

# Other Grammars

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

## State Grammar

$$G = (V, T, K, P, S, p_0)$$

where

$V$  is a finite alphabet

$T$  is a set of terminals,  $T \subset V$

$K$  is a finite set of **states**

$S$  is the start symbol,  $S \in V - T$

$p_0$  is the start state,  $p_0 \in K$

$P$  is a finite set of productions of the form

$$(A, p) \rightarrow (x, q),$$

where  $p, q \in K$ ,  $A \in V - T$ ,  $x \in V^*$

# State Grammar – Derivation Step

## Derivation Step

For  $(A, p) \rightarrow (x, q) \in P$ ,

$$u = (rAs, p),$$

$$v = (rxs, q),$$

where  $r, s \in V^*$ , and for every  $(B, p) \rightarrow (y, t) \in P$ ,  $B \notin \text{alph}(r)$ , we write

$$u \Rightarrow v [(A, p) \rightarrow (x, q)]$$

## Generated Language

$$L(G) = \{x \in T^* : (S, p_0) \Rightarrow^* (x, q) \text{ for some } q \in K\}$$

## Generative Power

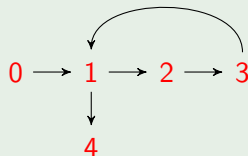
$$\mathcal{L}(ST) = \mathcal{L}(CS), \text{ and } \mathcal{L}(ST, \varepsilon) = \mathcal{L}(RE)$$

# State Grammar – Example

## Example

$$G = (\{S, A, B, C, a, b, c\}, \{a, b, c\}, \{0, 1, 2, 3, 4\}, P, S, 0)$$

$$P = \{(S, 0) \rightarrow (ABC, 1), \\ (A, 1) \rightarrow (aA, 2), \quad (A, 1) \rightarrow (a, 4), \\ (B, 2) \rightarrow (bB, 3), \quad (B, 4) \rightarrow (b, 4), \\ (C, 3) \rightarrow (cC, 1), \quad (C, 4) \rightarrow (c, 4)\}$$



$$\begin{aligned} (S, 0) &\Rightarrow (ABC, 1) \Rightarrow (aABC, 2) \Rightarrow (aAbBC, 3) \Rightarrow (aAbBcC, 1) \\ &\Rightarrow (aabBcC, 4) \Rightarrow (aabbcc, 4) \end{aligned}$$

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

## Queue Grammar

$$G = (V, T, W, F, P, s)$$

where

$V$  is a finite alphabet

$T$  is a set of terminals,  $T \subset V$

$W$  is a finite alphabet of **states**

$F$  is a set of final states,  $F \subset W$

$s$  is the start string,  $s \in (V - T)(W - F)$

$P$  is a finite set of productions of the form:  **$(a, b, x, c)$** , where

$$a \in V$$

$$b \in W - F$$

$$x \in V^*$$

$$c \in W$$

# Queue Grammar – Derivation Step

## Derivation Step

If

$$u = arb, \ v = rxc,$$

where  $a \in V$ ,  $r, x \in V^*$ ,  $b, c \in W$ , and

$$(a, b, x, c) \in P,$$

then

$$u \Rightarrow v [(a, b, x, c)]$$

## Generated Language

$$L(G) = \{w \in T^* : s \Rightarrow^* wf, f \in F\}$$

## Generative Power

$$\mathcal{L}(QG, \varepsilon) = \mathcal{L}(RE), \text{ and } \mathcal{L}(QG) = \mathcal{L}(CS)$$

## Example

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

$$P = \{1 : (A, \bar{a}, bAa, \bar{a}), \\ 2 : (A, \bar{a}, \varepsilon, \bar{b}), \\ 3 : (a, \bar{a}, a, \bar{a}), \\ 4 : (b, \bar{a}, b, \bar{a})\}$$

$$A\bar{a} \Rightarrow bAa\bar{a} [1] \Rightarrow Ab\bar{a} [4] \Rightarrow abbAa\bar{a} [1] \Rightarrow bbAaa\bar{a} [3] \Rightarrow bAaab\bar{a} [4] \\ \Rightarrow Aaabb\bar{a} [4] \Rightarrow aabb\bar{b} [2]$$

