Kybernetika

Roman Lukáš; Alexander Meduna

Multigenerative grammar systems and matrix grammars

Kybernetika, Vol. 46 (2010), No. 1, 68--82

Persistent URL: http://dml.cz/dmlcz/140054

Terms of use:

© Institute of Information Theory and Automation AS CR, 2010

Institute of Mathematics of the Academy of Sciences of the Czech Republic provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.

This paper has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* http://project.dml.cz

KYBERNETIKA — VOLUME *46* (2010), NUMBER 1, PAGES 68–82

MULTIGENERATIVE GRAMMAR SYSTEMS AND MATRIX GRAMMARS

Roman Lukáš and Alexander Meduna

Multigenerative grammar systems are based on cooperating context-free grammatical components that simultaneously generate their strings in a rule-controlled or nonterminal-controlled rewriting way, and after this simultaneous generation is completed, all the generated terminal strings are combined together by some common string operations, such as concatenation, and placed into the generated languages of these systems. The present paper proves that these systems are equivalent with the matrix grammars. In addition, we demonstrate that these systems with any number of grammatical components can be transformed to equivalent two-component versions of these systems. The paper points out that if these systems work in the leftmost rewriting way, they are more powerful than the systems working in a general way.

Keywords: multigenerative grammar systems, simultaneously controlled derivations, matrix grammars

Classification: 68Q05, 68Q45

1. INTRODUCTION

Indisputably, the investigation of cooperating distributed grammar systems represents a crucially important trend in today's formal language theory (see [1, 2, 3, 4, 8, 9, 10, 13, 15, 18]). In essence, these grammars consist of several cooperating grammatical components that generate a single string (see [6] for an overview of the key concepts and results). Recently, a completely new type of these grammar systems, called multigenerative grammar systems, have been introduced (see [14]).

As opposed to the other cooperating distributed grammar systems, all the grammatical components of the multigenerative grammar systems systems simultaneously generate their strings in a rule-controlled or nonterminal-controlled rewriting way, and this generation is performed in the leftmost ways – that is, during one generation step, each component rewrites the leftmost occurrence of a nonterminal in its sentential form. After this simultaneous leftmost generation is completed, all the generated strings are composed into a single string by some common string operation, such as concatenation. More precisely, for a positive integer n, an n-generative grammar system works with n context-free grammatical components, each of which makes a leftmost derivation, and these n leftmost derivations are simultaneously controlled by a finite set of n-tuples consisting of nonterminals or rules. In this way, the grammar system generates n terminal strings, which are combined together by operation union, concatenation or the selection of the first generated string. The main result concerning the power of these systems says that they characterize the family of recursively enumerable languages (see Theorem 3 in [14]).

In this paper, we discuss general versions of multigenerative grammar systems by dropping the requirement that each generation step is leftmost. In other words, each grammatical component rewrites any nonterminal occurrence in its sentential form; otherwise, they work as described above. We prove that multigenerative grammar systems generalized in this way are less powerful than their leftmost versions in the present paper. More specifically, they are equivalent to the matrix grammars, which generate a proper subfamily of the family of recursively enumerable languages. This result is indeed of some interest when compared to the corresponding results in terms of other language models. In terms of context-free grammars, their leftmost versions and their general versions are equally powerful (see Theorem 5.1.1.1 in [12]). In terms of programmed grammars, the leftmost versions are less powerful than the general versions (see Theorem 1.4.1 in [5]).

Considering these results, it comes as a surprise that general versions of multigenerative grammar systems are less powerful than their leftmost versions as proved in the present paper. In addition, we demonstrate that multigenerative grammar systems with any number of grammatical components can be transformed to equivalent two-component versions of these systems.

2. DEFINITIONS

This paper assumes that the reader is familiar with the formal language theory (see [11, 12, 16, 17]). For a set, Q, $\mathrm{card}(Q)$ 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.

Definition 2.1. 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, 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, this is written as $(r : A \to x)$. Let $u, v \in (N \cup T)^*$. For every $(r : A \to x \in P)$, we 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 $L(G) = \{w \in T^* | S \Rightarrow^* w \text{ in } G\}$.

