {}_{\min}\!\Rightarrow v\ [p]$

Results

Maximal and Minimal Languages

- $L(G, \text{max}) = \{x \in T^* : S \ {}_{\text{max}}\Rightarrow^* x\}$
- $L(G, \text{min}) = \{x \in T^* : S \ {}_{\text{min}}\Rightarrow^* x\}$

Language Families

- $\mathscr{L}(PSC, \text{max})$
- $\mathscr{L}(PSC, \text{min})$

Theorem

$\mathscr{L}(CS) = \mathscr{L}(PSC, \text{max})$.

Theorem

$\mathscr{L}(CS) = \mathscr{L}(PSC, \text{min})$.

Outline

Production Labels

Production Label

- for every grammar G, $lab(G)$ denotes the set of its production labels
- each $p \in lab(G)$ uniquely identifies one production:

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

Derivation Made by Productions

- if $x \Rightarrow y$ by $p : (A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n)$, we write

$$x \Rightarrow y \ [p]$$

- if $x \Rightarrow^* y$ by productions labeled with $p_1, \ldots, p_n$, we write

$$x \Rightarrow^* y \ [p_1 \ldots p_n]$$

Proper Generator of Its Sentences With Their Parses

Parse (Szilard Word, Control Word)

If

$$S \Rightarrow^* x\,[\rho],$$

where $x \in T^*$, $\rho \in lab(G)^*$, then x is a sentence generated according to parse ρ

Proper Generator of Its Sentences With Their Parses

- $G = (V, T, P, S)$, where $\mathrm{lab}(G) \subset T$, which satisfies

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

- leftmost generator uses leftmost derivations in a successful derivation

Results

- $G = (V, P, S, T)$ is a proper generator of its sentences with their parses
- weak identity π from V^* to $(V - lab(G))^*$:
  - $\pi(a) = a$ for each $a \in V - lab(G)$
  - $\pi(p) = \epsilon$ for each $p \in lab(G)$

Theorem

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

Theorem

For every recursively enumerable language L, there exists a propagating scattered context grammar $G = (V, T, P, S)$ such that G is a proper leftmost generator of its sentences with their parses, $|V - T| \leq 6$, and $L = \pi(L(G))$.

Outline

Applications in Linguistics

- there are scattered dependencies in natural languages

  He usually, but not always, goes to work early.

- the subject (he) depends on the predicator (goes):

  I usually, but not always, goes to work early.

  is grammatically incorrect

- the dependency can be easily captured by productions of scattered context grammars:

  $$(\text{He}, \text{goes}) \rightarrow (\text{I}, \text{go})$$

  transforms the original sentence to

  I usually, but not always, go to work early.

- these transformations are formalized and studied on simple sentences

Transformational Scattered Context Grammar

Transformational Scattered Context Grammar

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

V is the total vocabulary

$T \subset V$ is a set of terminals

P is a set of scattered context productions

$I \subset V$ is the input vocabulary

Transformation T Defined by G

- $T(G) = \{(x, y) : x \Rightarrow_G^* y, x \in I^*, y \in T^*\}$
- if $(x, y) \in T(G)$, then x is transformed to y by G
- x and y are called the input and the output sentence

Clauses with *neither* and *nor*

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

T is the set of all words, including all their inflectional forms

$I = \{\langle x \rangle : x \in T\}$

$V = T \cup I$

$$P = \{(\langle \text{neither} \rangle, \langle \text{nor} \rangle) \rightarrow (\text{both}, \text{and})\}$$
$$\cup \{(\langle x \rangle) \rightarrow (x) : x \in T - \{\text{neither, nor}\}\}$$

Example

$\langle \text{neither} \rangle \langle \text{thomas} \rangle \langle \text{nor} \rangle \langle \text{his} \rangle \langle \text{wife} \rangle \langle \text{went} \rangle \langle \text{to} \rangle \langle \text{the} \rangle \langle \text{party} \rangle$

$\Rightarrow_G$ both $\langle \text{thomas} \rangle$and $\langle \text{his} \rangle \langle \text{wife} \rangle \langle \text{went} \rangle \langle \text{to} \rangle \langle \text{the} \rangle \langle \text{party} \rangle$

$\Rightarrow_G$ both thomas and $\langle \text{his} \rangle \langle \text{wife} \rangle \langle \text{went} \rangle \langle \text{to} \rangle \langle \text{the} \rangle \langle \text{party} \rangle$

$\Rightarrow_G$ both thomas and his $\langle \text{wife} \rangle \langle \text{went} \rangle \langle \text{to} \rangle \langle \text{the} \rangle \langle \text{party} \rangle$

$\Rightarrow_G^5$ both thomas and his wife went to the party

Existential Clauses

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

T is the set of all words, including all their inflectional forms

$I = \{\langle x \rangle : x \in T\}$

$V = T \cup I \cup \{X\}$

$P = \{(\langle x \rangle, \langle \text{is} \rangle) \to (\text{there is } xX, \varepsilon), (\langle x \rangle, \langle \text{are} \rangle) \to (\text{there are } xX, \varepsilon),$
$(\langle x \rangle, \langle \text{was} \rangle) \to (\text{there was } xX, \varepsilon), (\langle x \rangle, \langle \text{were} \rangle) \to (\text{there were } xX, \varepsilon)$
$\cup \{(X, \langle x \rangle) \to (X, x) : x \in T\} \cup \{(X) \to (\varepsilon)\}$

Example

$\langle \text{a} \rangle \langle \text{nurse} \rangle \langle \text{was} \rangle \langle \text{present} \rangle \Rightarrow_G \text{there was a } X \langle \text{nurse} \rangle \langle \text{present} \rangle$
$\Rightarrow_G \text{there was a } X \text{ nurse } \langle \text{present} \rangle \Rightarrow_G \text{there was a } X \text{ nurse present}$
$\Rightarrow_G \text{there was a nurse present}$

Outline

Further Investigation

Possibilities of Practical Applications

- applications in compilers
- applications in natural language processing

Theoretical Properties

- our research rises several open problems:
  - Does generalized k-limited erasing cover all possible kinds of erasing which can be performed by propagating scattered context grammars?
  - For given n, is it possible to construct a propagating scattered context grammar which rewrites the first n nonterminals without any additional restrictions?
- other papers dealing with this topic introduced several open problems as well
- the problem of whether $\mathscr{L}(PSC) = \mathscr{L}(CS)$ remains still open

Thank you