Multigenerative Grammar Systems

Alexander Meduna*

meduna@fit.vutbr.cz

Roman Lukáš*

lukas@fit.vutbr.cz

Abstract: This paper presents new models for all recursive enumerable languages. These models are based on a 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.

Key Words: Grammar systems, Leftmost derivation, Recursive enumerable language.

1 Introduction

The formal language theory has recently intensively investigated various grammar systems (see [1], [2], [8]), which consist of several cooperating components, usually represented by grammars. Although this variaty 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-multigenerative 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.

In this paper, we prove that all the multigenerative grammar systems under discussion characterize the family of recursively enumerable languages. Besides this fundamental result, we give several transformation algorithms of these multigenerative grammar systems.

2 Preliminaries

This paper assumes that the reader is familiar with the formal language theory (see [4]). For a set, Q, *card*(Q) denotes the cardinality of Q. For an alphabet, V, V^* represents the free monoid generated by V under the operation of concatenation. The unit of V^* is denoted by ε. Set $V^+ = V^* - \{\varepsilon\}$; algebraically, V^+ is thus the free semigroup generated by V under the operation of concatenation. For every $w \in V^*$, $|w|$ denotes the length of w, **sym**(w, i) denotes the i-th symbol in w. A *context-free grammar* is a quadruple, $G = (N, T, P, S)$, where N and T are two disjoint alphabets. Symbols in N and T are referred to as *nonterminals* and *terminals*,

* Department of Information Systems, Faculty of Information Technology, Brno University of Technology, Božetěchova 2, Brno 612 66, Czech Republic

respectively, and $S \in N$ is the *start symbol* of G. P is a finite set of *rules* of the form $A \rightarrow x$, where $A \in N$ and $x \in (N \cup T)^*$. To declare that a label r denotes the rule, this is written as r: $A \rightarrow x$. The nonterminal A is the left-hand side of r, denoted by **lhs**(r). The string x is the right hand side of r, denoted by **rhs**(r). Let $u, v \in (N \cup T)^*$. For every $A \rightarrow x \in P$, write $uAv \Rightarrow uxv$. Let $\Rightarrow^*$ denote the transitive-reflexive closure of $\Rightarrow$. The *language of* G, $L(G)$, is defined as $L(G) = \{w: S \Rightarrow^* w \text{ in } G, \text{ for some } w \in T^*\}$.

3 n-multigenerative Nonterminal-synchronized Grammar System

3.1 Basic Definition

An *n-multigenerative nonterminal-synchronized grammar system* (MGN) is n+1 tuple

$$\Gamma = (G_1, G_2, \ldots, G_n, Q), \text{ where:}$$

- $G_i = (N_i, T_i, P_i, S_i)$ is a context-free grammar for each $i = 1, \ldots, n$,
- Q is a finite set of n-tuples of the form $(A_1, A_2, \ldots, A_n)$, where $A_i \in N_i$ for all $i = 1, \ldots, n$.

3.2 Sentential n-form

Let $\Gamma = (G_1, G_2, \ldots, G_n, Q)$ be a MGN. Then a *sentential n-form of MGN* is an n-tuple of the form $\chi = (x_1, x_2, \ldots, x_n)$, where $x_i \in (N_i \cup T_i)^*$ for all $i = 1, \ldots, n$.

3.3 Direct Derivation Step

Let $\Gamma = (G_1, G_2, \ldots, G_n, Q)$ be a MGN. Let $\chi = (u_1A_1v_1, u_2A_2v_2, \ldots, u_nA_nv_n)$ and $\overline{\chi} = (u_1x_1v_1, u_2x_2v_2, \ldots, u_nx_nv_n)$ 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 $i = 1, \ldots, n$. Let $A_i \rightarrow x_i \in P_i$ for all $i = 1, \ldots, n$ and $(A_1, A_2, \ldots, A_n) \in Q$. Then χ directly derives $\overline{\chi}$ in Γ, denoted by $\chi \Rightarrow \overline{\chi}$.

3.4 Sequence of Derivation Steps, Part 1

Let $\Gamma = (G_1, G_2, \ldots, G_n, Q)$ be a MGN.

