PARALLEL PARSING BASED UPON GENERAL MULTIGENERATIVE GRAMMAR SYSTEMS

Zbyněk Sopuch
TID 2011
Supervisor: Alexander Meduna

Contents

- Multigenerative grammar systems
  - Classification
  - Synchronization
  - Modes of *n*-languages
- Independent grammar parsing
- Keeping synchronization during parsing
  - Parsing with simulation
  - Parsing without simulation
- Issues during parsing of modes

Multigenerative grammar system (MGS)

- *n*-generative grammar system:
  (*n+1*)-tuple $\Gamma = (G_1, G_2, \ldots, G_n, Q)$, where:
  - G_i … $i = 1..n$, a context free grammar
  - Q … a synchronization component
- The number of grammars can be reduce to 2 without any effect on a generative power

Classification of MGS

- Canonical multigenerative grammar systems
  - G_i is a *LL*-grammar
- General multigenerative grammar systems
  - G_i is a classic context free grammar
- Hybrid multigenerative grammar systems
  - G_i can be a classic CFG or a *LL*-grammar, but the type of each must be known

Synchronization of MGS

- Nonterminal-synchronized (*n*-MGN)
  - Q is set of *n*-tuples of the form:

    $$(A_1, \ldots, A_n): A_i \in N_i$$

- Rule-synchronized (*n*-MGR)
  - Q is set of *n*-tuples of the form:

    $$(p_1, \ldots, p_n): p_i \in P_i$$

- *The generative power of n-MGR and n-MGN is the same (can be automatically convert).*

n-language of n-MGN

- n-string $\chi = (x_1, x_2, \ldots, x_n)$, where $x_i \in (N_i \cup T_i)^*$
- $\chi \Rightarrow \chi`$ and $\chi \Rightarrow^* \chi`$ in the common way
  - $\chi = (u_1 A_1 v_1, u_2 A_2 v_2, \ldots, u_n A_n v_n)$
  - $\chi` = (u_1 x_1 v_1, u_2 x_2 v_2, \ldots, u_n x_n v_n)$
  - p_i: $A_i \rightarrow x_i \in P_i$, where $(A_1, A_2, \ldots, A_n) \in Q$
- If n-MGN Γ, then $n\text{-}L(\Gamma) = \{(w_1, w_2, \ldots, w_n),: (S_1, S_2, \ldots, S_n) \Rightarrow^* (w_1, w_2, \ldots, w_n)\}$

Example of n-MGN

- $\Gamma = (G_1, G_2, Q)$ is n-MGN, where:
  - $G_1 = (\{S_1, A_1\}, \{a, b, c\},$
    $\{S_1 \longrightarrow aS_1, S_1 \longrightarrow aA_1, A_1 \longrightarrow bA_1c, A_1 \longrightarrow bc\}, S_1)$
  - $G_2 = (\{S_2, A_2\}, \{d\}, \{S_2 \longrightarrow S_2A_2, S_2 \longrightarrow A_2, A_2 \longrightarrow d\}, S_2)$
  - $Q = \{(S_1, S_2), (A_1, A_2)\}$

- $L_1(G_1) = \{a^n b^m c^m \mid n > 0, m > 0\}$
- $L_2(G_2) = \{d^n \mid n > 0\}$
- n-language $n\text{-}L(\Gamma) = \{(a^n b^n c^n, d^n) \mid n > 0\}$

- $\Gamma = (G_1, G_2, Q)$ is n-MGN, where:
  - $G_1 = (\{S_1, A_1\}, \{a, b, c\},$
    $\{S_1 \rightarrow aS_1, S_1 \rightarrow aA_1, A_1 \rightarrow bA_1c, A_1 \rightarrow bc\}, S_1)$
  - $G_2 = (\{S_2, A_2\}, \{d\}, \{S_2 \rightarrow S_2A_2, S_2 \rightarrow A_2, A_2 \rightarrow d\}, S_2)$
  - $Q = \{(S_1, S_2), (A_1, A_2)\}$

