

MULTIGENERATIVE GRAMMAR SYSTEMS

Ing. Roman LUKÁŠ, Doctoral Degree Programme (2)
Dept. of Information Systems, FIT, BUT
E-mail: lukas@fit.vutbr.cz

Supervised by: Dr. Alexander Meduna

ABSTRACT

This paper presents new models for all recursive enumerable languages. These models are based on multigenerative grammar systems that simultaneously generate several strings in a parallel way. The components of these models are context-free grammars, working in a leftmost way. The rewritten nonterminals are determined by a finite set of nonterminal sequences.

1 INTRODUCTION

The formal language theory has recently intensively investigated various grammar systems, which consist of several cooperating components, usually represented by grammars. Although this variety is extremely broad, all these grammar systems always makes a derivation that generates a single string. In this paper, however, we introduce grammar systems that simultaneously generate several strings, which are subsequently composed in a single string by some common string operation, such as concatenation.

More precisely, for a positive integer n , an n -generative grammar system discussed in this paper works with n context-free grammatical components in a leftmost way—that is, in every derivation step, each of these components rewrites the leftmost nonterminal occurring in its current sentential form. These n leftmost derivations are controlled n -tuples of nonterminals or rules. Under a control like this, the grammar system generates n strings, out of which the strings that belong to the generated language are made by some basic operations. Specifically, these operations include union, concatenation and a selection of the string generated by the first component.

2 DEFINITIONS

Definition 2.1: An n -multigenerative nonterminal-synchronized grammar system (n-MGN) is an $n+1$ tuple,

$$\Gamma = (G_1, G_2, \dots, G_n, Q),$$

where $G_i = (N_i, T_i, P_i, S_i)$ is a context-free grammar for each $i = 1, \dots, n$, and Q is a finite set

of n -tuples of the form (A_1, A_2, \dots, A_n) , where $A_i \in N_i$ for all $i = 1, \dots, n$. Let $\Gamma = (G_1, G_2, \dots, G_n, Q)$ be an n -MGN. Then, a *sentential n -form* of n -MGN is an n -tuple of the form $\chi = (x_1, x_2, \dots, x_n)$, where $x_i \in (N_i \cup T_i)^*$ for all $i = 1, \dots, n$. Let $\chi = (u_1A_1v_1, u_2A_2v_2, \dots, u_nA_nv_n)$ and $\bar{\chi} = (u_1x_1v_1, u_2x_2v_2, \dots, u_nx_nv_n)$ be two sentential n -form, where $A_i \in N_i$, $u_i \in T_i^*$, and $v_i, x_i \in (N_i \cup T_i)^*$ for all $i = 1, \dots, n$. Let $A_i \rightarrow x_i \in P_i$ for all $i = 1, \dots, n$ and $(A_1, A_2, \dots, A_n) \in Q$. Then χ directly derives $\bar{\chi}$ in Γ , denoted by $\chi \Rightarrow \bar{\chi}$. In the standard way, we generalize \Rightarrow to \Rightarrow^k , $k \geq 0$, \Rightarrow^+ , and \Rightarrow^* .

The n -language of Γ , n - $L(\Gamma)$, is defined as

$$n\text{-}L(\Gamma) = \{(w_1, w_2, \dots, w_n) : (S_1, S_2, \dots, S_n) \Rightarrow^* (w_1, w_2, \dots, w_n), w_i \in T_i^* \text{ for all } i = 1, \dots, n\}.$$

The language generated by Γ in the union mode, $L_{union}(\Gamma)$, is defined as

$$L_{union}(\Gamma) = \{w : (w_1, w_2, \dots, w_n) \in n\text{-}L(\Gamma), w \in \{w_i : i = 1, \dots, n\}\}.$$

The language generated by Γ in the concatenation mode, $L_{conc}(\Gamma)$, is defined as

$$L_{conc}(\Gamma) = \{w_1w_2\dots w_n : (w_1, w_2, \dots, w_n) \in n\text{-}L(\Gamma)\}$$

The language generated by Γ in the first mode, $L_{first}(\Gamma)$, is defined as