- Let χ by any sentential n-form of Γ. Γ makes a *zero-step* derivation from χ to χ, which is written as $\chi \Rightarrow^0 \chi$.
- Let there exists a sequence of sentential n-forms $\chi_0, \chi_1, \ldots, \chi_k$ for some $k \geq 1$ such that $\chi_{i-1} \Rightarrow \chi_i$ for all $i = 1, \ldots, k$. Then, Γ makes *k-step derivation* from χ_0 to χ_k, which is written as $\chi_0 \Rightarrow^n \chi_k$.

3.5 Sequence of Derivation Steps, Part 2

Let $\Gamma = (G_1, G_2, \ldots, G_n, Q)$ be a MGN, let χ and $\overline{\chi}$ be two sentential n-forms of Γ.

- If there exists $k \geq 1$ so $\chi \Rightarrow^k \overline{\chi}$ in Γ, then $\chi \Rightarrow^+ \overline{\chi}$,
- If there exists $k \geq 0$ so $\chi \Rightarrow^k \overline{\chi}$ in Γ, then $\chi \Rightarrow^* \overline{\chi}$.

3.6 n-language

Let $\Gamma = (G_1, G_2, \ldots, G_n, Q)$ be a MGN. The n-language of Γ, n-$L(\Gamma)$, is defined as:

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

3.7 Three Types of Generated Languages

- 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:
$$L_{conc}(\Gamma) = \{w_1 w_2 \dots w_n: (w_1, w_2, \dots, w_n) \in n\text{-}L(\Gamma)\}$$
- The *language generated by Γ in the leftmost mode*, $L_{first}(\Gamma)$, is defined as:
$$L_{first}(\Gamma) = \{w_1: (w_1, w_2, \dots, w_n) \in n\text{-}L(\Gamma)\}$$

3.8 Example

$\Gamma = (G_1, G_2, Q)$, where:

- $G_1 = (\{S_1, A_1\}, \{a, b, c\}, \{S_1 \to aS_1, S_1 \to aA_1, A_1 \to bA_1c, A_1 \to bc\}, S_1)$,
- $G_2 = (\{S_2, A_2\}, \{d\}, \{S_2 \to S_2A_2, S_2 \to A_2, A_2 \to d\}, S_2)$,
- $Q = \{(S_1, S_2), (A_1, A_2)\}$

  is a 2-multigenerative nonterminal-synchronized grammar system.

Notice that this system generates following languages in the different modes:

- $L_{union}(\Gamma) = \{a^n b^n c^n: n \geq 1\} \cup \{d^n: n \geq 1\}$,
- $L_{conc}(\Gamma) = \{a^n b^n c^n d^n: n \geq 1\}$,
- $L_{first}(\Gamma) = \{a^n b^n c^n: n \geq 1\}$.

4 n-multigenerative Rule-synchronized Grammar System

4.1 Basic Definition

An *n-multigenerative rule-synchronized grammar system* (MGR) is n+1 tuple
$$\Gamma = (G_1, G_2, \dots, G_n, Q), \text{ where:}$$

- $G_i = (N_i, T_i, P_i, S_i)$ is a context-free grammar for each $i = 1, \dots, n$,
- 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 $i = 1, \dots, n$.

4.2 Sentential n-form

A sentential n-form for MGR is defined analogically as the sentential n-form for a MGN.

4.3 Direct Derivation Step

Let $\Gamma = (G_1, G_2, \dots, G_n, Q)$ be a 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: $A_i \to x_i \in P_i$ for all 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 $\chi \Rightarrow \bar{\chi}$.

4.4 Sequence of Derivation Steps

A sequence of derivation steps for MGR is defined analogically as the sequence of derivation steps for a MGN.

4.5 n-language

An n-language for MGR is defined analogically as the n-language for a MGN.

4.6 Three Types of Generated Languages

A language generated by MGN in the X mode, for each $X \in \{union, conc, first\}$, is defined analogically as the language generated by MGR in the X mode.

4.7 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 this system generates following languages in the different modes:

- $L_{union}(\Gamma) = \{a^nb^nc^n: n \geq 1\} \cup \{d^n: n \geq 1\}$,
- $L_{conc}(\Gamma) = \{a^nb^nc^nd^n: n \geq 1\}$,
- $L_{first}(\Gamma) = \{a^nb^nc^n: n \geq 1\}$.

