

Multigenerative grammar systems

Zámečníková Eva
FIT BUT

2.12.2009

Outline

- Multigenerative grammar systems
 - Canonical multigenerative grammar systems
 - General multigenerative grammar systems
 - Hybrid multigenerative grammar systems
 - Definitions
 - Examples
 - Generative power

Canonical multigenerative grammar systems

Definition:

- $(n+1)$ -tuple $\Gamma = (G_1, G_2, \dots, G_n, Q)$
where
 - $G_i = (N_i, T_i, P_i, S_i)$ is CFG for each $i = 1, 2, \dots, n$
 - Q – control component – two variants
 - finite set in the form (A_1, A_2, \dots, A_n) , where $A_i \in N_i$
 - finite set in the form (p_1, p_2, \dots, p_n) , where $p_i \in P_i$ for each $i = 1, 2, \dots, n$.

Multiform

- Let $\Gamma = (G_1, G_2, \dots, G_n, Q)$ be a n -CGN. Then sentential ***multiform*** is n -tuple $\chi = (x_1, x_2, \dots, x_n)$ where $x_i \in (N \cup T)^*$ for all $i = 1, 2, \dots, n$.

Canonical multigenerative grammar systems

- Definition of direct derivation step:
 - Let $\Gamma = (G_1, G_2, \dots, G_n, Q)$ be a n -CGN.
 - Let $\chi_1 = (u_1 A_1 v_1, u_2 A_2 v_2, \dots, u_n A_n v_n)$ and $\chi_2 = (u_1 x_1 v_1, u_2 x_2 v_2, \dots, u_n x_n v_n)$ be two multiforms, where $A_i \in N_i$, $u_i \in T_i^*$, $v_i, x_i \in (N_i \cup T_i)^*$ for each $i = 1, 2, \dots, n$.
 - Let $A_i \rightarrow x_i \in P_i$ for each $i = 1, 2, \dots, n$.
 - Let $(A_1, A_2, \dots, A_n) \in Q$.
- Then χ_1 directly derive χ_2 , we write $\chi_1 \Rightarrow \chi_2$.

Generated n -language

Let $\Gamma = (G_1, G_2, \dots, G_n, Q)$ be a n -CGN. Then the generated n -language $n\text{-}L(\Gamma)$ is defined:

$n\text{-}L(\Gamma) = \{(w_1, w_2, \dots, w_n) :$

$(S_1, S_2, \dots, S_n) \Rightarrow^* (w_1, w_2, \dots, w_n),$

$w_i \in T_i^*$ for all $i = 1, 2, \dots, n\}$

- 3 different modes
 - L_{union}
 - L_{conc}
 - L_{first}

Example of nonterminal synchronized canonical multigenerative grammar system (2-CGN)

- Let's have 3-tuple $\Gamma = (G_1, G_2, Q)$ where:
 - $G_1 = (\{S_1, A_1\}, \{a, b, c\}, \{1:S_1 \rightarrow aS_1, 2:S_1 \rightarrow aA_1, 3:A_1 \rightarrow bA_1c, 4:A_1 \rightarrow bc\}, S_1)$
 - $G_2 = (\{S_2, A_2\}, \{d\}, \{1:S_2 \rightarrow S_2A_2, 2:S_2 \rightarrow A_2, 3:A_2 \rightarrow d\}, S_2)$
 - $Q = \{(S_1, S_2), (A_1, A_2)\}$
- We can generate e.g. these sequences:
 - $(S_1, S_2) \Rightarrow (aA_1, A_2) \Rightarrow (abc, d)$
 - $(S_1, S_2) \Rightarrow (aS_1, S_2A_2) \Rightarrow (aaA_1, A_2A_2) \Rightarrow (aabA_1c, dA_2) \Rightarrow (aabbcc, dd)$
 - ...

Illustration of previous example

Rules of G_1 :

$$1: S_1 \rightarrow aS_1$$

$$2: S_1 \rightarrow aA_1$$

$$3: A_1 \rightarrow bA_1c$$

$$4: A_1 \rightarrow bc$$

Rules of G_2 :

$$1: S_2 \rightarrow S_2A_2$$

$$2: S_2 \rightarrow A_2$$

$$3: A_2 \rightarrow d$$

Q :

$$1: (S_1, S_2)$$

$$2: (A_1, A_2)$$

$$\begin{array}{c} S_1 \\ [1] \downarrow \\ aS_1 \\ [2] \downarrow \\ aaA_1 \\ [3] \downarrow \\ aabA_1c \\ [4] \downarrow \\ aabbcc \end{array}$$

$$\begin{array}{c} S_2 \\ [1] \downarrow \\ S_2A_2 \\ [2] \downarrow \\ A_2A_2 \\ [3] \downarrow \\ dA_2 \\ [3] \downarrow \\ dd \end{array} \quad \begin{array}{c} (S_1, S_2) \in Q \\ (S_1, S_2) \in Q \\ (A_1, A_2) \in Q \\ (A_1, A_2) \in Q \end{array}$$