Definition 2.2. 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 $n \geq 1$, $x_{i-1} \Rightarrow_G x_i[r_i]$ in G for all $i = 1, \ldots, n$ and $r_1 r_2 \ldots r_n \in M$. Then matrix grammar H makes direct derivation step from x_0 to x_n, denoted as $x_0 \Rightarrow_H x_n$. Let $\Rightarrow^*$ denote the transitive-reflexive closure of $\Rightarrow$. The language of H, $L(H)$, is defined as $L(H) = \{w \in T^* | S \Rightarrow^* w \text{ in } H\}$.

Definition 2.3. A *general n-generative rule-synchronized grammar system* (n-GGR) 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 $(p_1, p_2, \ldots, p_n)$, where $p_i \in P_i$ for all $i = 1, \ldots, n$. Let $\Gamma = (G_1, G_2, \ldots, G_n, Q)$ be an n-GGR. Then, a sentential n-form of n-GGR 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 $\chi = (u_1 A_1 v_1, u_2 A_2 v_2, \ldots, u_n A_n v_n)$ and $\overline{\chi} = (u_1 x_1 v_1, u_2 x_2 v_2, \ldots, u_n x_n v_n)$ be two sentential n-form, where $A_i \in N_i$ and $u_i, v_i, x_i \in (N_i \cup T_i)^*$ for all $i = 1, \ldots, n$. Let $(p_i : A_i \to x_i) \in P_i$ for all $i = 1, \ldots, n$ and $(p_1, p_2, \ldots, p_n) \in Q$. Then χ directly derives $\overline{\chi}$ in Γ, denoted by $\chi \Rightarrow \overline{\chi}$. In the standard way, we generalize $\Rightarrow$ to $\Rightarrow^k$ for all $k \geq 0$, $\Rightarrow^*$, and $\Rightarrow^+$.

The *n-language of* Γ, n-$L(\Gamma)$, is defined as
n-$L(\Gamma) = \{(w_1, w_2, \ldots, w_n) | (S_1, S_2, \ldots, S_n) \Rightarrow^* (w_1, w_2, \ldots, w_n), w_i \in T_i^* \text{ for all } i = 1, \ldots, n\}$

The *language generated by* Γ *in the union mode*, $L_{\text{union}}(\Gamma)$, is defined as
$L_{\text{union}}(\Gamma) = \bigcup_{i=1}^{n} \{w_i | (w_1, w_2, \ldots, w_n) \in n\text{-}L(\Gamma)\}$

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

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

Example 2.4. $\Gamma = (G_1, G_2, Q)$, where

$$\begin{aligned}
G_1 &= (\{S_1, A_1\}, \{a, b, c\}, \{(1 : S_1 \to aS_1), (2 : S_1 \to aA_1), (3 : A_1 \to bA_1c), \\
&\qquad (4 : A_1 \to bc)\}, S_1), \\
G_2 &= (\{S_2\}, \{d\}, \{(1 : S_2 \to S_2S_2), (2 : S_2 \to S_2), (3 : S_2 \to d)\}, S_2), \\
Q &= \{(1,1), (2,2), (3,3), (4,3)\}
\end{aligned}$$

is a general 2-generative rule-synchronized grammar system.

Notice that 2-$L(\Gamma) = \{(a^n b^n c^n, d^n) | n \geq 1\}$, $L_{\text{union}}(\Gamma) = \{a^n b^n c^n | n \geq 1\} \cup$ $\cup \{d^n | n \geq 1\}$, $L_{\text{conc}}(\Gamma) = \{a^n b^n c^n d^n | n \geq 1\}$, and $L_{\text{first}}(\Gamma) = \{a^n b^n c^n | n \geq 1\}$.

3. RESULTS

In this section, we prove that all variants of multigenerative grammar systems defined in the previous section are equivalent to the matrix grammars.

Algorithm 3.1. A conversion of an n-GGR in the union mode to an equivalent matrix grammar

