On Nondeterminism in Programmed Grammars

Lukáš Vrábel and Petr Zemek

Brno University of Technology, Faculty of Information Technology
Božetěchova 2, 612 00 Brno, CZ
http://www.fit.vutbr.cz/∼{ivrabel, izemek}

BRNO UNIVERSITY OF TECHNOLOGY

FIT

FACULTY OF INFORMATION TECHNOLOGY

Research Group of Formal Models Seminar, 2011-03-02

Outline

- **Preliminaries and Introduction**
- **Part I: Degree of Nondeterminism**
- **Part II: Number of Nondeterministic Rules**
- **Part III: Overall Nondeterminism**
- **Concluding Remarks and Open Problems**

Acknowledgment

This presentation is partially based on: A. Meduna, L. Vrábel, P. Zemek: On Nondeterminism in Programmed Grammars, In: *AFL'11: Automata and Formal Languages 2011* (submitted).

Definition

A *programmed grammar* is a quintuple

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

where

- N is an alphabet of *nonterminals*;
- T is an alphabet of *terminals* ($N \cap T = \emptyset$);
- $S \in N$ is the *starting nonterminal*;
- Ψ is an alphabet of *rule labels*;
- P is a finite set of *rules* of the form

  $$(r\colon A \rightarrow x, \sigma_r),$$

  where $r \in \Psi$, $A \in N$, $x \in (N \cup T)^*$, and $\sigma_r \subseteq \Psi$.

Definition

The relation of a *direct derivation*, symbolically denoted by $\Rightarrow$, is defined over $(N \cup T)^* \times \Psi$ as follows:

$$(u, r) \Rightarrow (v, s)$$

if and only if

$$u = u_1 A u_2,\ v = u_1 x u_2,\ (r\colon A \to x, \sigma_r) \in P, \text{ and } s \in \sigma_r.$$

Definition

The relation of a *direct derivation*, symbolically denoted by $\Rightarrow$, is defined over $(N \cup T)^* \times \Psi$ as follows:

$$(u, r) \Rightarrow (v, s)$$

if and only if

$$u = u_1 A u_2,\ v = u_1 x u_2,\ (r\colon A \to x, \sigma_r) \in P, \text{ and } s \in \sigma_r.$$

Definition

The *language generated by G*, $L(G)$, is defined as

$$L(G) = \{w \in T^* \mid (S, r) \Rightarrow^* (w, s), \text{ for some } r, s \in \Psi\}.$$

Programmed Grammars – Definition

Definition

The relation of a *direct derivation*, symbolically denoted by $\Rightarrow$, is defined over $(N \cup T)^* \times \Psi$ as follows:

$$(u, r) \Rightarrow (v, s)$$

if and only if

$$u = u_1 A u_2,\ v = u_1 x u_2,\ (r\colon A \rightarrow x, \sigma_r) \in P, \text{ and } s \in \sigma_r.$$

Definition

The *language generated by G*, $L(G)$, is defined as

$$L(G) = \{w \in T^* \mid (S, r) \Rightarrow^* (w, s), \text{ for some } r, s \in \Psi\}.$$

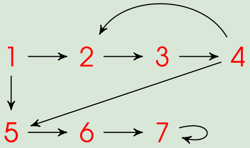
P . . . the family of languages generated by programmed grammars

Programmed Grammars – Example

Example

$$
\begin{aligned}
&(1\colon S \to ABC, \{2,5\})\\
&(2\colon A \to aA, \{3\})\\
&(3\colon B \to bB, \{4\})\\
&(4\colon C \to cC, \{2,5\})\\
&(5\colon A \to a, \{6\})\\
&(6\colon B \to b, \{7\})\\
&(7\colon C \to c, \{7\})
\end{aligned}
$$

$$
\begin{aligned}
(S,1) &\Rightarrow (ABC,2)\\
&\Rightarrow (aABC,3)\\
&\Rightarrow (aAbBC,4)\\
&\Rightarrow (aAbBcC,5)\\
&\Rightarrow (aabBcC,6)\\
&\Rightarrow (aabbcC,7)\\
&\Rightarrow (aabbcc,7)
\end{aligned}
$$



$$L(G) = \{a^n b^n c^n \mid n \geq 1\}$$

Definition

Let $G = (N, T, S, \Psi, P)$ be a programmed grammar. G is of *degree of nondeterminism n*, where $n \geq 1$, if every $(r\colon A \to x, \sigma_r) \in P$ satisfies

$$\text{card}(\sigma_r) \leq n.$$

By $\text{dnd}(G)$, we denote the degree of nondeterminism of G.