$\underline{S}$	$\Leftarrow$	$(S_1, S_2) \in Q$	$\Rightarrow$	$\underline{S}$
$a\,\underline{S}$	$\Leftarrow$	$(S_1, S_2) \in Q$	$\Rightarrow$	$\underline{S}\,A$
$a\,a\,\underline{A}$	$\Leftarrow$	$(A_1, A_2) \in Q$	$\Rightarrow$	$\underline{A}\,A$
$a\,a\,b\,\underline{A}\,c$	$\Leftarrow$	$(A_1, A_2) \in Q$	$\Rightarrow$	$d\,\underline{A}$
$a\,a\,b\,b\,c\,c$	$\Leftarrow$	$(A_1, A_2) \in Q$	$\Rightarrow$	$d\,d$

Example of n-MGR

- $\Gamma = (G_1, G_2, Q)$ is n-MGR, where:
  - $G_1 = (\{S_1, A_1\}, \{a, b, c\},$
    $\{1\text{: } S_1 \rightarrow aS_1, 2\text{: } S_1 \rightarrow aA_1, 3\text{: } A_1 \rightarrow bA_1c, 4\text{: } A_1 \rightarrow bc\},$
    $S_1)$
  - $G_2 = (\{S_2, A_2\}, \{d\},$
    $\{1\text{: } S_2 \rightarrow S_2A_2, 2\text{: } S_2 \rightarrow A_2, 3\text{: } A_2 \rightarrow d\}, S_2)$
  - $Q = \{(1, 1), (2, 2), (3, 3), (4, 3)\}$
  - n-MGN: $Q = \{(S_1, S_2), (A_1, A_2)\}$

- n-language $n\text{-}L(\Gamma) = \{(a^nb^nc^n, d^n) \mid n > 0\}$

Modes of *n*-language

- *n*-language $\rightarrow$ language: *n*-ary operation $\oplus$

$L_{\oplus} = \{\oplus\, w_1, w_2, \ldots, w_n \mid (w_1, w_2, \ldots, w_n) \in n\text{-}L(\Gamma)\}$

- *Union:*
  - $L_{\text{union}}(\Gamma) = \{w_{1,} w_{2,} \ldots, w_n \mid (w_1, w_2, \ldots, w_n) \in n\text{-}L(\Gamma)\}$
- *Concatenation:*
  - $L_{\text{conc}}(\Gamma) = \{w_1 w_2 \ldots w_n \mid (w_1, w_2, \ldots, w_n) \in n\text{-}L(\Gamma)\}$
- *First component:*
  - $L_{\text{first}}(\Gamma) = \{w_1 \mid (w_1, w_2, \ldots, w_n) \in n\text{-}L(\Gamma)\}$

Example of modes

- $n\text{-}L(\Gamma) = \{(a^n b^n c^n, d^n) \mid n > 0\}$

- *Union:*
  - $L_{\text{union}}(\Gamma) = \{(a^n b^n c^n) \mid n > 0\} \cup \{(d^n) \mid n > 0\}$
- *Concatenation:*
  - $L_{\text{conc}}(\Gamma) = \{(a^n b^n c^n d^n) \mid n > 0\}$
- *First component:*
  - $L_{\text{first}}(\Gamma) = \{(a^n b^n c^n) \mid n > 0\}$
- *The generative power is the same.*

Parsing for general MGR

- $\Gamma = (G_1 , G_2 , \ldots , G_n , Q)$
- n-language $\rightarrow$ n-string $\chi = (x_1, x_2, \ldots, x_n)$

- $x_1 \rightarrow G_1, x_2 \rightarrow G_2, x_3 \rightarrow G_3, \ldots$
- The strings can be assigned to appropriate grammars

- If the strings are parsed independently like CFG:
  - If the parsing of at least one fails, whole parsing fails
  - But if all parsing succeed, the whole parsing can fail

Example of the issue in an independent parsing

- $\Gamma = (G_1, G_2, Q)$ is n-MGN, where:
  - $G_1 = (\{S_1, A_1\}, \{a, b, c\},$
    $\{S_1 \rightarrow aS_1, S_1 \rightarrow aA_1, A_1 \rightarrow bA_1c, A_1 \rightarrow bc\}, S_1)$
  - $G_2 = (\{S_2, A_2\}, \{d\}, \{S_2 \rightarrow S_2A_2, S_2 \rightarrow A_2, A_2 \rightarrow d\}, S_2)$
  - $Q = \{(S_1, S_2), (A_1, A_2)\}$