Generated languages in different modes

- Generated n -language:
$$2\text{-L}(\Gamma) = \{(a^n b^n c^n, d^n) : n \geq 1\}$$
- Generated languages in different modes:
$$L_{union}(\Gamma) = \{a^n b^n c^n : n \geq 1\} \cup \{d^n : n \geq 1\}$$
$$L_{conc}(\Gamma) = \{a^n b^n c^n d^n : n \geq 1\}$$
$$L_{first}(\Gamma) = \{a^n b^n c^n : n \geq 1\}$$

Example of rule synchronized canonical multigenerative grammar system (2-CGR)

- Let's have 3-tuple $\Gamma = (G_1, G_2, Q)$ where:
 - $G_1 = (\{S_1, A_1\}, \{a, b, c\}, \{1:S_1 \rightarrow aS_1, 2:S_1 \rightarrow aA_1, 3:A_1 \rightarrow bA_1c, 4:A_1 \rightarrow bc\}, S_1)$
 - $G_2 = (\{S_2, A_2\}, \{d\}, \{1:S_2 \rightarrow S_2A_2, 2:S_2 \rightarrow A_2, 3:A_2 \rightarrow d\}, S_2)$
 - $Q = \{(1, 1), (2, 2), (3, 3), (4, 3)\}$
- We can generate e.g. these sequences
 - $(S_1, S_2) \Rightarrow (aA_1, S_2) \Rightarrow (abc, d)$
 - $(S_1, S_2) \Rightarrow (aS_1, S_2A_2) \Rightarrow (aaA_1, A_2A_2) \Rightarrow (aabA_1c, dA_2) \Rightarrow (aabbc, dd)$
 - ...

Illustration of previous example

Rules of G_1 :

$$1: S_1 \rightarrow aS_1$$

$$2: S_1 \rightarrow aA_1$$

$$3: A_1 \rightarrow bA_1c$$

$$4: A_1 \rightarrow bc$$

Rules of G_2 :

$$1: S_2 \rightarrow S_2A_2$$

$$2: S_2 \rightarrow A_2$$

$$3: A_2 \rightarrow d$$

Q :

$$\begin{array}{ll} (1,1) & (2,2) \\ (3,3) & (4,3) \end{array}$$

$$\begin{array}{c} S_1 \\ [1] \downarrow \\ aS_1 \\ [2] \downarrow \\ aaA_1 \\ [3] \downarrow \\ aabA_1c \\ [4] \downarrow \\ aabbcc \end{array}$$

$$\begin{array}{c} S_2 \\ [1] \downarrow \\ S_2A_2 \\ [2] \downarrow \\ A_2A_2 \\ [3] \downarrow \\ dA_2 \\ [3] \downarrow \\ dd \end{array}$$

$$(1, 1) \in Q$$

$$(2, 2) \in Q$$

$$(3, 3) \in Q$$

$$(4, 3) \in Q$$

Relation between n -CGN and n -CGR and generative power

- conversion of any n -CGN system to n -CGR system and back
- equation of generated n -languages
- Generative power:
 - $n\text{-L}(n\text{-CGN}) = n\text{-L}(n\text{-CGR})$
 - $L_X(n\text{-CGN}) = L_Y(n\text{-CGR})$, where $X, Y \in \{\text{union}, \text{conc}, \text{first}\}$
 - for each $L(RE)$ over T exists **2-CGR** = (G_1, G_2, Q) such that:
 $L_{\text{union}} = L(RE)$
 $L_{\text{conc}} = L(RE)$
 $L_{\text{first}} = L(RE)$

General multigenerative grammatical systems

Definition:

- $(n+1)$ -tuple $\Gamma = (G_1, G_2, \dots, G_n, Q)$

where

- $G_i = (N_i, T_i, P_i, S_i)$ is CFG for each $i = 1, 2, \dots, n$
- Q - control component – two variants
 - finite set in the form (A_1, A_2, \dots, A_n) , where $A_i \in N_i$
 - finite set in the form (p_1, p_2, \dots, p_n) , where $p_i \in P_i$ for each $i = 1, 2, \dots, n$.

