Context-Conditional Grammars: An Overview

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

A bit of history

- 1970, Van der Walt introduces Random Context Grammars
    - Context-free grammars where two finite sets of symbols (permitting and forbidding context) are associated with each production.

- This is about variants of generalized RC grammars (strings are permitted in permitting and forbidding contexts).

Context-Conditional Grammars: Definition

Definition

A context-conditional grammar (cc-grammar) is a quadruple

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

where

- N is a nonterminal alphabet,
- T is a terminal alphabet such that $N \cap T = \emptyset$, $V = N \cup T$,
- $S \in N$ is the start symbol, and
- P is a finite set of productions of the form

$$(X \to \alpha, Per, For),$$

$X \in N$, $\alpha \in V^*$, and $Per, For \subseteq V^+$ are finite sets.

Context-Conditional Grammars: Example

Example

Consider a grammar

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

with P consisting of the following productions:

1. $(S \to AB, \emptyset, \emptyset)$,
2. $(A \to c, \{B\}, \{ABB\})$,
3. $(B \to d, \{AA, AB\}, \{A, B, S\})$.

Then, G is a cc-grammar.

Context-Conditional Grammars: Definitions

Permitting and Forbidding contexts.

Definition (Derivation Step)

For $u, v \in (N \cup T)^*$, and $(X \to \alpha, Per, For) \in P$,

$$uXv \Rightarrow u\alpha v,$$

if

$$Per \subseteq sub(uXv)^1 \text{ and } For \cap sub(uXv) = \emptyset\,.$$

Definition (Language)

$$L(G) = \{w \in T^* : S \Rightarrow^* w\} \text{ and}$$

$$\mathbf{CCG} = \{L(G) : G \text{ is a cc-grammar}\}$$

[1] sub(x)={u : u is a subword of x}

Context-Conditional Grammars: Example

Example

Consider a cc-grammar

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

with P consisting of the following productions:

1. $(S \rightarrow AB, \emptyset, \emptyset)$,
2. $(A \rightarrow c, \{B\}, \{ABB\})$,
3. $(B \rightarrow d, \{AA, AB\}, \{A, B, S\})$.

Then,

$$\begin{array}{lcll} AAB & \Rightarrow & AcB & [(A \rightarrow c, \{B\}, \{ABB\})] \\ AAB & \not\Rightarrow & AAd & [(B \rightarrow d, \{AA, AB\}, \{A, B, S\})] \end{array}$$

Context-Conditional Grammars: Definitions

Definition (Conditional Production)

$(X \to \alpha, Per, For) \in P$ is said to be conditional if

$$Per \cup For \neq \emptyset\,.$$

Definition (Degree)

G has degree (i, j) if for all productions

$$(X \to \alpha, Per, For) \in P\,,$$

$$|x| \leq i, \quad x \in Per$$

and

$$|y| \leq j, \quad y \in For\,.$$

Context-Conditional Grammars: Example

Example

Consider the previous cc-grammar

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

with P consisting of the following productions:

1. $(S \rightarrow AB, \emptyset, \emptyset)$ is not conditional,
2. $(A \rightarrow c, \{B\}, \{ABB\})$ is conditional,
3. $(B \rightarrow d, \{AA, AB\}, \{A, B, S\})$ is conditional.

Then, G has degree $(2, 3)$.

Context-Conditional Grammars: Results

Theorem
CCG = **RE**

Proof.
No surprise, **RC** = **RE** (hard, see Dassow and Paun). □

Context-Conditional Grammars: Results

Theorem
CCG = **RE**

Proof.
No surprise, **RC** = **RE** (hard, see Dassow and Paun). □

Theorem
*Let $L \in$ **RE**, then L is generated by a cc-grammar of degree $(2, 1)$ with no more than 6 conditional productions and 7 nonterminals.*

Context-Conditional Grammars: Results

Theorem
Context-conditional grammars with regular productions have the same generative power as regular grammars.

Theorem
Context-conditional grammars with linear productions have the same generative power as linear grammars.

Simple Context-Conditional Grammars: Definition

Definition

A simple context-conditional grammar (scc-grammar) is a cc-grammar

$$G = (N, T, P, S)\,,$$

where

$$(X \rightarrow \alpha, Per, For) \in P$$

implies that

$$\emptyset \in \{Per, For\}\,.$$

Simple Context-Conditional Grammars: Results

Theorem

*Let $L \in$ **RE**, then L is generated by a scc-grammar of degree $(2, 1)$ with no more than 7 conditional productions and 8 nonterminals.*

Proof.

