General Multigenerative Grammar Systems

Roman Lukáš[1], Alexander Meduna[1]

[1] Dept. of Information Systems, Faculty of Information Technology, Brno University of Technology, Božetěchova 1, 612 66 Brno, Czech Republic
{lukas, meduna}@fit.vutbr.cz

Abstract. This paper presents new models for generating matrix 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 general way. The rewritten nonterminals are determined by a finite set of nonterminal sequences.

Keywords: Grammar system, matrix grammar, general derivation.

1 Introduction

The formal language theory has intensively investigated various grammar systems (see [1], [2], [8]), which consist of several cooperating components, usually represented by grammars. Although this variety is extremely broad, all these grammar systems always use 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 general way—that is, in every derivation step, each of these components rewrites any nonterminal occurring in its current sentential form. These n derivations are controled 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 languages, which is generated by matrix grammars. 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.

A *context-free grammar* is a quadruple, $G = (N, T, P, S)$, where N and T are disjoint alphabets. Symbols in N and T are referred to as *nonterminals* and *terminals*, respectively, and $S \in N$ is the *start symbol* of G. P is a finite set of *rules* of the form $A \to x$, where $A \in N$ and $x \in (N \cup T)^*$. To declare that a label r denotes the rule, we write as r: $A \to x$. Let $u, v \in (N \cup T)^*$. For every r: $A \to x \in P$, write $uAv \Rightarrow uxv$ $[r]$, or simply $uAv \Rightarrow uxv$. Let $\Rightarrow^*$ denote the transitive-reflexive closure of $\Rightarrow$. The *language of* G, $L(G)$, is defined as $L(G) = \{w\colon S \Rightarrow^* w \text{ in } G, \text{ for some } w \in T^*\}$.

A *matrix grammar* is a pair, $H = (G, M)$, where $G = (N, T, P, S)$ is a context-free grammar and M is a finite language over alphabet P, $M \subseteq P^*$. Let $x_0, x_1, \ldots, x_n \in (N \cup T)^*$ for any $n > 0$, $x_{i-1} \Rightarrow x_i$ $[p_i]$ in G for all $i = 1, \ldots, n$ and $p_1p_2\ldots p_n \in M$. Then matrix grammar H makes direct derivation step from x_0 to x_n, denoted as $x_0 \Rightarrow x_n$. Let $\Rightarrow^*$ denote the transitive-reflexive closure of $\Rightarrow$. The *language of* H, $L(H)$, is defined as $L(H) = \{w\colon S \Rightarrow^* w \text{ in } H, \text{ for some } w \in T^*\}$.

3 Definitions

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

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

where $G_i = (N_i, T_i, P_i, S_i)$ is a context-free grammar for each $i = 1, \ldots, n$, and 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 $i = 1, \ldots, n$. Then, a *sentential n-form* of n-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 $i = 1, \ldots, n$. Let $\chi = (u_1A_1v_1, u_2A_2v_2, \ldots, u_nA_nv_n)$ and $\bar{\chi} = (u_1x_1v_1, u_2x_2v_2, \ldots, u_nx_nv_n)$ be two sentential n-form, where $A_i \in N_i$, $u_i, v_i, x_i \in (N_i \cup T_i)^*$ for all $i = 1, \ldots, n$. Let $A_i \to x_i \in P_i$ for all $i = 1, \ldots, n$ and $(A_1, A_2, \ldots, A_n) \in Q$. Then χ directly derives $\bar{\chi}$ 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, $n\text{-}L(\Gamma)$, is defined as

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

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

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

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

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

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

$$L_{first}(\Gamma) = \{w_1\colon (w_1, w_2, \ldots, w_n) \in n\text{-}L(\Gamma)\}.$$

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

Definition 2. An *n-multigenerative rule-synchronized grammar system* (n-MGR) is n+1 tuple