- *Input:* An n-GGR $\Gamma = (G_1, G_2, \ldots G_n, Q)$.
- *Output:* A matrix grammar $H = (G, M)$ satisfying $L_{\text{union}}(\Gamma) = L(H)$.
- *Method:*
  - Let $G_i = (N_i, T_i, P_i, S_i)$ for all $i = 1, \ldots, n$, and without loss of generality, we can assume that for any $j, k = 1, \ldots, n$, where $j \neq k$, it holds: $N_j \cap N_k = \emptyset$; let us choose arbitrary S satisfying $S \notin \bigcup_{j=1}^{n} N_j$. Then:
  - $G = (N, T, P, S)$, where:
    $N := \{S\} \cup (\bigcup_{i=1}^{n} N_i) \cup (\bigcup_{i=1}^{n} \{\overline{A} | A \in N_i\})$;
    $T := \bigcup_{i=1}^{n} T_i$;
    $P := \{(s_1 : S \to S_1 h(S_2) \ldots h(S_n)), (s_2 : S \to h(S_1) S_2 \ldots h(S_n)), \ldots (s_n : S \to h(S_1) h(S_2) \ldots S_n)\} \cup (\bigcup_{i=1}^{n} P_i) \cup (\bigcup_{i=1}^{n} \{h(A) \to h(x) | A \to x \in P_i\})$, 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 $a \in \bigcup_{i=1}^{n} T_i$ and $h(A) = \overline{A}$ for all $A \in \bigcup_{i=1}^{n} N_i$.
  - $M = \{s_1, s_2, \ldots, s_n\} \cup \{p_1 \overline{p_2} \ldots \overline{p_n} | (p_1, p_2, \ldots, p_n) \in Q\} \cup$
    $\{\overline{p_1} p_2 \ldots \overline{p_n} | (p_1, p_2, \ldots, p_n) \in Q\} \cup \ldots \cup \{\overline{p_1}\,\overline{p_2} \ldots p_n | (p_1, p_2, \ldots, p_n) \in Q\}$.
    Notation:
    Let $(p : A \to x)$ be a rule. Then, $\overline{p}$ denotes the rule $h(A) \to h(x)$.

Claim 3.2. Let $(S_1, S_2, \ldots, S_n) \Rightarrow^m (y_1, y_2, \ldots, y_n)$ in Γ, where $m \geq 0, y_i \in (N_i \cup T_i)^*$ for all $i = 1, \ldots, n$. Then, $S \Rightarrow^{m+1} h(y_1) h(y_2) \ldots h(y_{j-1}) y_j h(y_{j+1}) \ldots h(y_n)$ for any $j = 1, \ldots, n$ in H.

Proof. This claim is proved by induction on $m \geq 0$.

Basis:
Let $m = 0$. Then, $(S_1, S_2, \ldots, S_n) \Rightarrow^0 (S_1, S_2, \ldots, S_n)$ in Γ.
Notice that $S \Rightarrow^1 h(S_1) h(S_2) \ldots h(S_{j-1}) S_j h(S_{j+1}) \ldots h(S_n)$ in H for any $j = 1, \ldots, n$, because $(s_j : S \to h(S_1) h(S_2) \ldots h(S_{j-1}) S_j h(S_{j+1}) \ldots h(S_n)) \in M$.

Induction hypothesis:
Assume that the claim holds for all m-step derivations, where $m = 0, \ldots, k$, for some $k \geq 0$.

Induction step:
Consider $(S_1, S_2, \ldots, S_n) \Rightarrow^{k+1} (y_1, y_2, \ldots, y_n)$ in Γ. Then, there exists a sentential n-form $(u_1 A_1 v_1, u_2 A_2 v_2, \ldots, u_n A_n v_n)$, where $u_i, v_i \in (T_i \cup N_i)^*, A_i \in N_i$ such that $(S_1, S_2, \ldots, S_n) \Rightarrow^k (u_1 A_1 v_1, u_2 A_2 v_2, \ldots, u_n A_n v_n) \Rightarrow (u_1 x_1 v_1, u_2 x_2 v_2, \ldots, u_n x_n v_n)$ in Γ, where $u_i x_i v_i = y_i$ for all $i = 1, \ldots, n$.

First, observe that $(S_1, S_2, \dots, S_n) \Rightarrow^k (u_1A_1v_1, u_2A_2v_2, \dots, u_nA_nv_n)$ in Γ implies
$S \Rightarrow^{k+1} h(u_1A_1v_1)h(u_2A_2v_2)\dots h(u_{j-1}A_{j-1}v_{j-1})u_jA_jv_jh(u_{j+1}A_{j+1}v_{j+1})\dots \dots h(u_nA_nv_n)$ for any $j = 1, \dots, n$ in H by the induction hypothesis.