Based on the Geffert normal form. □

Simple Context-Conditional Grammars: Results

Proof Prerequisite.

Every *RE* language is generated by a grammar

$$G_1 = (\{S, A, B, C\}, T, P \cup \{ABC \rightarrow \varepsilon\}, S)$$

where P contains context-free productions of the form

$$S \rightarrow uSa,\ S \rightarrow uSv,\ S \rightarrow uv,$$

where $u \in \{A, AB\}^*$, $v \in \{BC, C\}^*$, $a \in T$.
In addition, $w \in \mathscr{L}(G_1)$ iff

$$S \Rightarrow_P^* w_1 ABC w_2 w \Rightarrow_{\{ABC \rightarrow \varepsilon\}}^* w,$$

where $w_1 \in \{A, AB\}^*$, $w_2 \in \{BC, C\}^*$, and $w \in T^*$. □

Proof Construction.

$G = (\{S, A, B, C, A', B', C', B''\}, T, P_1 \cup P_2, S)$, where

$$P_1 = \{(X \to \alpha, \emptyset, \emptyset) : X \to \alpha \in P\},$$

and P_2 contains:

1. $(A \to A', \emptyset, \{A', B''\})$
2. $(B \to B', \emptyset, \{B', B''\})$
3. $(C \to C', \emptyset, \{C', B''\})$
4. $(B' \to B'', \{A'B', B'C'\}, \emptyset)$
5. $(A' \to \varepsilon, \{B''\}, \emptyset)$
6. $(C' \to \varepsilon, \{B''\}, \emptyset)$
7. $(B'' \to \varepsilon, \emptyset, \{A', C'\})$

Derivation ($ABC \to \varepsilon$)

$$S \quad \Rightarrow^*_{P_1} w_1 ABC w_2 w$$

□

Proof Construction.

$G = (\{S, A, B, C, A', B', C', B''\}, T, P_1 \cup P_2, S)$, where

$$P_1 = \{(X \rightarrow \alpha, \emptyset, \emptyset) : X \rightarrow \alpha \in P\},$$

and P_2 contains:

1. $(A \rightarrow A', \emptyset, \{A', B''\})$
2. $(B \rightarrow B', \emptyset, \{B', B''\})$
3. $(C \rightarrow C', \emptyset, \{C', B''\})$
4. $(B' \rightarrow B'', \{A'B', B'C'\}, \emptyset)$
5. $(A' \rightarrow \varepsilon, \{B''\}, \emptyset)$
6. $(C' \rightarrow \varepsilon, \{B''\}, \emptyset)$
7. $(B'' \rightarrow \varepsilon, \emptyset, \{A', C'\})$

Derivation ($ABC \rightarrow \varepsilon$)

$$\begin{aligned} S &\Rightarrow^*_{P_1} w_1 ABC w_2 w \\ &\Rightarrow w_1 A' B C w_2 w \end{aligned}$$

□

Proof Construction.

$G = (\{S, A, B, C, A', B', C', B''\}, T, P_1 \cup P_2, S)$, where

$$P_1 = \{(X \rightarrow \alpha, \emptyset, \emptyset) : X \rightarrow \alpha \in P\},$$

and P_2 contains:

1. $(A \rightarrow A', \emptyset, \{A', B''\})$
2. $(B \rightarrow B', \emptyset, \{B', B''\})$
3. $(C \rightarrow C', \emptyset, \{C', B''\})$
4. $(B' \rightarrow B'', \{A'B', B'C'\}, \emptyset)$
5. $(A' \rightarrow \varepsilon, \{B''\}, \emptyset)$
6. $(C' \rightarrow \varepsilon, \{B''\}, \emptyset)$
7. $(B'' \rightarrow \varepsilon, \emptyset, \{A', C'\})$

Derivation ($ABC \rightarrow \varepsilon$)

$$\begin{array}{lll} S & \Rightarrow^*_{P_1} & w_1 ABC w_2 w \\ & \Rightarrow & w_1 A'BC w_2 w \\ & \Rightarrow & w_1 A'B'C w_2 w \end{array}$$

□

Proof Construction.