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

Example: $\Gamma = (G_1, G_2, Q)$, where $G_1 = (\{S_1, A_1\}, \{a, b, c\}, \{S_1 \rightarrow aS_1, S_1 \rightarrow aA_1, A_1 \rightarrow bA_1c, A_1 \rightarrow bc\}, S_1)$, $G_2 = (\{S_2, A_2\}, \{d\}, \{S_2 \rightarrow S_2A_2, S_2 \rightarrow A_2, A_2 \rightarrow d\}, S_2)$, $Q = \{(S_1, A_1), (S_2, A_2)\}$ is a 2-multigenerative nonterminal-synchronized grammar system. Notice that $2\text{-}L(\Gamma) = \{(a^n b^n c^n, d^n) : n \geq 1\}$, $L_{union}(\Gamma) = \{a^n b^n c^n : n \geq 1\} \cup \{d^n : n \geq 1\}$, $L_{conc}(\Gamma) = \{a^n b^n c^n d^n : n \geq 1\}$, and $L_{first}(\Gamma) = \{a^n b^n c^n : n \geq 1\}$.

Definition 2.2: An n -multigenerative rule-synchronized grammar system (n -MGR) is $n+1$ tuple

$$\Gamma = (G_1, G_2, \dots, G_n, Q),$$

where $G_i = (N_i, T_i, P_i, S_i)$ is a context-free grammar for each $i = 1, \dots, n$, and Q is a finite set of n -tuples of the form (p_1, p_2, \dots, p_n) , where $p_i \in P_i$ for all $i = 1, \dots, n$. A sentential n -form for n -MGR is defined as the sentential n -form for an n -MGN. Let $\Gamma = (G_1, G_2, \dots, G_n, Q)$ be an n -MGR. Let $\chi = (u_1A_1v_1, u_2A_2v_2, \dots, u_nA_nv_n)$ and $\bar{\chi} = (u_1x_1v_1, u_2x_2v_2, \dots, u_nx_nv_n)$ are two sentential n -form, where $A_i \in N_i$, $u_i \in T_i^*$, and $v_i, x_i \in (N_i \cup T_i)^*$ for all $i = 1, \dots, n$. Let $p_i: A_i \rightarrow x_i \in P_i$ for all $i = 1, \dots, n$ and $(p_1, p_2, \dots, p_n) \in Q$. Then χ directly derives $\bar{\chi}$ in Γ , denoted by $\chi \Rightarrow \bar{\chi}$. An n -language for any n -MGR is defined as the n -language for any n -MGN, and a language generated by n -MGN in the X mode, for each $X \in \{union, conc, first\}$, is defined as the language generated by n -MGR in the X mode.

Example: $\Gamma = (G_1, G_2, Q)$, where $G_1 = (\{S_1, A_1\}, \{a, b, c\}, \{\mathbf{1}: S_1 \rightarrow aS_1, \mathbf{2}: S_1 \rightarrow aA_1, \mathbf{3}: A_1 \rightarrow bA_1c, \mathbf{4}: A_1 \rightarrow bc\}, S_1)$, $G_2 = (\{S_2\}, \{d\}, \{\mathbf{1}: S_2 \rightarrow S_2S_2, \mathbf{2}: S_2 \rightarrow S_2, \mathbf{3}: S_2 \rightarrow d\}, S_2)$, $Q = \{(\mathbf{1}, \mathbf{1}), (\mathbf{2}, \mathbf{2}), (\mathbf{3}, \mathbf{3}), (\mathbf{4}, \mathbf{3})\}$, is 2-multigenerative rule-synchronized grammar system. Notice that $2\text{-}L(\Gamma) = \{(a^n b^n c^n, d^n) : n \geq 1\}$, $L_{union}(\Gamma) = \{a^n b^n c^n : n \geq 1\} \cup \{d^n : n \geq 1\}$, $L_{conc}(\Gamma) = \{a^n b^n c^n d^n : n \geq 1\}$, and $L_{first}(\Gamma) = \{a^n b^n c^n : n \geq 1\}$.