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

where $G_i = (N_i, T_i, P_i, S_i)$ is a context-free grammar for each $i = 1, \ldots, n$, and Q is a finite set of n-tuples of the form $(p_1, p_2, \ldots, p_n)$, where $p_i \in P_i$ for all $i = 1, \ldots, n$. A sentential n-form for n-MGR is defined as the sentential n-form for an n-MGN. Let $\chi = (u_1A_1v_1, u_2A_2v_2, \ldots, u_nA_nv_n)$ and $\bar{\chi} = (u_1x_1v_1, u_2x_2v_2, \ldots, u_nx_nv_n)$ are two sentential n-form, where $A_i \in N_i$, $u_i, v_i, x_i \in (N_i \cup T_i)^*$ for all $i = 1, \ldots, n$. Let p_i: $A_i \rightarrow x_i \in P_i$ for all $i = 1, \ldots, n$ and $(p_1, p_2, \ldots, p_n) \in Q$. 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 2. $\Gamma = (G_1, G_2, Q)$, 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. We have $2\text{-}L(\Gamma) = \{(a^nb^nc^n, d^n): n \geq 1\}$, $L_{union}(\Gamma) = \{a^nb^nc^n: n \geq 1\} \cup \{d^n: n \geq 1\}$, $L_{conc}(\Gamma) = \{a^nb^nc^nd^n: n \geq 1\}$, and $L_{first}(\Gamma) = \{a^nb^nc^n: n \geq 1\}$.

3 Results

Algorithm 1. Conversion of n-MGN to n-MGR

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

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

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

Algorithm 2. Conversion of n-MGR to n-MGN

- ***Input:*** n-MGR $\Gamma = (G_1, G_2, \ldots, G_n, Q)$
- ***Output:*** n-MGN $\overline{\Gamma} = (\overline{G}_1, \overline{G}_2, \ldots, \overline{G}_n, \overline{Q})$ such that $n\text{-}L(\Gamma) = n\text{-}L(\overline{\Gamma})$
- ***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 $A \in N_i$.

  $\overline{Q} = \{(<A_1, x_1>, <A_2, x_2>, \ldots, <A_n, x_n>): (A_1 \to x_1, A_2 \to x_2, \ldots, A_n \to x_n) \in Q\} \cup \{(S_1, S_2, \ldots, S_n)\}$.

Claim 1. Let Γ be any n-MGN, let $\overline{\Gamma}$ be any n-MGR and let $n\text{-}L(\Gamma) = n\text{-}L(\overline{\Gamma})$. Then, $L_X(\Gamma) = L_X(\overline{\Gamma})$, for each $X \in \{union, conc, first\}$.

Proof.

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

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

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

□

Theorem 1. The class of languages generated by n-MGN in the X mode, where $X \in \{union, conc, first\}$ is equivalent to the class of language generated by n-MGR in the X mode.

Proof. This follows from Algorithm 1, Algorithm 2 and Claim 1.

□

Algorithm 3. Conversion of n-MGR in the concatenation mode to matrix grammar

- ***Input:*** n-MGR $\Gamma = (G_1, G_2, \ldots, G_n, Q)$
- ***Output:*** Matrix grammar $H = (G, M)$ such that $L_{conc}(\Gamma) = L(H)$
- ***Method:***

Let $G_i = (N_i, T_i, P_i, S_i)$ for all $i = 1, \ldots, n$, and let for any $j, k = 1, \ldots, n$, where $j \neq k$ holds: $N_j \cap N_k = \varnothing$; $S \notin N_j$. Then:

$G = (N, T, P, S)$, where:

$$N = \{S\} \cup (\bigcup_{i=1}^{n} N_i); \; T = \bigcup_{i=1}^{n} T_i;$$

$$P = \{s\colon S \rightarrow S_1S_2\ldots S_n\} \cup (\bigcup_{i=1}^{n} P_i);$$

$M = \{s\} \cup \{p_1p_2\ldots p_n\colon (p_1, p_2, \ldots, p_n) \in Q\}$.