- generated n -language:

$$n\text{-}L(\Gamma) = \{(w_1, w_2, \dots, w_n) :$$

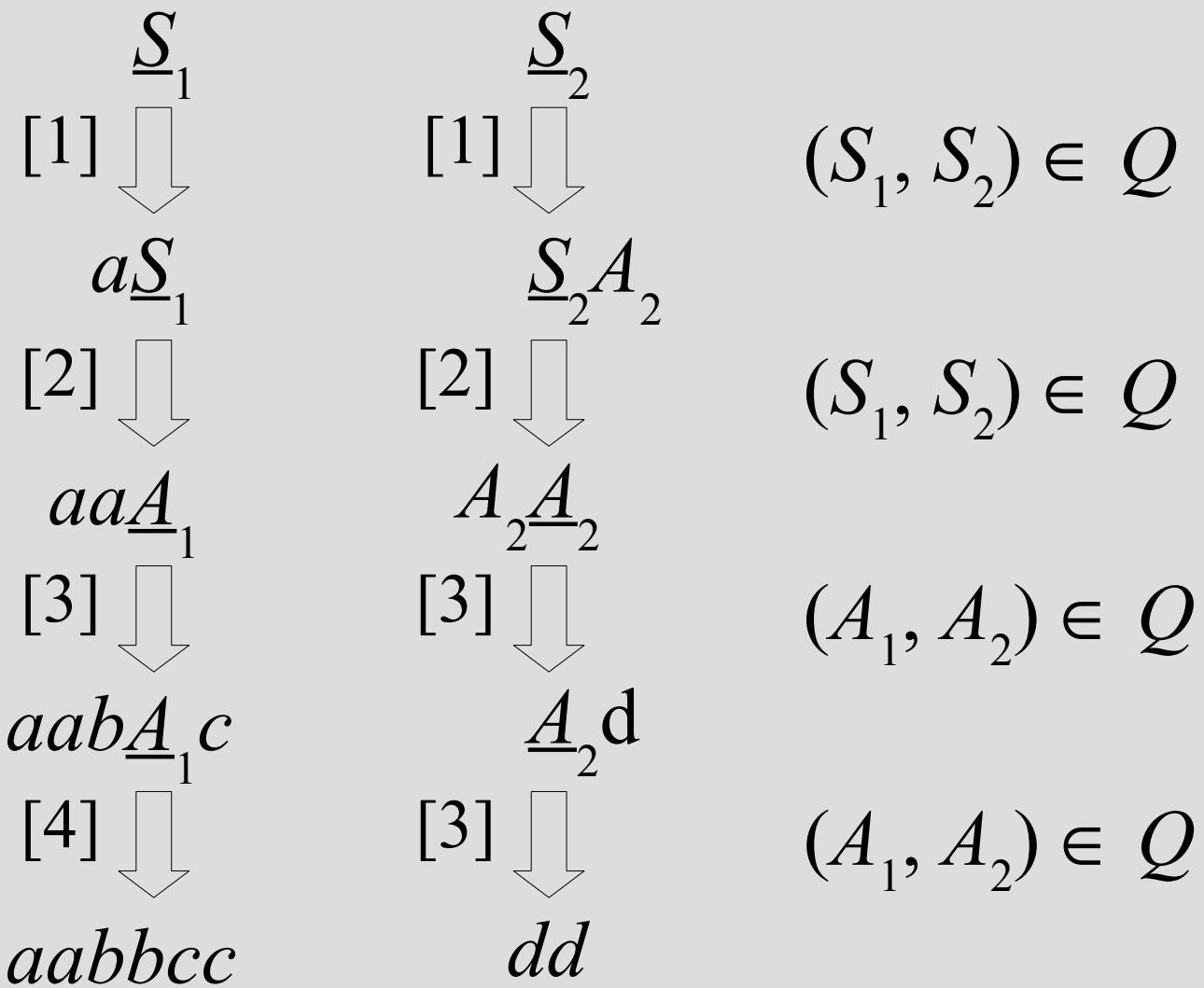
$$(S_1, S_2, \dots, S_n) \Rightarrow^* (w_1, w_2, \dots, w_n),$$

for all $i = 1, 2, \dots, n$

- 3 different modes

Example of nonterminal synchronized general multigenerative grammar system (2-GGN)

Rules of G_1:
1: $S_1 \rightarrow aS_1$
2: $S_1 \rightarrow aA_1$
3: $A_1 \rightarrow bA_1c$
4: $A_1 \rightarrow bc$
Rules of G_2:
1: $S_2 \rightarrow S_2A_2$
2: $S_2 \rightarrow A_2$
3: $A_2 \rightarrow d$
Q:
1: (S_1, S_2)
2: (A_1, A_2)



Relations between n -GGN and n -GGR

- conversion of any n -GGN system to n -GGR system and back
- equation of generated n -languages
- we can convert n -GGR in any mode to equivalent matrix grammar and back
 - reduced number of components to 2
- Generative power
 - $L_X(n\text{-GGN}) = L_Y(n\text{-GGR})$, where $X, Y \in \{\text{union}, \text{conc}, \text{first}\}$

Hybrid multigenerative systems

- same definition as n -CGN resp. n -CGR or n -GGN resp. n -GGR
- explicit specification of use of leftmost derivation or general derivation
- used in practise
 - generalized translation among more languages