5 Conversions Between MGN and MGR

5.1 Algorithm 1: Conversion From MGN to MGR

INPUT: MGN $\Gamma = (G_1, G_2, \ldots, G_n, Q)$

OUTPUT: MGR $\overline{\Gamma} = (G_1, G_2, \ldots, G_n, \overline{Q})$; $L_X(\Gamma) = L_X(\overline{\Gamma})$, for each $X \in \{union, conc, first\}$

METHOD:

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

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

5.2 Algorithm 2: Conversion From MGR to MGN

INPUT: MGR $\Gamma = (G_1, G_2, \ldots, G_n, Q)$

OUTPUT: MGN $\overline{\Gamma} = (\overline{G}_1, \overline{G}_2, \ldots, \overline{G}_n, \overline{Q})$; $L_X(\Gamma) = L_X(\overline{\Gamma})$, where $X \in \{union, conc, first\}$

METHOD:

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

- $\overline{G}_i = (\overline{N}_i, T_i, \overline{P}_i, S_i)$ for all $i = 1, \ldots, n$, where:
    - $\overline{N}_i := \{<A, x>: A \to x \in P_i\} \cup \{S_i\}$,
    - $\overline{P}_i := \{<A, x> \to y: A \to x \in P_i, y \in \tau_i(x)\} \cup \{S_i \to y: y \in \tau_i(S_i)\}$,

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

    $\tau_i(a) = \{a\}$ for all $a \in T_i$; $\tau_i(A) = \{<A, x>: A \to x \in P_i\}$ for all $A \in N_i$.
- $\overline{Q} := \{(<A_1, x_1>, <A_2, x_2>, \ldots, <A_n, x_n>): (A_1 \to x_1, A_2 \to x_2, \ldots, A_n \to x_n) \in Q\} \cup$
  $\{(S_1, S_2, \ldots, S_n)\}$

5.3 Corollary

The class of languages generated by MGNs in the X mode, where $X \in \{union, conc, first\}$ is equivalent with the class of language generated by MGRs in the X mode.

Proof: This corollary follows from Algorithm 1 and Algorithm 2.

6 Generative Power of MGN and MGR

6.1 Theorem 1

For every recursive enumerable language L over an alphabet T there exists a MGR,

$$\Gamma = ((\overline{N}_1, T, \overline{P}_1, S_1), (\overline{N}_2, T, \overline{P}_2, S_2), Q), \text{ such that:}$$

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

Proof: Recall that for every recursive enumerable language L over an alphabet T there exist two context-free grammars $G_1 = (N_1, \overline{T}, P_1, S_1)$, $G_2 = (N_2, \overline{T}, P_2, S_2)$ and homomorphism h: from $\overline{T}$ to T^* such that $L = \{h(x) : x \in L(G_1) \cap L(G_2)\}$. (see Theorem 10.3.1 in [3]). Furthermore, for every context-free grammar, there exists an equivalent context-free grammar in Greibach normal form (see Section 5.1.4.2 in [4]). Hence, without lost of generality, we can assume that G_1 and G_2 are in Greibach normal form. Consider an 2-MGR $\Gamma = (G_1, G_2, Q)$, where:

$G_1 = (\overline{N}_1, T, \overline{P}_1, S_1)$, where $\overline{N}_1 = N_1 \cup \{\overline{a} : a \in \overline{T}\}$, $\overline{P}_1 = \{A \to \overline{a}x : A \to ax \in P_1, a \in \overline{T}, x \in N_1^*\} \cup \{\overline{a} \to h(a): a \in \overline{T}\}$

$G_2 = (\overline{N}_2, T, \overline{P}_2, S_2)$, where $\overline{N}_2 = N_2 \cup \{\overline{a} : a \in \overline{T}\}$, $\overline{P}_2 = \{A \to \overline{a}x : A \to ax \in P_2, a \in \overline{T}, x \in N_2^*\} \cup \{\overline{a} \to h(a): a \in \overline{T}\}$

$Q = \{(A_1 \to \bar{a}x_1, A_2 \to \bar{a}x_2): A_1 \to \bar{a}x_1 \in \bar{P}_1,\ A_2 \to \bar{a}x_2 \in \bar{P}_2, a \in \bar{T}\} \cup \{(\bar{a} \to h(a), \bar{a} \to h(a)): a \in \bar{T}\}$

Now, we prove that for a 2-MGR $\Gamma = (G_1, G_2, Q)$ holds:

1) $\{w: (S_1, S_2) \Rightarrow^* (w, w)\} = L$.