$G = (\{S, A, B, C, A', B', C', B''\}, T, P_1 \cup P_2, S)$, where

$$P_1 = \{(X \rightarrow \alpha, \emptyset, \emptyset) : X \rightarrow \alpha \in P\},$$

and P_2 contains:

1. $(A \rightarrow A', \emptyset, \{A', B''\})$
2. $(B \rightarrow B', \emptyset, \{B', B''\})$
3. $(C \rightarrow C', \emptyset, \{C', B''\})$
4. $(B' \rightarrow B'', \{A'B', B'C'\}, \emptyset)$
5. $(A' \rightarrow \varepsilon, \{B''\}, \emptyset)$
6. $(C' \rightarrow \varepsilon, \{B''\}, \emptyset)$
7. $(B'' \rightarrow \varepsilon, \emptyset, \{A', C'\})$

Derivation ($ABC \rightarrow \varepsilon$)

$$\begin{aligned} S &\Rightarrow^*_{P_1} w_1 ABC w_2 w \\ &\Rightarrow w_1 A'BC w_2 w \\ &\Rightarrow w_1 A'B'C w_2 w \\ &\Rightarrow w_1 A'B'C' w_2 w \end{aligned}$$

□

Proof Construction.
$G = (\{S, A, B, C, A', B', C', B''\}, T, P_1 \cup P_2, S)$, where

$$P_1 = \{(X \rightarrow \alpha, \emptyset, \emptyset) : X \rightarrow \alpha \in P\},$$

and P_2 contains:

1. $(A \rightarrow A', \emptyset, \{A', B''\})$
2. $(B \rightarrow B', \emptyset, \{B', B''\})$
3. $(C \rightarrow C', \emptyset, \{C', B''\})$
4. $(B' \rightarrow B'', \{A'B', B'C'\}, \emptyset)$
5. $(A' \rightarrow \varepsilon, \{B''\}, \emptyset)$
6. $(C' \rightarrow \varepsilon, \{B''\}, \emptyset)$
7. $(B'' \rightarrow \varepsilon, \emptyset, \{A', C'\})$

Derivation ($ABC \rightarrow \varepsilon$)

$$\begin{aligned} S &\Rightarrow^*_{P_1} w_1 ABC w_2 w \\ &\Rightarrow w_1 A' BC w_2 w \\ &\Rightarrow w_1 A'B'C w_2 w \\ &\Rightarrow w_1 A'B'C' w_2 w \\ &\Rightarrow w_1 A'B''C' w_2 w \end{aligned}$$

□

Proof Construction.
$G = (\{S, A, B, C, A', B', C', B''\}, T, P_1 \cup P_2, S)$, where

$$P_1 = \{(X \rightarrow \alpha, \emptyset, \emptyset) : X \rightarrow \alpha \in P\},$$

and P_2 contains:

1. $(A \rightarrow A', \emptyset, \{A', B''\})$
2. $(B \rightarrow B', \emptyset, \{B', B''\})$
3. $(C \rightarrow C', \emptyset, \{C', B''\})$
4. $(B' \rightarrow B'', \{A'B', B'C'\}, \emptyset)$
5. $(A' \rightarrow \varepsilon, \{B''\}, \emptyset)$
6. $(C' \rightarrow \varepsilon, \{B''\}, \emptyset)$
7. $(B'' \rightarrow \varepsilon, \emptyset, \{A', C'\})$

Derivation ($ABC \rightarrow \varepsilon$)

$$\begin{aligned} S &\Rightarrow^*_{P_1} w_1 ABC w_2 w \\ &\Rightarrow w_1 A'BC w_2 w \\ &\Rightarrow w_1 A'B'C w_2 w \\ &\Rightarrow w_1 A'B'C' w_2 w \\ &\Rightarrow w_1 A'B''C' w_2 w \\ &\Rightarrow w_1 B''C' w_2 w \end{aligned}$$

□

Proof Construction.

$G = (\{S, A, B, C, A', B', C', B''\}, T, P_1 \cup P_2, S)$, where

$$P_1 = \{(X \rightarrow \alpha, \emptyset, \emptyset) : X \rightarrow \alpha \in P\},$$

and P_2 contains:

1. $(A \rightarrow A', \emptyset, \{A', B''\})$
2. $(B \rightarrow B', \emptyset, \{B', B''\})$
3. $(C \rightarrow C', \emptyset, \{C', B''\})$
4. $(B' \rightarrow B'', \{A'B', B'C'\}, \emptyset)$
5. $(A' \rightarrow \varepsilon, \{B''\}, \emptyset)$
6. $(C' \rightarrow \varepsilon, \{B''\}, \emptyset)$
7. $(B'' \rightarrow \varepsilon, \emptyset, \{A', C'\})$

Derivation ($ABC \rightarrow \varepsilon$)

$$\begin{array}{lll} S & \Rightarrow^*_{P_1} & w_1ABCw_2w \\ & \Rightarrow & w_1A'BCw_2w \\ & \Rightarrow & w_1A'B'Cw_2w \\ & \Rightarrow & w_1A'B'C'w_2w \\ & \Rightarrow & w_1A'B''C'w_2w \\ & \Rightarrow & w_1B''C'w_2w \\ & \Rightarrow & w_1B''w_2w \end{array}$$

□

Proof Construction.