$\mathbf{DND}(\mathbf{P}, n)$ . . . the family of languages generated by programmed grammars of degree of nondeterminism n

Question

What happens if we limit the degree of nondeterminism?

Question

What happens if we limit the degree of nondeterminism?

Theorem

$\mathbf{DND}(\mathbf{P}, 1) = \mathbf{FIN}$

FIN . . . the family of finite languages

Question

What happens if we limit the degree of nondeterminism?

Theorem

$\mathbf{DND}(\mathbf{P}, 1) = \mathbf{FIN}$

FIN . . . the family of finite languages

Theorem

$\mathbf{DND}(\mathbf{P}, 2) = \mathbf{P}$

Part II: Number of Nondeterministic Rules

Question

What happens if we limit the number of nondeterministic rules?

Question

What happens if we limit the number of nondeterministic rules?

${}_n\mathbf{P}$ . . . the family of languages generated by programmed grammars with n nondeterministic rules

Question

What happens if we limit the number of nondeterministic rules?

${}_{n}\mathbf{P}$ . . . the family of languages generated by programmed grammars with n nondeterministic rules

Theorem

${}_{1}\mathbf{P} = \mathbf{P}$

Definition

Let $G = (N, T, S, \Psi, P)$ be a programmed grammar. For each $(r\colon A \to x, \sigma_r) \in P$, let $\zeta(r)$ be defined as

$$\zeta(r) = \begin{cases} \mathrm{card}(\sigma_r) & \text{if } \mathrm{card}(\sigma_r) \geq 2 \\ 0 & \text{otherwise.} \end{cases}$$

The *overall nondeterminism of G* is denoted by $\mathrm{ond}(G)$ and defined as

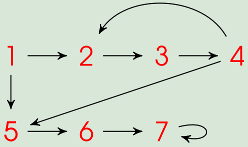
$$\mathrm{ond}(G) = \sum_{r \in \Psi} \zeta(r).$$

$\mathbf{OND}(\mathbf{P}, n)$ . . . the family of languages generated by programmed grammars with overall nondeterminism n

Part III: Overall Nondeterminism – Example

Example

$(1\colon S \to ABC, \{2,5\})$
$(2\colon A \to aA, \{3\})$
$(3\colon B \to bB, \{4\})$
$(4\colon C \to cC, \{2,5\})$
$(5\colon A \to a, \{6\})$
$(6\colon B \to b, \{7\})$
$(7\colon C \to c, \{7\})$



ond(G)

$\mathrm{ond}(G) = 4$

Part III: Overall Nondeterminism

Question

What happens if we limit the overall nondeterminism?

Question

What happens if we limit the overall nondeterminism?

Theorem

$\mathbf{OND}(\mathbf{P}, n) \subset \mathbf{OND}(\mathbf{P}, n + 1)$

Open Problems

- Appearance checking?
- Propagating programmed grammars?

References

M. Barbaiani, C. Bibire, J. Dassow, A. Delaney, S. Fazekas, M. Ionescu, G. Liu, A. Lodhi, and B. Nagy.
The power of programmed grammars with graphs from various classes.
Journal of Applied Mathematics & Computing, 22(1–2):21–38, 2006.

H. Bordihn and M. Holzer.
Programmed grammars and their relation to the LBA problem.
Acta Informatica, 43(4):223–242, 2006.

J. Dassow and G. Păun.
Regulated Rewriting in Formal Language Theory.
Springer, New York, 1989.

D. J. Rosenkrantz.
Programmed grammars and classes of formal languages.
Journal of the ACM, 16(1):107–131, 1969.

The thank you slide.