- $L_1(G_1) = \{a^n b^m c^m\}$, $L_2(G_2) = \{d^n\}$
- n-language $n\text{-}L(\Gamma) = \{(a^n b^n c^n, d^n)\}$

- aabbbccc $\in L_1$, dd $\in L_2$, (aabbbccc, dd) $\notin n\text{-}L(\Gamma)$
- Missing a synchronization which is forbidding some derivations

Inclusion of synchronization

- After the parsing phase
  - Independent parsing of CFGs with back verification of synchronization
  - Useful for the modes *first component* and *union*

- During the parsing phase
  - Inclusion of synchronization to process of parsing
  - Can be used for the mode *concatenation*

Back verification of synchronization

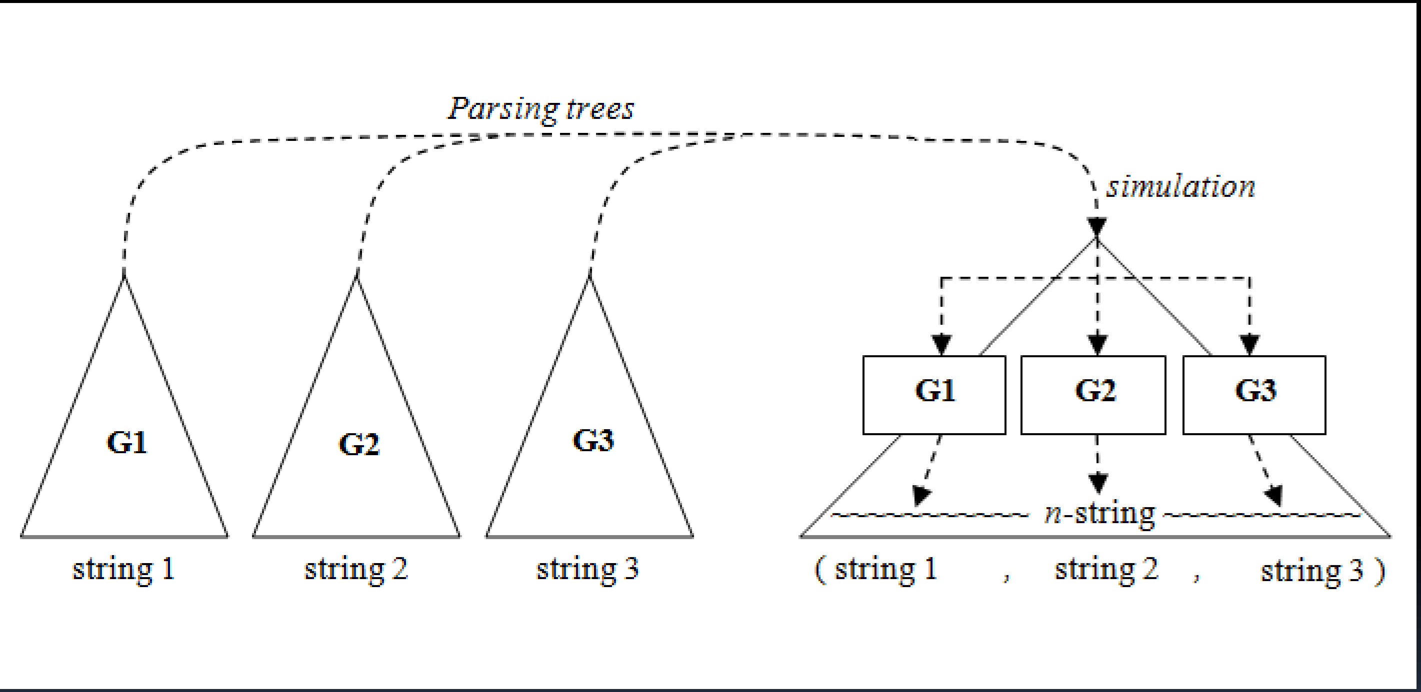

Figure 1: Back verification of synchronization.