3 RESULTS

Algorithm 3.1: Conversion of n-MGN to n-MGR

- **Input:** n-MGN $\Gamma = (G_1, G_2, \dots, G_n, Q)$
- **Output:** n-MGR $\bar{\Gamma} = (G_1, G_2, \dots, G_n, \bar{Q})$; $n-L(\Gamma) = n-L(\bar{\Gamma})$
- **Method:**

Let $G_i = (N_i, T_i, P_i, S_i)$ for all $i = 1, \dots, n$, then:

$$\bar{Q} := \{(A_1 \rightarrow x_1, A_2 \rightarrow x_2, \dots, A_n \rightarrow x_n) : A_i \rightarrow x_i \in P_i \text{ for all } i = 1, \dots, n, \text{ and} \\ (A_1, A_2, \dots, A_n) \in Q\}$$

Algorithm 3.2: Conversion of n-MGR to n-MGN

- **Input:** n-MGR $\Gamma = (G_1, G_2, \dots, G_n, Q)$
- **Output:** n-MGN $\bar{\Gamma} = (\bar{G}_1, \bar{G}_2, \dots, \bar{G}_n, \bar{Q})$; $n-L(\Gamma) = n-L(\bar{\Gamma})$
- **Method:**

Let $G_i = (N_i, T_i, P_i, S_i)$ for all $i = 1, \dots, n$, then:

$\bar{G}_i = (\bar{N}_i, T_i, \bar{P}_i, S_i)$ for all $i = 1, \dots, n$, where:

$$\bar{N}_i := \{\langle A, x \rangle : A \rightarrow x \in P_i\} \cup \{S_i\},$$

$$\bar{P}_i := \{\langle A, x \rangle \rightarrow y : A \rightarrow x \in P_i, y \in \tau_i(x)\} \cup \{S_i \rightarrow y : y \in \tau_i(S_i)\},$$

where τ_i is a substitution from $N_i \cup T_i$ to $\bar{N}_i \cup T_i$ defined as:

$$\tau_i(a) = \{a\} \text{ for all } a \in T_i; \tau_i(A) = \{\langle A, x \rangle : A \rightarrow x \in P_i\} \text{ for all } A \in N_i.$$

$$\bar{Q} := \{(\langle A_1, x_1 \rangle, \langle A_2, x_2 \rangle, \dots, \langle A_n, x_n \rangle) : (A_1 \rightarrow x_1, A_2 \rightarrow x_2, \dots, A_n \rightarrow x_n) \in Q\} \cup \\ \{(S_1, S_2, \dots, S_n)\}$$

Claim 3.3: Let Γ be any n-MGN, let $\bar{\Gamma}$ be any n-MGR and let $n-L(\Gamma) = n-L(\bar{\Gamma})$. Then, $L_X(\Gamma) = L_X(\bar{\Gamma})$, for each $X \in \{\text{union}, \text{conc}, \text{first}\}$.

Proof:

I. We prove that $L_{\text{union}}(\Gamma) = L_{\text{union}}(\bar{\Gamma})$: $L_{\text{union}}(\Gamma) = \{w : (w_1, w_2, \dots, w_n) \in n-L(\Gamma), w \in \{w_i : i = 1, \dots, n\}\} = \{w : (w_1, w_2, \dots, w_n) \in n-L(\bar{\Gamma}), w \in \{w_i : i = 1, \dots, n\}\} = L_{\text{union}}(\bar{\Gamma})$.

II. We prove that $L_{\text{conc}}(\Gamma) = L_{\text{conc}}(\bar{\Gamma})$: $L_{\text{conc}}(\Gamma) = \{w_1 w_2 \dots w_n : (w_1, w_2, \dots, w_n) \in n-L(\Gamma)\} = \{w_1 w_2 \dots w_n : (w_1, w_2, \dots, w_n) \in n-L(\bar{\Gamma})\} = L_{\text{conc}}(\bar{\Gamma})$.

III. We prove that $L_{\text{first}}(\Gamma) = L_{\text{first}}(\bar{\Gamma})$: $L_{\text{first}}(\Gamma) = \{w_1 : (w_1, w_2, \dots, w_n) \in n-L(\Gamma)\} = \{w_1 : (w_1, w_2, \dots, w_n) \in n-L(\bar{\Gamma})\} = L_{\text{first}}(\bar{\Gamma})$.

Corollary 3.4: The class of languages generated by n-MGN in the X mode, where $X \in \{\text{union}, \text{conc}, \text{first}\}$ is equivalent with the class of language generated by n-MGR in the X mode.

Proof: This corollary follows from Algorithm 3.1, Algorithm 3.2 and Claim 3.3.