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

Claim A: $\{w: (S_1, S_2) \Rightarrow^* (w, w)\} = L$:

Proof: I. We prove that $L \subseteq \{w: (S_1, S_2) \Rightarrow^* (w, w)\}$:

Let $w \in L$ be any string. Then, there exists a string $a_1a_2...a_n \in \bar{T}^*$ such that:

1) $a_1a_2...a_n \in L(G_1)$
2) $a_1a_2...a_n \in L(G_2)$
3) $h(a_1a_2...a_n) = w$

It means that there exist following derivations in G_1 and G_2:

1. $S_1 \Rightarrow a_1x_1\ [p_1] \Rightarrow a_1a_2x_2\ [p_2] \Rightarrow \ldots \Rightarrow a_1a_2...a_n\ [p_n]$,
2. $S_2 \Rightarrow a_1y_1\ [r_1] \Rightarrow a_1a_2y_2\ [r_2] \Rightarrow \ldots \Rightarrow a_1a_2...a_n\ [r_n]$,

where $a_i \in \bar{T}$, $x_i \in N_1^*$, $y_i \in N_2^*$, $p_i \in P_1$, $r_i \in P_2$ for all $i = 1, \ldots, n$.

Observe that for all $i = 1, \ldots, n$ holds: **sym**(**rhs**(p_i), 1) = **sym**(**rhs**(r_i), 1) = a_i.

A construction of a set Q, $Q = \{(A_1 \to \bar{a}x_1, A_2 \to \bar{a}x_2): A_1 \to \bar{a}x_1 \in \bar{P}_1,\ A_2 \to \bar{a}x_2 \in \bar{P}_2, a \in \bar{T}\} \cup \{(\bar{a} \to h(a), \bar{a} \to h(a)): a \in \bar{T}\}$, implies:

1) Let p_i: $A_i \to a_iu_i \in \bar{P}_1$, r_i: $B_i \to a_iv_i \in \bar{P}_2$. Then $(A_i \to \bar{a}_iu_i, B_i \to \bar{a}_iv_i) \in Q$ for all $i = 1,..., n$

2) $(\bar{a}_i \to h(a_i), \bar{a}_i \to h(a_i)) \in Q$ for all $i =1, \ldots, n$.

Then, there exists a sequence of derivation in 2-MGR Γ:

$(S_1, S_2) \Rightarrow (\bar{a}_1x_1, \bar{a}_1y_1) \Rightarrow (h(a_1)x_1, h(a_1)y_1)$

$\Rightarrow (h(a_1)\bar{a}_2x_2, h(a_1)\bar{a}_2y_2) \Rightarrow (h(a_1)h(a_2)x_2, h(a_1)h(a_2)y_2)$

$\Rightarrow \ldots$

$\Rightarrow (h(a_1)h(a_2)\ldots h(a_n), h(a_1)h(a_2)\ldots h(a_n)) =$

$(h(a_1a_2\ldots a_n), h(a_1a_2\ldots a_n)) = (w, w)$

It means that $(S_1, S_2) \Rightarrow^* (w, w)$ in Γ.

II. We prove, that $\{w: (S_1, S_2) \Rightarrow^* (w, w)\} \subseteq L$:

Let $(S_1, S_2) \Rightarrow^* (w, w)$ in Γ. Then, there exists a sequence of derivation in 2-MGR Γ:

$(S_1, S_2) \Rightarrow (\bar{a}_1x_1, \bar{a}_1y_1) \Rightarrow (h(a_1)x_1, h(a_1)y_1)$

$\Rightarrow (h(a_1)\bar{a}_2x_2, h(a_1)\bar{a}_2y_2) \Rightarrow (h(a_1)h(a_2)x_2, h(a_1)h(a_2)y_2)$