Issues of back verification

- Different parse trees for one string
  - The helpful limitation:
    There must be tree of the same height for each string.
- Halting problem (cycle in a parsing)
  - Can be partly solved: if there is at least one grammar with limited number of parse trees (without cycle or deterministic...), we can use it for generating of all possible heights of trees => all other trees have to have the same height as one of its parse tree
  - My solution in my Master‘s thesis was based on using a CYK normal form, but it was connected with decrease of generative power, because we can‘t generate strings of some length with binary rules

Issue of slowing rules

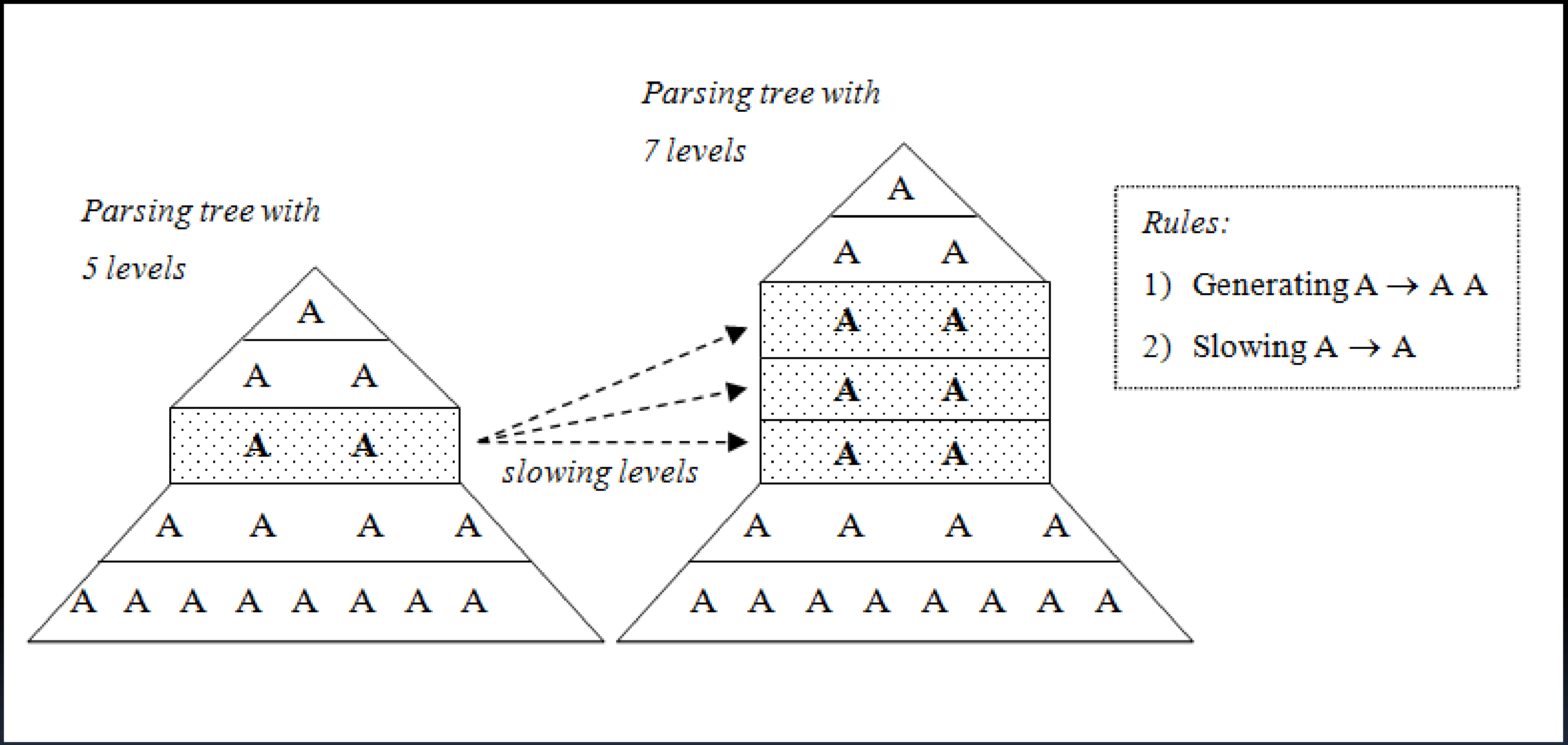

Figure 2: Two different trees for the same string