Theorem 3.5: For every recursive enumerable language L over an alphabet T there exist a 2-MGR, $\Gamma = ((\bar{N}_1, T, \bar{P}_1, S_1), (\bar{N}_2, T, \bar{P}_2, S_2), Q)$ such that:

- 1) $L = \{w: (S_1, S_2) \Rightarrow^* (w, w)\}$,
- 2) $\{w_1w_2: (S_1, S_2) \Rightarrow^* (w_1, w_2), w_1, w_2 \in T^*, w_1 \neq w_2\} = \emptyset$.

Theorem 3.6: For every recursive enumerable language L over an alphabet T there exist a 2-MGR, $\Gamma = (G_1, G_2, Q)$ such that: $L_{\text{union}}(\Gamma) = L$.

Proof: By the Theorem 3.5 holds: For every recursive enumerable language L over an alphabet T there exist a 2-MGR, $\bar{\Gamma} = ((N_1, T, P_1, S_1), (N_2, T, P_2, S_2), Q)$ such that $L = \{w: (S_1, S_2) \Rightarrow^* (w, w)\}$ and $\{w_1w_2: (S_1, S_2) \Rightarrow^* (w_1, w_2), w_1, w_2 \in T^*, w_1 \neq w_2\} = \emptyset$.

Let $\Gamma = \bar{\Gamma}$. Then, $L_{\text{union}}(\Gamma) = \{w: (S_1, S_2) \Rightarrow^* (w_1, w_2), w_i \in T^* \text{ for } i = 1, 2, w \in \{w_i: i = 1, 2\}\} = \{w: (S_1, S_2) \Rightarrow^* (w, w), w \in T^*\} \cup \{w: (S_1, S_2) \Rightarrow^* (w_1, w_2), w_i \in T^* \text{ for } i = 1, 2, w \in \{w_i: i = 1, 2\}, w_1 \neq w_2\} = \{w: (S_1, S_2) \Rightarrow^* (w, w), w \in T^*\} \cup \emptyset = \{w: (S_1, S_2) \Rightarrow^* (w, w), w \in T^*\} = L$

Theorem 3.7: For every recursive enumerable language L over an alphabet T there exist a 2-MGR, $\Gamma = (G_1, G_2, Q)$ such that: $L_{\text{first}}(\Gamma) = L$.

Proof: By the Theorem 3.5 holds: For every recursive enumerable language L over an alphabet T there exist a 2-MGR, $\bar{\Gamma} = ((N_1, T, P_1, S_1), (N_2, T, P_2, S_2), Q)$ such that $L = \{w: (S_1, S_2) \Rightarrow^* (w, w)\}$ and $\{w_1w_2: (S_1, S_2) \Rightarrow^* (w_1, w_2), w_1, w_2 \in T^*, w_1 \neq w_2\} = \emptyset$.

Let $\Gamma = \bar{\Gamma}$. Then, $L_{\text{first}}(\Gamma) = \{w_1: (S_1, S_2) \Rightarrow^* (w_1, w_2), w_i \in T^* \text{ for } i = 1, 2\} = \{w: (S_1, S_2) \Rightarrow^* (w, w), w \in T^*\} \cup \{w_1: (S_1, S_2) \Rightarrow^* (w_1, w_2), w_i \in T^* \text{ for } i = 1, 2, w_1 \neq w_2\} = \{w: (S_1, S_2) \Rightarrow^* (w, w), w \in T^*\} \cup \emptyset = \{w: (S_1, S_2) \Rightarrow^* (w, w), w \in T^*\} = L$

Theorem 3.8: For every recursive enumerable language L over an alphabet T there exist a 2-MGR, $\Gamma = (G_1, G_2, Q)$ such that: $L_{\text{conc}}(\Gamma) = L$.

Proof: By the Theorem 3.5 holds: For every recursive enumerable language L over an alphabet T there exist a 2-MGR, $\bar{\Gamma} = ((N_1, T, P_1, S_1), (N_2, T, P_2, S_2), Q)$ such that: $L = \{w: (S_1, S_2) \Rightarrow^* (w, w)\}$ and $\{w_1w_2: (S_1, S_2) \Rightarrow^* (w_1, w_2), w_1, w_2 \in T^*, w_1 \neq w_2\} = \emptyset$.

Let $G_1 = (N_1, T, P_1, S_1)$, $G_2 = (N_2, \emptyset, \bar{P}_2, S_2)$, where $\bar{P}_2 = \{A \rightarrow g(x): A \rightarrow x \in P_2\}$, where g is a homomorphism from $(N_2 \cup T)$ to N_2 defined as: For all $X \in N_2$: $g(X) = X$, for all $X \in T$: $g(X) = \varepsilon$. We prove that $L_{\text{conc}}(\Gamma) = L$.