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

# Simple Matrix Grammar

## Simple Matrix Grammar (of Degree $n$ )

$$G = (N_1, \dots, N_n, T, P, S)$$

where

$N_1, \dots, N_n$  are pairwise disjoint nonterminal alphabets

$P$  is a finite set of productions of the form

- $(S \rightarrow w)$ , where  $w \in T^*$
- $(S \rightarrow A_1 \dots A_n)$ , where  $A_i \in N_i$ ,  $i = 1, \dots, n$
- $(A_1 \rightarrow w_1, \dots, A_n \rightarrow w_n)$ , where  $w_i \in T^*$ ,  $i = 1, \dots, n$
- $(A_1 \rightarrow \alpha_1, \dots, A_n \rightarrow \alpha_n)$ , where

$$\alpha_i = x_{i1}A_{i1} \dots x_{ik}A_{ik}y_i,$$

$$x_{ij}, y_i \in T^*, A_{ij} \in N_i, i = 1, \dots, n, j = 1, \dots, k, k \geq 1$$

$T, S$  have the standard meaning,  $S \notin N_1 \cup \dots \cup N_n$



# Simple Matrix Grammar – Derivation Step

## Direct Derivation

If

- either  $u = S$  and  $(S \rightarrow v) \in P$

- or

- $u = y_1 A_1 z_1 \dots y_n A_n z_n$

- $v = y_1 w_1 z_1 \dots y_n w_n z_n$

- $(A_1 \rightarrow w_1, \dots, A_n \rightarrow w_n) \in P$

where  $y_i \in T^*$ ,  $A_i \in N_i$ ,  $z_i \in (N_i \cup T)^*$ , for all  $i = 1, \dots, n$

then

$$u \Rightarrow v$$

## Note

The generated language and  $\Rightarrow^*$  are defined as usual

# Simple Matrix Grammar – Generative Power

## (Right) Linear Simple Matrix Grammar (of Degree $n$ )

If the productions  $A_i \rightarrow \alpha_i$ ,  $1 \leq i \leq n$ , from  $(A_1 \rightarrow \alpha_1, \dots, A_n \rightarrow \alpha_n)$  are

linear then  $G$  is linear simple matrix grammar (*LSM*)

right linear then  $G$  is right linear simple matrix grammar (*RLSM*)

## Generative Power

For all  $n \geq 1$ ,

- $\mathcal{L}(SM, n) \subset \mathcal{L}(SM, n+1) \subset \mathcal{L}(CS)$
- $\mathcal{L}(LSM, n) \subset \mathcal{L}(LSM, n+1)$
- $\mathcal{L}(RLSM, n) \subset \mathcal{L}(RLSM, n+1)$

# Right Linear Simple Matrix Grammar – Example

## Example

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

where

$$P = \{1 : (S \rightarrow AB), \\ 2 : (A \rightarrow aA, B \rightarrow aB), \\ 3 : (A \rightarrow bA, B \rightarrow bB), \\ 4 : (A \rightarrow \varepsilon, B \rightarrow \varepsilon)\}$$

$$S \Rightarrow AB [1] \Rightarrow aAaB [2] \Rightarrow abAabB [3] \Rightarrow abaAabaB [2] \Rightarrow abaaba [4]$$

$$L(G) = \{ww : w \in \{a, b\}^*\}$$

# Bibliography I



O. Ibarra.

Simple matrix languages.

*Information and Control*, 17(4):359–394, 1970.



T. Kasai.

An hierarchy between context-free and context-sensitive languages.

*Journal of Computer and System Sciences*, 4(5):492–508, 1970.



H. C. M. Kleijn and G. Rozenberg.

On the generative power of regular pattern grammars.

*Acta Informatica*, 20:391–411, 1983.



W. Kuich and H. A. Maurer.

Tuple languages.

In *Proceedings on International Computer Symposium*, pages 881–891, Bonn, 1970.



Gh. Păun.

Linear simple matrix languages.

*Elektronische Informationsverarbeitung und Kybernetik*,  
14(7/8):377–384, 1978.