Hybrid multigenerative grammar systems

Multitranslation

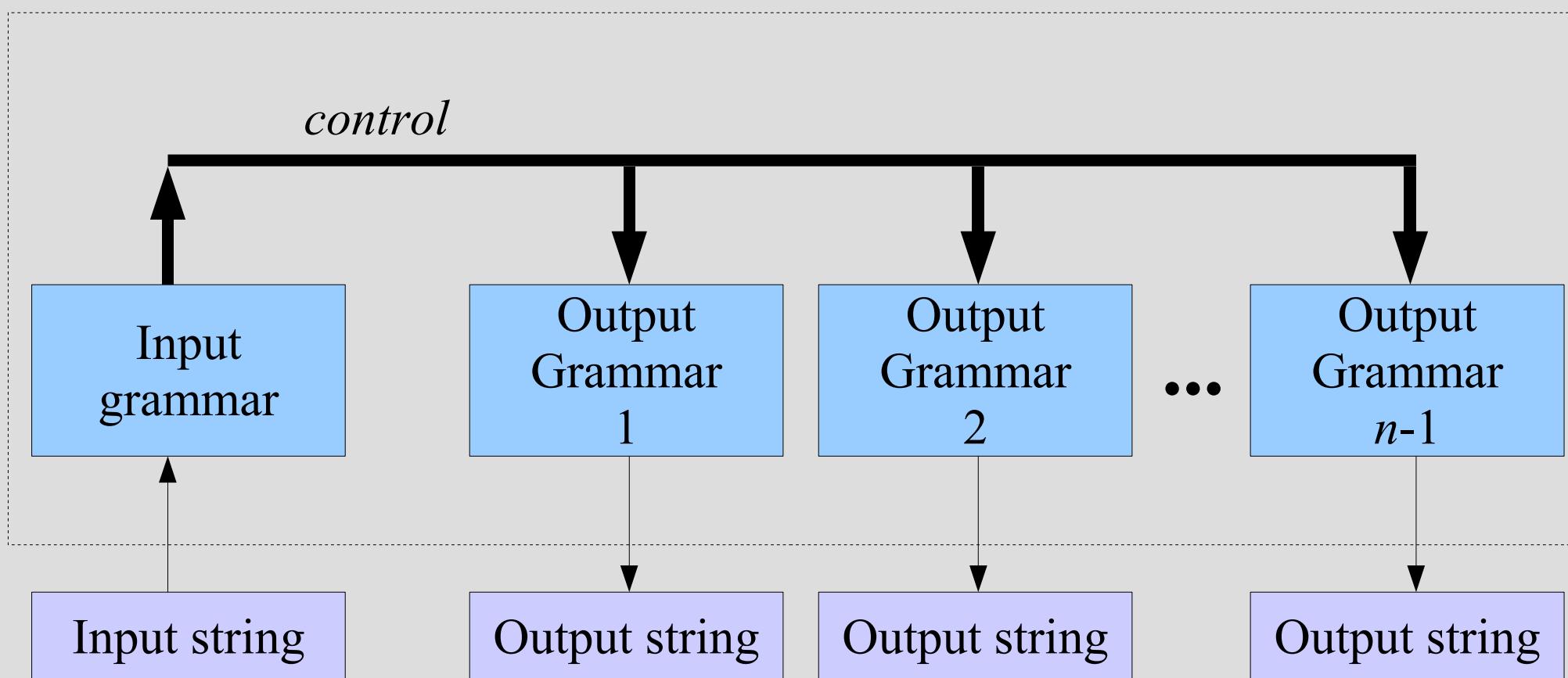
- using output multistrings which are generated from multigenerative grammar systems we can define n -ary relation, so some *general translation* between more languages

Multitranslation

- one grammar is *input*, other $n-1$ grammars are *output*
- using the input grammar we make syntactic analysis of an input string which in this grammar system controls generating of $n-1$ output strings using output grammars

Scheme of multitranslation

- Grammar system consisting of n grammars



Use of multitranslation model

- translator can be used as a deterministic acceptor of some more difficult languages
- a sentence from one natural language can be translated to more other natural languages
- assembly code to be translated to more binary codes for different processors

Multitranslation

- we define few restriction of grammar systems to make the multitranslate *deterministic*

Multitranslation can be:

- serial
- parallel

Serial translation

Procedure

- syntactic analysis of an input string is done
- left or right parse is given to output grammars and on base of this information are generated output strings
- all steps listed above are done one by one

Parallel translation

- syntactic analysis of an input string also controls the generating of output strings by output grammars
- at the moment when the syntactic analysis of input string is finished, the output grammars already have generated the output strings
- all is done in parallel

Literature

Lukáš, R.: Multigenerativní gramatické sytémy,
disertační práce, Brno, FIT VUT v Brně, 2006