$\Rightarrow \ldots$

$\Rightarrow (h(a_1)h(a_2)\ldots h(a_n), h(a_1)h(a_2)\ldots h(a_n)) =$

$(h(a_1a_2\ldots a_n), h(a_1a_2\ldots a_n)) = (w, w)$

Analogically as in part I., we can prove that there exist derivations in G_1 and G_2 of the forms:

1. $S_1 \Rightarrow a_1x_1\ [p_1] \Rightarrow a_1a_2x_2\ [p_2] \Rightarrow \ldots \Rightarrow a_1a_2\ldots a_n\ [p_n]$,
2. $S_2 \Rightarrow a_1y_1\ [r_1] \Rightarrow a_1a_2y_2\ [r_2] \Rightarrow \ldots \Rightarrow a_1a_2\ldots a_n\ [r_n]$.

It means that $a_1a_2\ldots a_n \in L(G_1)$, $a_1a_2\ldots a_n \in L(G_2)$ and $h(a_1a_2\ldots a_n) = w$, and so $w \in L$.

Claim B: $\{w_1w_2: (S_1, S_2) \Rightarrow^* (w_1, w_2), w_1, w_2 \in T^*, w_1 \neq w_2\} = \emptyset$
Proof (by contradiction): Let $\{w_1w_2: (S_1, S_2) \Rightarrow^* (w_1, w_2), w_1, w_2 \in T^*, w_1 \neq w_2\} \neq \emptyset$. Then, there exist two different strings $w_1 = h(a_1)h(a_2)\ldots h(a_n)$, $w_2 = h(b_1)h(b_2)\ldots h(b_n)$, such that: $(S_1, S_2) \Rightarrow^* (w_1, w_2)$.

I. Assume that for all $i = 1, \ldots, n$ holds: $a_i = b_i$: Then $w_1 = h(a_1)h(a_2)\ldots h(a_n) = h(b_1)h(b_2)\ldots h(b_n) = w_2$. But w_1 and w_2 are different string. That is a contradiction.

II. Assume that there exists $k \leq n$, such that $a_k \neq b_k$: Then, there exists a sequence of derivations in Γ of the form:

$$(S_1, S_2) \Rightarrow (\bar{a}_1x_1, \bar{a}_1y_1) \Rightarrow (h(a_1)x_1, h(a_1)y_1)$$
$$\Rightarrow (h(a_1)\bar{a}_2x_2, h(a_1)\bar{a}_2y_2) \Rightarrow (h(a_1)h(a_2)x_2, h(a_1)h(a_2)y_2)$$
$$\Rightarrow \ldots$$
$$\Rightarrow (h(a_1)h(a_2)\ldots h(a_{k-1})x_{k-1}, h(a_1)h(a_2)\ldots h(a_{k-1})y_{k-1})$$

Then, there must exists a derivation $(x_{k-1}, y_{k-1}) \Rightarrow (\bar{a}_kx_k, \bar{b}_ky_k)$, where $\bar{a}_k \neq \bar{b}_k$. Because $Q = \{(A_1 \rightarrow \bar{a}x_1, A_2 \rightarrow \bar{a}x_2): A_1 \rightarrow \bar{a}x_1 \in \bar{P}_1, A_2 \rightarrow \bar{a}x_2 \in \bar{P}_2, a \in \bar{T}\} \cup \{(\bar{a} \rightarrow h(a), \bar{a} \rightarrow h(a)): a \in \bar{T}\}$, there must by used a pair of rules (p, r) such that **sym**(**rhs**(p), 1) = **sym**(**rhs**(r), 1). Next derivation must be of the form: $(x_{k-1}, y_{k-1}) \Rightarrow (\bar{a}_kx_k, \bar{b}_ky_k)$, where $\bar{a}_k = \bar{b}_k$. That is a contradiction.