Furthermore, let $(u_1A_1v_1, u_2A_2v_2, \dots, u_nA_nv_n) \Rightarrow (u_1x_1v_1, u_2x_2v_2, \dots, u_nx_nv_n)$ in Γ Then, it holds: $((p_1 : A_1 \to x_1), (p_2 : A_2 \to x_2), \dots, (p_n : A_n \to x_n)) \in Q$. Algorithm 1 implies that
$\overline{p_1}\overline{p_2}\dots\overline{p_{j-1}}p_j\overline{p_{j+1}}\dots\overline{p_n} \in M$ for any $j = 1, \dots, n$. Hence,
$h(u_1A_1v_1)h(u_2A_2v_2)\dots h(u_{j-1}A_{j-1}v_{j-1})u_jA_jv_jh(u_{j+1}A_{j+1}v_{j+1})\dots h(u_nA_nv_n) \Rightarrow h(u_1x_1v_1)h(u_2x_2v_2)\dots h(u_{j-1}x_{j-1}v_{j-1})u_jx_jv_jh(u_{j+1}x_{j+1}v_{j+1})\dots h(u_nx_nv_n)$ in H by the matrix $\overline{p_1}\overline{p_2}\dots\overline{p_{j-1}}p_j\overline{p_{j+1}}\dots\overline{p_n}$ for any $j = 1, \dots, n$.

As a result, we obtain:
$S \Rightarrow^{k+2} h(u_1x_1v_1)h(u_2x_2v_2)\dots h(u_{j-1}x_{j-1}v_{j-1})u_jx_jv_jh(u_{j+1}x_{j+1}v_{j+1})\dots h(u_nx_nv_n)$ in H for any $j = 1, \dots, n$. $\square$

Claim 3.3. Consider derivation steps $S \Rightarrow^m y$ in H, where $m \geq 1, y \in (N \cup T)^*$. Then, there exist $j \in \{1, \dots, n\}$ and $y_i \in (N_i \cup T_i)^*$ for $i = 1, \dots, n$ such that $(S_1, \dots, S_n) \Rightarrow^{m-1} (y_1, \dots, y_n)$ in Γ and $y = h(y_1)\dots h(y_{j-1})y_jh(y_{j+1})\dots h(y_n)$.

Proof. This claim is proved by induction on $m \geq 1$.

Basis:
Let $m = 1$. Then, there exists exactly one of the following one-step derivation in H: $S \Rightarrow^1 S_1h(S_2)\dots h(S_n)$ by the matrix s_1 or $S \Rightarrow^1 h(S_1)S_2\dots h(S_n)$ by the matrix s_2 or …or $S \Rightarrow^1 h(S_1)h(S_2)\dots S_n$ by the matrix s_n. Notice that $(S_1, S_2, \dots, S_n) \Rightarrow^0 (S_1, S_2, \dots, S_n)$ in Γ trivially.

Induction hypothesis:
Assume that the claim holds for all m-step derivations, where $m = 1, \dots, k$, for some $k \geq 1$.

Induction step:
Consider $S \Rightarrow^{k+1} y$ in H. Then, there exists a sentential form w such that $S \Rightarrow^k w \Rightarrow y$ in H, where $w, y \in (N \cup T)^*$.

As $w \Rightarrow y$ in H, this derivation step can use only a matrix of a following form $p_1p_2\dots p_{j-1}p_jp_{j+1}\dots p_n \in Q$, where p_j is a rule from P_j and $\overline{p_i} \in h(P_i)$ for $i = 1, \dots, j-1, j+1, \dots, n$. Hence, $w \Rightarrow y$ can be written as
$h(w_i)\dots h(w_{j-1})w_jh(w_{j+1})\dots h(w_n) \Rightarrow z_1\dots z_n$, where $w_j \Rightarrow z_j$ by the rule p_j and $h(w_i) \Rightarrow z_i$ by $\overline{p_i}$ for $i = 1, \dots, j-1, j+1, \dots, n$. Each rule $\overline{p_i}$ rewrites a barred nonterminal $\overline{A_i} \in h(N_i)$. Of course, then each rule p_i can be used to rewrite the respective occurrence of a non-barred nonterminal A_i in w_i in such a way that $w_i \Rightarrow y_i$ and $h(y_i) = z_i$, for all $i = 1, \dots, j-1, j+1, \dots, n$. By setting $y_j = z_j$, we obtain $(w_1, \dots, w_n) \Rightarrow (y_1, \dots, y_n)$ in Γ and $y = h(y_1)\dots h(y_{j-1})y_jh(y_{j+1})\dots h(y_n)$.

