Weak leftmost derivations in cooperative distributed grammar systems

Alexander Meduna, Filip Goldefus

MEMICS 2009

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

Contents of presentation

- Motivation
- Nondeterminism of a derivation step for grammars
  - Selection of a nonterminal to rewrite
    - Leftmost derivation
  - Selection of a production using in a derivation step
    - Ordered grammars
  - Generative power
- Grammar systems
  - Definition
  - Nondeterminism for grammar systems
  - Modifications of grammar systems
- Conclusion
- Bibliography

Motivation

- SW simulations – connections with concrete data structures
  - Queue, stack, list, …
- Derivation steps
  - Grammars and grammar systems are nondeterministic
- Simulations of **grammar systems**
  - Limiting nondeterminism → increase of generative power
- Comparing with **FA** and **TM** – accepting models
  - Limiting nondeterminism → same generative power
- Comparing with **PDA** and **DPDA**
  - Limiting nondeterminism → decrease of gen. power

Grammars

- **Grammar** - informally
  - Generating model from theory of formal languages
  - String of symbols → string of terminal symbols
    - Rewriting of nonterminals – using derivation steps
    - Language generation – set of terminal strings
- Formal **definition**
  - $G = (N, T, S, P)$
    - N is a finite set of nonterminal symbols (i.e. A, B, C, ...)
    - T is a finite set of nonterminal symbols (i.e. a, b, c, ...)
    - S is the starting nonterminal of the grammar G
    - P is a finite **set** of productions

Grammars – productions

- Types of grammars – Chomsky hierarchy
  - **Regular grammars** – $N \rightarrow TN$ or $N \rightarrow T$
    - A → aB, B → b, …
  - **Context-free grammars -** $N \rightarrow (T \cup N)^*$
    - A → ε, B → C, A → abcDEf
  - **Context sensitive grammars** – $(T \cup N)^* N (T \cup N)^* \rightarrow (T \cup N)^*$
    - $\alpha \rightarrow \beta, |\alpha| \leq |\beta|$
    - A →bA, C → CD, AbC → G, AB → DEF, …
  - **Unrestricted grammars** – $(T \cup N)^* N (T \cup N)^* \rightarrow (T \cup N)^*$
    - ABC → ε, F →gH, AbC → G, AB → DEF, …

Grammars – derivation step

- **Derivation step** of a grammar $G = (N, T, S, P)$
  - Application of a production on a string of symbols
    - Generate a string of terminal symbols
  - Denoted by $\Rightarrow$
  - **Definition**
    - $u\alpha v \Rightarrow u\beta v$, iff $\alpha \rightarrow \beta \in P$
  - Example
    - $AbCDef \Rightarrow AbxZyDef$, $C \rightarrow xZy \in P$
- **Language of a grammar** $G = (N, T, S, P)$
  - A set of strings from terminals derivated from the starting nonterminal S
  - $L(G) = \{w \in T^* \mid S \Rightarrow^* w\}$

Grammars – example

- Let $G = (N, T, S, P)$ be a grammar
  - $N = \{A, B, C, S\}$, $T = \{a, b, c\}$, $P = \{$
    $S \longrightarrow ABC,$
    $A \longrightarrow a \mid AA,$
    $B \longrightarrow b \mid BB,$
    $C \longrightarrow c\}$

$$S \Rightarrow^{*} AaAbbBC \Rightarrow ?$$

  - Interrelations between nondeterminisms
    - Selection of a nonterminal to rewrite
    - Selection of a production to use in derivation

Nondeterminism

- **Regular grammars**
  - A sentential form contains one nonterminal
    - P = {..., A → bC, A → b, ...}
    - abcdbA ⇒?
- **Context-free grammars**
  - A sentential form contains any number of nonterminals
    - P = {..., A → BCc, A → a, ...}
    - AaBbCaBA ⇒?
- **Context grammars** and **unrestricted grammars**
  - A sentential form contains any number of nonterminals
    - Both types of nondeterminism

Restrictions – Leftmost derivation

- **Regular grammars**
  - Trivial
- **Context-free grammars**
  - Leftmost nonterminal
    - Production for the nonterminal
  - Equivalence of leftmost derivation
- **Context** and **unrestricted grammars**
  - Example: *G* = ({S, A, B, C}, {b, c}, *S, P)*
  - *P* = {S → AAB, AB → BC, B → b, C → c}

S⇒ AAB⇒ABC⇒BCC⇒bCC⇒bcC⇒bcc

One of possible derivations.