6.2 Theorem 2:

For every recursive enumerable language L over an alphabet T there exists a MGR,

$$\Gamma = (G_1, G_2, Q), \text{ such that: } L_{union}(\Gamma) = L.$$

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

6.3 Theorem 3:

For every recursive enumerable language L over an alphabet T there exists a MGR,

$$\Gamma = (G_1, G_2, Q), \text{ such that: } L_{first}(\Gamma) = L.$$

Proof: By the Theorem 1 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 $\Gamma = \overline{\Gamma}$. Then, $L_{first}(\Gamma) = \{w_1: (S_1, S_2) \Rightarrow^* (w_1, w_2), w_i \in T^* \text{ for } i = 1, 2\} = \{w: (S_1, S_2) \Rightarrow^* (w, w), w \in T^*\} \cup \{w_1: (S_1, S_2) \Rightarrow^* (w_1, w_2), w_i \in T^* \text{ for } i = 1, 2, w_1 \neq w_2\} = \{w: (S_1, S_2) \Rightarrow^* (w, w), w \in T^*\} \cup \varnothing = \{w: (S_1, S_2) \Rightarrow^* (w, w), w \in T^*\} = L$

6.4 Theorem 4:

For every recursive enumerable language L over an alphabet T there exists a MGR,

$$\Gamma = (G_1, G_2, Q), \text{ such that: } L_{conc}(\Gamma) = L.$$

Proof: By the Theorem 1 holds: For every recursive enumerable language L over an alphabet T there exist a 2-MGR, $\overline{\Gamma} = ((N_1, T, P_1, S_1), (N_2, T, P_2, S_2), Q)$ such that: $L = \{w: (S_1, S_2) \Rightarrow^* (w, w)\}$ and $\{w_1w_2: (S_1, S_2) \Rightarrow^* (w_1, w_2), w_1, w_2 \in T^*, w_1 \neq w_2\} = \varnothing$.

Let $G_1 = (N_1, T, P_1, S_1)$, $G_2 = (N_2, \varnothing, \overline{P_2}, S_2)$, where $\overline{P_2} = \{ A \to g(x): A \to x \in P_2\}$, 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 $L_{conc}(\Gamma) = L$.

I. We prove that $L \subseteq L_{conc}(\Gamma)$: Let $w \in L$. Then, there exists a sequence of derivation in $\overline{\Gamma}$ of the form: $(S_1, S_2) \Rightarrow^* (w, w)$, thus, there exist a sequence of derivations in Γ of the form $(S_1, S_2) \Rightarrow^* (w, g(w))$. 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, $w\varepsilon = w \in L_{conc}(\Gamma)$.

II. We prove that $L_{conc}(\Gamma) \subseteq L$: Let $w \in L_{conc}(\Gamma)$. Then, there exists a sequence of derivations $(S_1, S_2) \Rightarrow^* (w, \varepsilon)$ in Γ, because G_2 derives only the empty string. $g(x) = \varepsilon$ for all $x \in T^*$, so there exists a sequence of derivation in $\overline{\Gamma}$ of the form: $(S_1, S_2) \Rightarrow^* (w, x)$, where x is any string. Theorem 3 implies that $x = w$, thus: $(S_1, S_2) \Rightarrow^* (w, w)$. Thus, $w \in L$.

7 Conclusion

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

References

1. Csuhaj-Varju, E., Dassow, J., Kelemen, J., Paun, Gh.: Grammar Systems: A Grammatical Approach to Distribution and Cooperation, Gordon and Breach, London, 1994.
2. Dassow, J., Paun, Gh., and Rozenberg, G.: Grammar Systems, In Handbook of Formal Languages, Rozenberg, G. and Salomaa, A. (eds.), Volumes 2, Springer, Berlin, 1997.
3. Harrison, Michael A.: Introduction to Formal Language Theory. Addison-Wesley, London, 1978.
4. Meduna, A.: Automata and Languages: Theory and Applications, Springer, London, 2000.
5. Meduna, A.: Two-Way Metalinear PC Grammar Systems and Their Descriptional Complexity, Acta Cybernetica, 2003.
6. Paun, Gh., Salomaa, A. and S. Vicolov, S.: On the generative capacity of parallel communicating grammar systems. International Journal of Computer Mathematics 45, 45-59, 1992.
7. Salomaa, A.: Formal Languages, Academic Press, New York, 1973.
8. Vaszil, G.: On simulating Non-returning PC grammar systems with returning systems, Theoretical Computer Science (209) 1-2, 319-329, 1998.