As a result, we obtain:
$(S_1, S_2, \dots, S_{j-1}, S_j, S_{j+1}, \dots, S_n) \Rightarrow^k$
$(u_1x_1v_1, u_2x_2v_2, \dots, u_{j-1}x_{j-1}v_{j-1}, u_jx_jv_j, u_{j+1}x_{j+1}v_{j+1}, \dots, u_nx_nj_n)$ in Γ so that $y = u_1x_1v_1u_2x_2v_2\dots u_{j-1}x_{j-1}v_{j-1}u_jx_jv_ju_{j+1}x_{j+1}v_{j+1}\dots u_nx_nv_n$. $\square$

Theorem 3.4. Let $\Gamma = (G_1, G_2, \dots G_n, Q)$ be a n-GGR. On input Γ, Algorithm 1 halts and correctly constructs a matrix grammar $H = (G, M)$ such that $L_{\text{union}}(\Gamma) = L(H)$.

Proof. Consider Claim 1 for any $m \geq 0$ and $y_i \in T_i^*$ for all $i = 1, \dots, n$. Notice that $h(a) = \varepsilon$ for all $a \in T_i$. We obtain an implication of the form: if $(S_1, S_2, \dots, S_n) \Rightarrow^* (y_1, y_2, \dots, y_n)$ in Γ, then $S \Rightarrow^* y_j$ for any $j = 1, \dots, n$ in H. Hence, $L_{\text{union}}(\Gamma) \subseteq L(H)$. Consider Claim 2 for any $m \geq 1$ and $y \in T^*$. Notice that $h(a) = \varepsilon$ for all $a \in T_i$. We obtain an implication of the form: if $S \Rightarrow^* y$ in H, then $(S_1, S_2, \dots, S_n) \Rightarrow^* (y_1, y_2, \dots, y_n)$ in Γ, and there exist an index $j = 1, \dots, n$ such that $y = y_j$. Hence, $L(H) \subseteq L_{\text{union}}(\Gamma)$. □

Algorithm 3.5. A conversion of an n-GGR in the concatenation mode to an equivalent matrix grammar

- *Input:* An n-GGR $\Gamma = (G_1, G_2, \dots G_n, Q)$.
- *Output:* A matrix grammar $H = (G, M)$ satisfying $L_{\text{conc}}(\Gamma) = L(H)$.
- *Method:*
  - Let $G_i = (N_i, T_i, P_i, S_i)$ for all $i = 1, \dots, n$, and without loss of generality, we can assume that for any $j, k = 1, \dots, n$, where $j \neq k$, it holds: $N_j \cap N_k = \emptyset$; let us choose arbitrary S satisfying $S \notin \bigcup_{j=1}^{n} N_j$. Then:
  - $G = (N, T, P, S)$, where:
    $N := \{S\} \cup (\bigcup_{i=1}^{n} N_i)$;
    $T := \bigcup_{i=1}^{n} T_i$;
    $P := \{(s : S \to S_1 S_2 \dots S_n)\} \cup (\bigcup_{i=1}^{n} P_i)$.
  - $M = \{s\} \cup \{p_1 p_2 \dots p_n | (p_1, p_2, \dots, p_n) \in Q\}$.

Claim 3.6. Consider a sequence of derivation steps $(S_1, S_2, \dots, S_n) \Rightarrow^m (y_1, y_2, \dots, y_n)$ in Γ, where $m \geq 0, y_i \in (N_i \cup T_i)^*$ for all $i = 1, \dots, n$. Then, $S \Rightarrow^{m+1} y_1 y_2 \dots y_n$.

Proof. This claim is proved by induction on $m \geq 0$.

Basis:
Let $m = 0$. Then, $(S_1, S_2, \dots, S_n) \Rightarrow^0 (S_1, S_2, \dots, S_n)$ in Γ.
Notice that $S \Rightarrow^1 S_1 S_2 \dots S_n$ in H, because $(s : S \to S_1 S_2 \dots S_n) \in M$.

Induction hypothesis:
Assume that the claim holds for all m-step derivations, where $m = 0, \dots, k$, for some $k \geq 0$.