$G = (\{S, A, B, C, A', B', C', B''\}, T, P_1 \cup P_2, S)$, where

$$P_1 = \{(X \rightarrow \alpha, \emptyset, \emptyset) : X \rightarrow \alpha \in P\},$$

and P_2 contains:

1. $(A \rightarrow A', \emptyset, \{A', B''\})$
2. $(B \rightarrow B', \emptyset, \{B', B''\})$
3. $(C \rightarrow C', \emptyset, \{C', B''\})$
4. $(B' \rightarrow B'', \{A'B', B'C'\}, \emptyset)$
5. $(A' \rightarrow \varepsilon, \{B''\}, \emptyset)$
6. $(C' \rightarrow \varepsilon, \{B''\}, \emptyset)$
7. $(B'' \rightarrow \varepsilon, \emptyset, \{A', C'\})$

Derivation ($ABC \rightarrow \varepsilon$)

$$\begin{aligned} S &\Rightarrow^*_{P_1} w_1 ABC w_2 w \\ &\Rightarrow w_1 A'BC w_2 w \\ &\Rightarrow w_1 A'B'C w_2 w \\ &\Rightarrow w_1 A'B'C' w_2 w \\ &\Rightarrow w_1 A'B''C' w_2 w \\ &\Rightarrow w_1 B''C' w_2 w \\ &\Rightarrow w_1 B'' w_2 w \\ &\Rightarrow w_1 w_2 w \end{aligned}$$

□

Generalized Forbidding Grammars: Definition

Definition

A generalized forbidding grammar (gf-grammar) is a cc-grammar

$$G = (N, T, P, S)\,,$$

where

$$(X \to \alpha, Per, For) \in P$$

implies that

$$Per = \emptyset\,.$$

Generalized Forbidding Grammars: Definition

Definition

A generalized forbidding grammar (gf-grammar) is a cc-grammar

$$G = (N, T, P, S)\,,$$

where

$$(X \rightarrow \alpha, Per, For) \in P$$

implies that

$$Per = \emptyset\,.$$

$(X \rightarrow \alpha, \emptyset, For)$ is simplified to $(X \rightarrow \alpha, For)$.

Generalized Forbidding Grammars: Definition

Definition

A generalized forbidding grammar (gf-grammar) is a cc-grammar

$$G = (N, T, P, S)\,,$$

where

$$(X \to \alpha, Per, For) \in P$$

implies that

$$Per = \emptyset\,.$$

$(X \to \alpha, \emptyset, For)$ is simplified to $(X \to \alpha, For)$.

Definition (Degree)

G has degree i if G has degree (k, i) as a cc-grammar, for some $k \geq 0$.

Generalized Forbidding Grammars: Example

Example

Consider a gf-grammar (forbidding gr.=no strings, only symbols)

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

with P consisting of the following productions:

1. $(A \rightarrow BB, \{C\})$
2. $(B \rightarrow C, \{A\})$
3. $(C \rightarrow A, \{a, B\})$
4. $(C \rightarrow a, \{A, B\})$

Then, G has degree 1 and $AA \Rightarrow BBA \Rightarrow BBBB \Rightarrow^4 CCCC \Rightarrow^4 aaaa$.
Thus,

$$L(G) = \{a^{2^n} : n \geq 1\}.$$

Generalized Forbidding Grammars: Results

Theorem (Meduna, 1990)

$\mathbf{GFG} = \mathbf{RE}$

Theorem (Bordihn and Fernau, 1995)

$\mathbf{F} \subset \mathbf{REC}$ *(forbidding grammars=no strings, only symbols)*

Generalized Forbidding Grammars: Results

Theorem (Meduna, 1990)

$\mathbf{GFG} = \mathbf{RE}$

Theorem (Bordihn and Fernau, 1995)

$\mathbf{F} \subset \mathbf{REC}$ *(forbidding grammars=no strings, only symbols)*

Theorem

Let $L \in \mathbf{RE}$, then L is generated by a gf-grammar of degree 2 with no more than 8 conditional productions and 10 nonterminals.