Involving a synchronization during the parsing phase

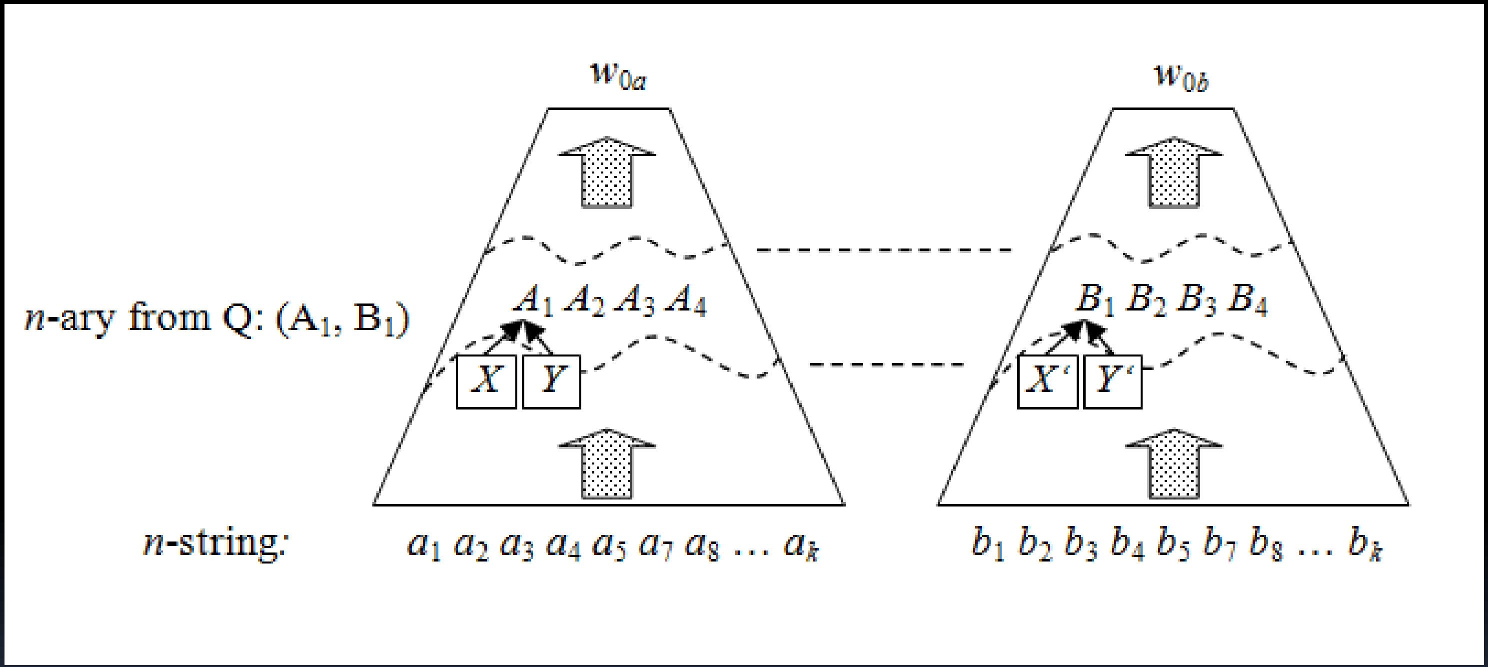

Figure 3: Controlling of synchronization during the parsing phase

Issues of „in-the-middle“ verification

- No mathematical prove – yet
- Significant reduction in the number of parsing trees, but not only one tree
- Cycles in the parsing are still possible
- All part of *n*-string are necessary
  - It‘s an issue with modes of *n*-languages

Issues in the parsing of modes of *n*-languages

- The biggest issues is lost of context between the grammars and strings from *n*-strings
  - Except of the mode of the first component, we don‘t know which grammar generated that string
  - Except of the mode of concatenation, there is only one string from *n*-string to parse