Induction step:
Consider $(S_1, S_2, \dots, S_n) \Rightarrow^{k+1} (y_1, y_2, \dots, y_n)$ in Γ. Then, there exists a sentential n-form $(u_1 A_1 v_1, u_2 A_2 v_2, \dots, u_n A_n v_n)$, where $u_i, v_i \in (T_i \cup N_i)^*, A_i \in N_i$ such that $(S_1, S_2, \dots, S_n) \Rightarrow^k (u_1 A_1 v_1, u_2 A_2 v_2, \dots, u_n A_n v_n) \Rightarrow (u_1 x_1 v_1, u_2 x_2 v_2, \dots, u_n x_n v_n)$ in Γ, where $u_i x_i v_i = y_i$ for all $i = 1, \dots, n$.

First, observe that $(S_1, S_2, \dots, S_n) \Rightarrow^k (u_1A_1v_1, u_2A_2v_2, \dots, u_nA_nv_n)$ in Γ implies
$S \Rightarrow^{k+1} u_1A_1v_1u_2A_2v_2 \dots u_nA_nv_n$ in H by the induction hypothesis.

Furthermore, let $(u_1A_1v_1, u_2A_2v_2, \dots, u_nA_nv_n) \Rightarrow (u_1x_1v_1, u_2x_2v_2, \dots, u_nx_nv_n)$ in Γ. Then, it holds: $((p_1 : A_1 \to x_1), (p_2 : A_2 \to x_2), \dots, (p_n : A_n \to x_n)) \in Q$. Algorithm 2 implies that $p_1p_2 \dots p_n \in M$. Hence,
$u_1A_1v_1u_2A_2v_2 \dots u_nA_nv_n \Rightarrow u_1x_1v_1u_2x_2v_2 \dots u_nx_nv_n$ in H by the matrix $p_1p_2 \dots p_n$.

As a result, we obtain:

$S \Rightarrow^{k+2} u_1x_1v_1u_2x_2v_2 \dots u_nx_nv_n$ in H. □

Claim 3.7. Let $S \Rightarrow^m y$ in H, where $m \geq 1, y \in (N \cup T)^*$. Then, $(S_1, S_2, \dots, S_n) \Rightarrow^{m-1} (y_1, y_2, \dots, y_n)$ in Γ, where $y_i \in (N_i \cup T_i)^*$ for all $i = 1, \dots, n$ such that $y = y_1y_2 \dots y_n$.

P r o o f. This claim is proved by induction on $m \geq 1$.

Basis:
Let $m = 1$. Then, there exists exactly one one-step derivation in H: $S \Rightarrow^1 S_1S_2 \dots, S_n$ by the matrix s. Notice that $(S_1, S_2, \dots, S_n) \Rightarrow^0 (S_1, S_2, \dots, S_n)$ in Γ trivially.

Induction hypothesis:
Assume that the claim holds for all m-step derivations, where $m = 1, \dots, k$, for some $k \geq 1$.

Induction step:
Consider $S \Rightarrow^{k+1} y$ in H. Then, there exists a sentential form w such that $S \Rightarrow^k w \Rightarrow y$ in H, where $w, y \in (N \cup T)^*$.

First, observe that $S \Rightarrow^k w$ in H implies that $(S_1, S_2, \dots, S_n) \Rightarrow^{k-1} (w_1, w_2, \dots, w_n)$ in Γ so that $w = w_1w_2 \dots w_n$, where $w_i \in (N_i \cup T_i)^*$ for all $i = 1, \dots, n$, by the induction hypothesis.

Furthermore, let $w \Rightarrow y$ in H by the matrix $p_1p_2 \dots p_n \in M$, where $w = w_1w_2 \dots w_n$. Let p_i be a rule of the form $A_i \to x_i$. The rule p_i can be applied only inside substring w_i, for all $i = 1, \dots, n$. Assume that $w_i = u_iA_iv_i$, where $u_i, v_i \in (N \cup T)^*, A_i \in N_i$ for all $i = 1, \dots, n$. There exist a derivation step
$u_1A_1v_1u_2A_2v_2 \dots u_nA_nv_n \Rightarrow u_1x_1v_1u_2x_2v_2 \dots u_nx_nv_n$ in H by the matrix $p_1p_2 \dots p_n \in M$. Algorithm 2 implies that $((p_1 : A_1 \to x_1), (p_2 : A_2 \to x_2), \dots, (p_n : A_n \to x_n)) \in Q$, because $p_1p_2 \dots p_n \in M$. Hence,
$(u_1A_1v_1, u_2A_2v_2, \dots, u_nA_nj_n \Rightarrow (u_1x_1v_1, u_2x_2v_2, \dots, u_nx_nj_n)$ in Γ.