Algorithm 4. Conversion of n-MGR in the first mode to matrix grammar

- ***Input:*** n-MGR $\Gamma = (G_1, G_2, \ldots, G_n, Q)$
- ***Output:*** Matrix grammar $H = (G, M)$ such that $L_{first}(\Gamma) = L(H)$
- ***Method:***

Let $G_i = (N_i, T_i, P_i, S_i)$ for all $i = 1, \ldots, n$, and let for any $j, k = 1, \ldots, n$, where $j \neq k$ holds: $N_j \cap N_k = \varnothing$; $S \notin N_j$. Then:

$G = (N, T, P, S)$, where:

$$N = \{S\} \cup N_1 \cup (\bigcup_{i=2}^{n} \{\overline{A} : A \in N_i\}); \; T = T_1;$$

$P = \{ s\colon S \rightarrow S_1h(S_2)\ldots h(S_n) \} \cup P_1 \cup$
$(\bigcup_{i=2}^{n} \{h(A) \rightarrow h(x) : A \rightarrow x \in P_i\})$, where h is a homomorphism from
$(\bigcup_{i=2}^{n} N_i) \cup (\bigcup_{i=2}^{n} T_i)$ to $\bigcup_{i=2}^{n} \{\overline{A} : A \in N_i\}$ defined as: $h(a) = \varepsilon$ for all
$a \in \bigcup_{i=2}^{n} T_i$; $h(A) = \overline{A}$ for all $A \in \bigcup_{i=2}^{n} N_i$;

$M = \{s\} \cup \{ p_1\overline{p}_2\ldots\overline{p}_n : (p_1, p_2, \ldots, p_n) \in Q\}$.

Convention: Let $p\colon A \rightarrow x$ be a rule. Then, label $\overline{p}$ denotes rule $h(A) \rightarrow h(x)$.

Algorithm 5. Conversion of n-MGR in the union mode to matrix grammar

- ***Input:*** n-MGR $\Gamma = (G_1, G_2, \ldots, G_n, Q)$
- ***Output:*** Matrix grammar $H = (G, M)$ such that $L_{union}(\Gamma) = L(H)$
- ***Method:***

  Let $G_i = (N_i, T_i, P_i, S_i)$ for all $i = 1, \ldots, n$, and let for any $j, k = 1, \ldots, n$, where $j \neq k$ holds: $N_j \cap N_k = \emptyset$; $S \notin N_j$. Then:

  $G = (N, T, P, S)$, where:

$$N = \{S\} \cup (\bigcup_{i=1}^{n} N_i) \cup (\bigcup_{i=1}^{n} \{\overline{A} : A \in N_i\}); \; T = \bigcup_{i=1}^{n} T_i;$$

$P = \{$ s_1: $S \rightarrow S_1 h(S_2) \ldots h(S_n)$, s_2: $S \rightarrow h(S_1) S_2 \ldots h(S_n)$, ...
s_n: $S \rightarrow h(S_1) h(S_2) \ldots S_n$ $\}$ $\cup$

$(\bigcup_{i=1}^{n} P_i) \cup (\bigcup_{i=1}^{n} \{h(A) \rightarrow h(x) : A \rightarrow x \in P_i\})$, where h is a homomorphism from $(\bigcup_{i=1}^{n} N_i) \cup (\bigcup_{i=1}^{n} T_i)$ to $\bigcup_{i=1}^{n} \{\overline{A} : A \in N_i\}$ defined as: $h(a) = \varepsilon$ for all $a \in \bigcup_{i=1}^{n} T_i$; $h(A) = \overline{A}$ for all $A \in \bigcup_{i=1}^{n} N_i$;

$$M = \{s_1, s_2, \ldots, s_n\} \cup \{ p_1 \overline{p}_2 \ldots \overline{p}_n : (p_1, p_2, \ldots, p_n) \in Q\}$$
$$\cup \{ \overline{p}_1 p_2 \ldots \overline{p}_n : (p_1, p_2, \ldots, p_n) \in Q\}$$
$$\cup \ldots$$
$$\cup \{ \overline{p}_1 \overline{p}_2 \ldots p_n : (p_1, p_2, \ldots, p_n) \in Q\}.$$

Theorem 2. For every n-MGR in the X mode, where $X \in \{union, conc, first\}$, there is an equivalent matrix grammar.

Proof. This follows from Algorithm 3, Algorithm 4 and Algorithm 5.