I. We prove that $L \subseteq L_{\text{conc}}(\Gamma)$: Let $w \in L$. Then, there exists a sequence of derivation in $\bar{\Gamma}$ $(S_1, S_2) \Rightarrow^* (w, w)$, thus, there exist a sequence of derivations in Γ $(S_1, S_2) \Rightarrow^* (w, g(w))$. Because $g(a) = \varepsilon$ for all $a \in T$, then $g(w) = \varepsilon$ for all $w \in T^*$. Thus, there exists a sequence of derivations $(S_1, S_2) \Rightarrow^* (w, \varepsilon)$ in Γ . Hence, $w\varepsilon = w \in L_{\text{conc}}(\Gamma)$.

II. We prove that $L_{conc}(\Gamma) \subseteq L$: Let $w \in L_{conc}(\Gamma)$. Then, there exist a sequence of derivations $(S_1, S_2) \Rightarrow^* (w, \varepsilon)$ in Γ , because G_2 derives only empty string (or nothing). $g(x) = \varepsilon$ for all $x \in T^*$, so there exists a sequence of derivation in $\bar{\Gamma}$ of the form: $(S_1, S_2) \Rightarrow^* (w, x)$, where x is any string. Theorem 3.5 implies that $x = w$, thus: $(S_1, S_2) \Rightarrow^* (w, w)$. Thus, $w \in L$.

4 CONCLUSION

Let $L(2\text{-MGN}_X)$ and $L(2\text{-MGR}_X)$ denote the language families defined by 2-MGN in the X mode and 2-MGR in the X mode, respectively, where $X \in \{\text{union}, \text{conc}, \text{first}\}$, let $L(\text{RE})$ denote the family of recursive enumerable languages. From the previous results, we obtain $L(\text{RE}) = L(\text{MGN}_X) = L(\text{MGR}_X)$.

REFERENCES

- [1] Csuhaj-Varju, E.: Cooperating grammar systems. Power and Parameters, LNCS 812, Springer, Berlin, 67-84, 1994.
- [2] Csuhaj-Varju, E., Kelemen, J.: On the Power of Cooperation: a regular Representation of R. E. Languages, Theor. Computer Sci. 81, 305-310, 1991.
- [3] Csuhaj-Varju, E., Dassow, J., Kelemen, J., Paun, Gh.: Grammar Systems: A Grammatical Approach to Distribution and Cooperation, Gordon and Breach, London, 1994.
- [4] Csuhaj-Varju, E., Salomaa, A.: Networks of language processors: parallel communicating systems. EATCS Bulletin 66, 122-138, 1997.
- [5] Dassow, J., Paun, Gh., Rozenberg, G.: Grammar Systems. In Handbook of Formal Languages, Rozenberg, G. and Salomaa, A. (eds.), Volumes 2, Springer, Berlin 1997
- [6] Kleijn, H. C. M., Rozenberg, G.: On the Generative Power of Regular Pattern Grammars, Acta Informatica 20, 391-411, 1983.
- [7] Meduna, A.: Automata and Languages: Theory and Applications, Springer, London, 2000.
- [8] Meduna, A.: Simultaneously One-Turn Two-Pushdown Automata, International Journal of Computer Mathematics 80, 679-687, 2003.
- [9] Paun, Gh., Salomaa, A., S. Vicolov, S.: On the generative capacity of parallel communicating grammar systems. International Journal of Computer Mathematics 45, 45-59, 1992.
- [10] Paun, Gh., Santean, L.: Parallel communicating grammar systems: the regular case, Ann. Univ. Buc., Ser. Matem.-Inform. 38, 55-63, 1989.
- [11] Paun, Gh., Santean, L.: Further Remarks about Parallel Communicating Grammar Systems, International Journal of Computer Mathematics 34, 187-203, 1990.
- [12] Salomaa, A.: Formal Languages, Academic Press, New York, 1973.
- [13] Santean, L.: Parallel Communicating Systems, EATCS Bulletin, 160-171, 1990.
- [14] Vaszil, G.: On simulating Non-returning PC grammar systems with returning systems, Theoretical Computer Science (209) 1-2, 319-329, 1998.