As a result, we obtain:

$(S_1, S_2, \dots, S_n) \Rightarrow^k (u_1x_1v_1, u_2x_2v_2, \dots, u_nx_nj_n)$ in Γ so that
$y = u_1x_1v_1u_2x_2v_2 \dots u_nx_nv_n$. □

Theorem 3.8. Let $\Gamma = (G_1, G_2, \dots G_n, Q)$ be a n-GGR. On input Γ, Algorithm 2 halts and correctly constructs a matrix grammar $H = (G, M)$ such that $L_{\text{conc}}(\Gamma) = L(H)$.

P r o o f. Consider Claim 3 for any $m \geq 0$ and $y_i \in T_i^*$ for all $i = 1, \dots, n$. We obtain an implication of the form: if $(S_1, S_2, \dots, S_n) \Rightarrow^* (y_1, y_2, \dots, y_n)$ in Γ, then $S \Rightarrow^* y_1 y_2 \dots y_n$ in H. Hence, $L_{\text{conc}}(\Gamma) \subseteq L(H)$. Consider Claim 4 for any $m \geq 1$ and $y \in T^*$. We obtain an implication of the form: if $S \Rightarrow^* y$ in H, then $(S_1, S_2, \dots, S_n) \Rightarrow^* (y_1, y_2, \dots, y_n)$ in Γ, such that $y = y_1 y_2 \dots y_n$. Hence, $L(H) \subseteq L_{\text{conc}}(\Gamma)$. □

Algorithm 3.9. A conversion of an n-GGR in the first mode to an equivalent matrix grammar

- *Input:* An n-GGR $\Gamma = (G_1, G_2, \dots G_n, Q)$.
- *Output:* A matrix grammar $H = (G, M)$ satisfying $L_{\text{first}}(\Gamma) = L(H)$.
- *Method:*
  - Let $G_i = (N_i, T_i, P_i, S_i)$ for all $i = 1, \dots, n$, and without loss of generality, we can assume that for any $j, k = 1, \dots, n$, where $j \neq k$, it holds: $N_j \cap N_k = \emptyset$; let us choose arbitrary S satisfying $S \notin \bigcup_{j=1}^{n} N_j$. Then:
  - $G = (N, T, P, S)$, where:
    $N := \{S\} \cup N_1 \cup (\bigcup_{i=2}^{n} \{\overline{A} : A \in N_i\})$;
    $T := T_1$;
    $P := \{(s : S \to S_1 h(S_2) \dots h(S_n))\} \cup P_1 \cup (\bigcup_{i=2}^{n} \{h(A) \to h(x) | A \to x \in P_i\})$,
    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 $a \in \bigcup_{i=2}^{n} T_i$ and $h(A) = \overline{A}$ for all $A \in \bigcup_{i=2}^{n} N_i$.
  - $M = \{s\} \cup \{p_1 \overline{p_2} \dots \overline{p_n} | (p_1, p_2, \dots, p_n) \in Q\}$.
    Notation:
    Let $p = A \to x$ be a rule. Then, $\overline{p}$ denotes the rule $h(A) \to h(x)$.

Claim 3.10. Let $(S_1, S_2, \dots, S_n) \Rightarrow^m (y_1, y_2, \dots, y_n)$ in Γ, where $m \geq 0, y_i \in (N_i \cup T_i)^*$ for all $i = 1, \dots, n$. Then, $S \Rightarrow^{m+1} y_1 h(y_2) \dots h(y_n)$ in H.

P r o o f. This claim is proved by induction on $m \geq 0$.

Basis:

Let $m = 0$. Then, $(S_1, S_2, \dots, S_n) \Rightarrow^0 (S_1, S_2, \dots, S_n)$ in Γ.
Notice that $S \Rightarrow^1 S_1 h(S_2) \dots h(S_n)$ in H, because $(s : S \to S_1 h(S_2) \dots h(S_n)) \in M$.

Induction hypothesis:

Assume that the claim holds for all m-step derivations, where $m = 0, \dots, k$, for some $k \geq 0$.