□

Algorithm 6. Conversion of matrix grammar to 2-MGR

- ***Input:*** Matrix grammar $H = (G, M)$; string $\overline{w} \in \overline{T}^*$, where $\overline{T}$ is any alphabet
- ***Output:*** 2-MGR $\Gamma = (G_1, G_2, Q)$; $\{w_1: (w_1, \overline{w}) \in 2\text{-}L(\Gamma)\} = L(H)$
- ***Method:***

Let $G = (N, T, P, S)$, then:

$G_1 = G$;

$G_2 = (N_2, T_2, P_2, S_2)$, where:

$N_2 = \{S_2\} \cup \{<p_1p_2... p_k, j>: p_1p_2... p_k \in M, 1 \leq j \leq k-1\}$; $T_2 = \overline{T}$;

$P_2 = \{S_2 \rightarrow <p_1p_2... p_k, 1>: p_1p_2... p_k \in M, k \geq 2\} \cup$

$\{<p_1p_2... p_k, j> \rightarrow <p_1p_2... p_k, j+1>: p_1p_2... p_k \in M, k \geq 2, 1 \leq j \leq k-2\}\} \cup$

$\{<p_1p_2... p_k, k-1> \rightarrow S_2: p_1p_2... p_k \in M, k \geq 2\} \cup$

$\{S_2 \rightarrow S_2: p_1 \in M, |p_1| = 1\} \cup$

$\{<p_1p_2... p_k, k-1> \rightarrow \overline{w}: p_1p_2... p_k \in M, k \geq 2\} \cup$

$\{S_2 \rightarrow \overline{w}: p_1 \in M, |p_1| = 1\}$;

$Q = \{(p_1, S_2 \rightarrow <p_1p_2... p_k, 1>): p_1p_2... p_k \in M, k \geq 2\} \cup$

$\{(p_{j+1}, <p_1p_2... p_k, j> \rightarrow <p_1p_2... p_k, j+1>): p_1p_2... p_k \in M, k \geq 2, 1 \leq j \leq k-2\} \cup$

$\{(p_k, <p_1p_2... p_k, k-1> \rightarrow S_2): p_1p_2... p_k \in M, k \geq 2\} \cup$

$\{(p_1, S_2 \rightarrow S_2): p_1 \in M, |p_1| = 1\} \cup$

$\{(p_k, <p_1p_2... p_k, k-1> \rightarrow \overline{w}): p_1p_2... p_k \in M, k \geq 2\} \cup$

$\{(p_1, S_2 \rightarrow \overline{w}): p_1 \in M, |p_1| = 1\}$.

Claim 2. For every matrix grammar H, there is an equivalent 2-MGR in the concatenation mode.

Proof. Use Algorithm 6 with matrix grammar H and $\overline{w} = \varepsilon$ in the input.

□

Claim 3. For every matrix grammar H, there is an equivalent 2-MGR in the first mode.

Proof. Use Algorithm 6 with matrix grammar H and any string $\overline{w} \in \overline{T}^*$ in the input.

□

Claim 4. For every matrix grammar H, there is an equivalent 2-MGR in the union mode.

Proof. Use Algorithm 6 with matrix grammar H and $\overline{w}$ in the input, where $\overline{w}$ is any string in $L(H)$, provided that $L(H)$ is nonempty. Otherwise, $\overline{w}$ is any string.

□

Theorem 3. For every matrix grammar, there is an equivalent 2-MGR in the X mode, where $X \in \{union, conc, first\}$.

Proof. This follows from Claim 2, Claim 3 and Claim 4.

□

3 Conclusion

Let $\mathcal{L}$(n-MGN$_X$) and $\mathcal{L}$(n-MGR$_X$) denote the language families defined by n-MGN in the X mode and n-MGR in the X mode, respectively, where $X \in \{union, conc, first\}$, let $\mathcal{L}$(H) denote the family of languages generated by matrix grammars. From the previous results, we obtain:

- $\mathcal{L}(\text{H}) = \mathcal{L}(i\text{-MGN}_X)$, $i \geq 2$.
- $\mathcal{L}(\text{H}) = \mathcal{L}(i\text{-MGR}_X)$, $i \geq 2$.

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) 126-137
6. Paun, Gh., Salomaa, A. and S. Vicolov, S.: On the generative capacity of parallel communicating grammar systems. International Journal of Computer Mathematics 45, (1992) 45-59
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 (1998) 319-329