Restriction – ordered grammars

- **Motivation**
  - Implementation: productions stored in lists
    - Ordering on productions.
- **Definition of a ordered grammar**
  - $H = (G, O)$
    - G is a grammar
    - $O \subseteq P \times P$, partial ordering on productions denoted by <, transitive closure
    - **Derivation step**
      - $u \Rightarrow v$, using a production $\alpha \rightarrow \beta$, iff there is no production $\gamma \rightarrow \delta$, such as $\gamma \rightarrow \delta < \alpha \rightarrow \beta$ and $u \Rightarrow w$ using production $\gamma \rightarrow \delta$

Ordered grammars

- Let G = ({A, B, C, D, S}, {a, b, c, d}, S, P) be a grammar
- Productions
  - S → ABCD, B → b < D → d < A → a < C → c
  - B → BB < D → DD

$$S \Rightarrow ABCD \Rightarrow AbCD \Rightarrow AbCd \Rightarrow abCd \Rightarrow abcd$$

- **Generative power** – classes of languages
  - **Context-free grammars** – *CF*
  - **Context grammars** - *CS*
  - **Ordered grammars with context-free prod.** – *OCF*
  - **Forbidding grammars** – *FOR*

$$CF \subset FOR \subset CS$$
$$FOR = OCF$$

Grammar systems

- **Motivation**
  - Sequential jobs of multiple processor
  - Hypothetical model
- Collection of grammars
- One shared sentential form
- Control of derivation is passed by cooperative protocol
- **Definition**
  - Let $\boldsymbol{G_1} = (N, T, S, P_1), \ldots, \boldsymbol{G_n} = (N, T, S, P_n)$ be a collection of n **context-free grammars**

  - Then $\boldsymbol{\Gamma} = (N, T, S, P_1, \ldots, P_n)$ is a **cooperative distributed (CD) grammar system of degree n**

Grammar systems

- **Cooperation protocols** - $k \in \mathbb{N}$
  - **= *k***, a grammar performs exactly *k* steps
  - **≤ *k***, a grammar performs at most *k* steps
  - **≥ *k***, a grammar performs at least *k* steps
  - *, a grammar performs arbitrary number of derivation steps
  - ***t -*** *erminating*, a grammar performs as much as possible derivation steps
  - Combinations using logical operators
    - Interval - ***≥ k and ≤ l***
    - Limited terminating mode - ***= k and t***

Grammar systems - example

- Let $\boldsymbol{\Gamma}$ = ({A, B, S}, {a, b}, S, P_1, P_2) be a grammar system working in cooperating mode =2.
  - $P_1 = \{A \rightarrow aA' \mid a,\ B \rightarrow bB' \mid b\}$
  - $P_2 = \{A' \rightarrow A \mid a,\ B' \rightarrow B,\ S \rightarrow S,\ S \rightarrow AB\}$

$$\underline{S} \Rightarrow_2 \underline{S} \Rightarrow_2 \underline{A}B \Rightarrow_1 aA'\underline{B} \Rightarrow_1 a\underline{A'}bB' \Rightarrow_2 aab\underline{B'} \Rightarrow_2 aabb$$

$$L(\boldsymbol{\Gamma}) = \{a^n b^n \mid n \geq 1\}$$

Grammar systems – generative power

- **Notation**
  - $CD_n(f)$ – **cooperating distributed grammar systems working of degree n** working in cooperating mode ***f***
  - MAT – **matrix grammars**
  - $ET0L$ – **ET0L grammars**