- => it‘s necessary to use simulation

Parsing of *n*-languages in the mode of first component

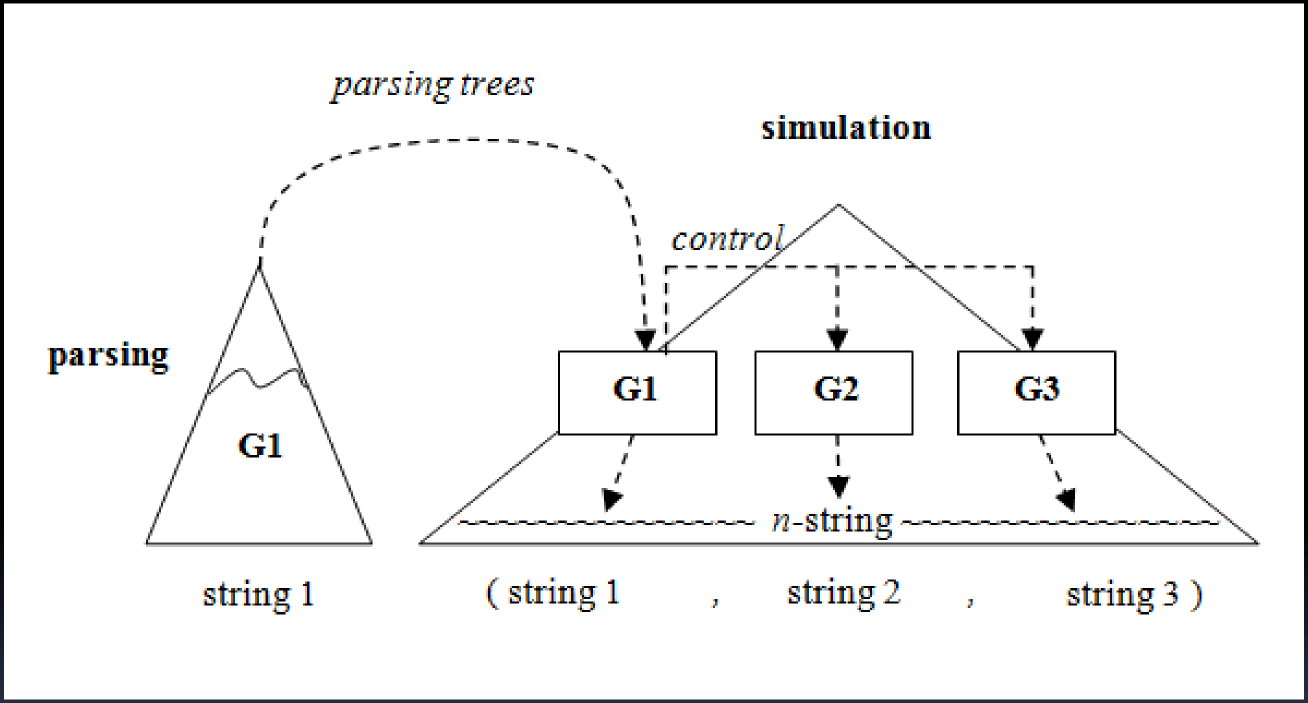

Figure 4: Using simulation to verification in the mode of the first component

Parsing with other modes

- Union: Almost the same as the mode of the first component
  - there is unknown which grammar is the right one => all grammars have to be tested

- Concatenation: Each string have to be spited into the *n* substrings and tested like *n*-string → there are many possibilities how to split

Conclusion

- An interesting young topic (this decade)
  - The stable base - basic rules of MGS are fully described and confirmed by formal proofs
  - A lot of opportunities to explore
- Parsing of general MGS
  - A lot of issues and no formal proves around my hypothesis yet
    - It‘s not deterministic => less effective
    - The number of possible parse trees can be significantly reduced, but still not to only one

References

1. Čermák, M.: *Multi–Languages and Systems of Formal Models*. Brno, FIT BUT, 2010

2. Lukáš, R., Meduna, A.: *Multigenerative grammar systems*. Brno, FIT BUT, 2006

3. Meduna, A., Sopuch, Z.: *Parallel multigenerative CYK-based parsing*. Brno, FIT BUT, 2011

Questions?

THANK YOU FOR YOUR ATTENTIONS