Generalized Forbidding Grammars: Results

Theorem (Meduna, 1990)

GFG $=$ **RE**

Theorem (Bordihn and Fernau, 1995)

F $\subset$ **REC** *(forbidding grammars=no strings, only symbols)*

Theorem

Let $L \in$ **RE***, then* L *is generated by a gf-grammar of degree* 2 *with no more than 8 conditional productions and 10 nonterminals.*

Theorem

Let $L \in$ **RE***, then* L *is generated by a gf-grammar of degree* 2 *with no more than 9 conditional productions and 8 nonterminals.*

Generalized Permitting Grammars: Definition

Definition

A generalized permitting grammar (gp-grammar) is a cc-grammar

$$G = (N, T, P, S)\,,$$

where

$$(X \rightarrow \alpha, Per, For) \in P$$

implies that

$$For = \emptyset\,.$$

- As far as I know, nobody has studied descriptional complexity;
- gp-grammars vs. type-0 grammars.

Semi-Conditional Grammars: Definition

Definition

A Semi-Conditional Grammars (sc-grammar) is a cc-grammar

$$G = (N, T, P, S)\,,$$

where

$$(X \to \alpha, Per, For) \in P$$

implies that

$$|Per|, |For| \leq 1\,.^{2}$$

[2] Each context contains no more than one nonempty string.

Semi-Conditional Grammars: Results

Theorem

Let $L \in \mathbf{RE}$, then L is generated by a sc-grammar of degree $(2, 1)$ with no more than 7 conditional productions and 8 nonterminals.

Semi-Conditional Grammars: Results

Theorem

Let $L \in \mathbf{RE}$, then L is generated by a sc-grammar of degree $(2, 1)$ with no more than 7 conditional productions and 8 nonterminals.

Theorem (Mayer, 1972)

Let $L \in \mathbf{RE}$, then L is generated by a sc-grammar of degree $(1, 1)$.

Simple Semi-Conditional Grammars: Definition

Definition
A simple semi-conditional grammar (ssc-grammar) is a cc-grammar

$$G = (N, T, P, S)\,,$$

where

$$(X \rightarrow \alpha, Per, For) \in P$$

implies that

$$|Per| + |For| \leq 1\,.$$ [3]

[3] In each production, there is no more than one nonempty string in the union of its contexts.

Simple Semi-Conditional Grammars: Definition

Definition

A simple semi-conditional grammar (ssc-grammar) is a cc-grammar

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

where

$$(X \rightarrow \alpha, Per, For) \in P$$

implies that

$$|Per| + |For| \leq 1 .$$[3]

$(X \rightarrow \alpha, \{p\}, \{f\})$ is simplified to $(X \rightarrow \alpha, p, f)$, and $\emptyset$ to 0.

[3] In each production, there is no more than one nonempty string in the union of its contexts.

Simple Semi-Conditional Grammars: Example

Example

Consider a ssc-grammar

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

with P consisting of the following productions:

1. $(S \rightarrow AB, 0, 0)$,
2. $(A \rightarrow c, 0, B)$,
3. $(B \rightarrow d, AB, 0)$.

Then, G has degree

$$(2, 1)\,,$$

and

$$\begin{array}{lcll} AB & \Rightarrow & Ad & [(B \rightarrow d, AB, 0)] \\ AB & \not\Rightarrow & cB & [(A \rightarrow c, 0, B)] \end{array}$$

Simple Semi-Conditional Grammars: Results

Theorem

Let $L \in \mathbf{RE}$, then L is generated by a ssc-grammar of degree $(2, 1)$ with no more than 9 conditional productions and 10 nonterminals.

Theorem (Masopust and Meduna)

Let $L \in \mathbf{RE}$, then L is generated by a ssc-grammar of degree $(1, 1)$.[4]

[4] This was an open problem formulated in the book by Meduna and Švec, Grammars with Context Conditions and Their Applications, John Wiley & Sons, New York, 2005.

Summary

Every recursively enumerable language is generated by a

1. cc-grammar of degree $(2, 1)$ with six conditional productions and seven nonterminals;
2. scc-grammar of degree $(2, 1)$ with seven conditional productions and eight nonterminals;
3. gf-grammar of degree two with eight conditional productions and ten nonterminals;
4. gf-grammar of degree two with nine conditional productions and eight nonterminals;
5. sc-grammar of degree $(2, 1)$ with seven conditional productions and eight nonterminals; and
6. ssc-grammar of degree $(2, 1)$ with nine conditional productions and ten nonterminals.

Bibliography

T. Masopust.
Regulated Formal Models and Their Reductions.
PhD thesis, Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic, 2007.