$$CD_\infty(f) = CF,\ f \in \{=1,\ *,\ \geq 1\} \cup \{\leq k \mid k \geq 1\}$$

$$CF = CD_1(f) \subset CD_2(f) \subseteq CD_r(f) \subseteq CD_\infty(f) \subseteq MAT,$$
$$f \in \{\geq k,\ \ =k \mid k \geq 2\},\ r \geq 3,$$

$$CF = CD_1\ (t) = CD_2(t) \subset CD_3(t) = CD_\infty(t) = ET0L$$

Graph controlled grammar systems

- **Motivation**
  - Limiting the nondeterminism of passing the control between grammars
  - Passing the control is random, same component can be chosen again
- **Definition**
  - $H = (\boldsymbol{\Gamma}, E)$ is **graph controlled cooperative distributed grammar system of degree n,** if
    - $\boldsymbol{\Gamma}$ is **cooperative distributed grammar system of degree n**
    - $E \subseteq K \times K$, $K = \{1, ..., n\}$ is a set of ordered pairs
      - If $(s, t) \in E$, then derivation is passed to component t after performing derivation steps by component s in chosen mode
- Graph controlling leads to increase of generative power

Grammar systems and leftmost derivations

- **Motivation**
  - Limiting nondeterminism of grammar systems
- **Two types of leftmost derivations** for grammar systems
  - **Weak leftmost derivation**
    - Leftmost **possible** nonterminal in sentential form is rewritten
    - Leads to increase of generative power
  - **Strong leftmost derivation**
    - Leftmost nonterminal in sentential form is rewritten
    - Generative power remains unchanged

Kuroda normal form

- Algorithm:
  - Input: **Arbitrary grammar *G***
  - Output: **Grammar *H* in Kuroda normal form**, such that $L(G) = L(H)$
- **Kuroda normal form** uses **four types of productions**

$$
\begin{aligned}
AB &\rightarrow CD \\
A &\rightarrow BC \\
A &\rightarrow a \\
A &\rightarrow \varepsilon
\end{aligned}
$$

- Used in proofs

Grammar systems and leftmost derivation - proof

- CD grammar system working in cooperating mode **=2**
- **Idea of proof**
  - Simulation of a unrestricted grammar by CD grammar system with leftmost derivation working in cooperating mode =2
  - Kuroda normal form types of productions
    - Productions of type $A \rightarrow BC$, $A \rightarrow b$, $A \rightarrow \varepsilon$ are in one component
    - For every production of type $AB \rightarrow CD$ there is unique component
- Generalisation of result – CD grammar system working in cooperating mode **=k** have power of unrestricted grammars

Ordered grammar systems

- **Motivation**
  - Limiting nondeterminism
  - Using ordered grammars as components
- **Idea of proof**
  - Simulation of **programmed grammars** with appearance checking by ordered CD grammar systems
  - Cooperating mode =1 is considered
  - For every production of programmed grammar there is one ordered grammar in ordered CD grammar system
  - **Programmed grammars** with appearance checking with erasing productions are as powerful as unrestricted grammars

Open areas

- Combination of limitations
  - Grammars
    - Combination of leftmost derivation and ordering on productions
  - Grammar systems
    - Combination of leftmost derivation and ordering on productions with graph controlled cooperation of grammars

Bibliography

- Goldefus Filip: Cooperating Distributed Grammar Systems and Graph Controlled Grammar Systems with Infinite Number of Components, In: Proceedings of the 15th Conference and Competition STUDENT EEICT 2009 Volume 4, Brno, CZ, UIFS FIT VUT, 2009, s. 400-404, ISBN 978-80-214-3870-5.
- Goldefus Filip: Left-Forbidding Cooperating Distributed Grammar Systems, In: Proceedings of the 14th Conference and Competition STUDENT EEICT 2008 Volume 4, Brno, CZ, FIT VUT, 2008, s. 403-407, ISBN 978-80-214-3617-6.
- Grzegorz Rozenberg, Arto Salomaa. Handbook of formal languages, vol. 2: linear modeling: background and application, Springer-Verlag New York, Inc, 1997. ISBN:3-540-60648-3.