Induction step:

Consider $(S_1, S_2, \dots, S_n) \Rightarrow^{k+1} (y_1, y_2, \dots, y_n)$ in Γ. Then, there exists a sentential n-form $(u_1 A_1 v_1, u_2 A_2 v_2, \dots, u_n A_n v_n)$, where $u_i, v_i \in (T_i \cup N_i)^*, A_i \in N_i$ such that $(S_1, S_2, \dots, S_n) \Rightarrow^k (u_1 A_1 v_1, u_2 A_2 v_2, \dots, u_n A_n v_n) \Rightarrow (u_1 x_1 v_1, u_2 x_2 v_2, \dots, u_n x_n v_n)$ in Γ, where $u_i x_i v_i = y_i$ for all $i = 1, \dots, n$.

First, observe that $(S_1, S_2, \ldots, S_n) \Rightarrow^k (u_1A_1v_1, u_2A_2v_2, \ldots, u_nA_nv_n)$ in Γ implies
$S \Rightarrow^{k+1} u_1A_1v_1h(u_2A_2v_2)\ldots h(u_nA_nv_n)$ in H by the induction hypothesis.

Furthermore, let $(u_1A_1v_1, u_2A_2v_2, \ldots, u_nA_nv_n) \Rightarrow (u_1x_1v_1, u_2x_2v_2, \ldots, u_nx_nv_n)$ in Γ Then, it holds: $((p_1 : A_1 \to x_1), (p_2 : A_2 \to x_2), \ldots, (p_n : A_n \to x_n)) \in Q$. Algorithm 3 implies that
$p_1\overline{p_2} \ldots \ldots \overline{p_n} \in M$. Hence,
$u_1A_1v_1h(u_2A_2v_2)\ldots h(u_nA_nv_n) \Rightarrow u_1x_1v_1h(u_2x_2v_2)\ldots h(u_nx_nv_n)$ in H by the matrix $p_1\overline{p_2}\ldots\overline{p_n}$.

As a result, we obtain:

$S \Rightarrow^{k+2} u_1x_1v_1h(u_2x_2v_2)\ldots h(u_nx_nv_n)$ in H. □

Claim 3.11. Let $S \Rightarrow^m y$ in H, where $m \geq 1, y \in (N \cup T)^*$. Then, $(S_1, S_2, \ldots, S_n) \Rightarrow^{m-1} (y_1, y_2, \ldots, y_n)$ in Γ, where $y_i \in (N_i \cup T_i)^*$ for all $i = 1, \ldots, n$ so that $y = y_1h(y_2)\ldots h(y_n)$.

Proof. This claim is proved by induction on $m \geq 1$.

Basis:
Let $m = 1$. Then, there exists exactly one one-step derivation in H: $S \Rightarrow^1 S_1h(S_2)\ldots h(S_n)$ by the matrix s. Notice that $(S_1, S_2, \ldots, S_n) \Rightarrow^0 (S_1, S_2, \ldots, S_n)$ in Γ trivially.

Induction hypothesis:
Assume that the claim holds for all m-step derivations, where $m = 1, \ldots, k$, for some $k \geq 1$.

Induction step:
Consider $S \Rightarrow^{k+1} y$ in H. Then, there is w such that $S \Rightarrow^k w \Rightarrow y$ in H, where $w, y \in (N \cup T)^*$.

First, observe that $S \Rightarrow^k w$ in H implies that $(S_1, S_2, \ldots, S_n) \Rightarrow^{k-1} (w_1, w_2, \ldots, w_n)$ in Γ so that $w = w_1h(w_2)\ldots h(w_n)$,where $w_i \in (N_i \cup T_i)^*$ for all $i = 1, \ldots, n$, by the induction hypothesis.

Furthermore, let $w \Rightarrow y$ in H, where $w = w_1h(w_2)\ldots h(w_n)$. Let p_1 be a rule of the form $A_1 \to x_1$. Let $\overline{p_i}$ be a rule of the form $h(A_i) \to h(x)$ for all $i = 2, \ldots, n$. The rule p_1 can be applied only inside substring w_1, the rule $\overline{p_i}$ can be applied only inside substring w_i, for all $i = 2, \ldots, n$. Assume that $w_i = u_iA_iv_i$, where $u_i, v_i \in (N_i \cup T_i)^*, A_i \in N_i$ for all $i = 1, \ldots, n$. There exists a derivation step
$u_1A_1v_1h(u_2A_2v_2)\ldots h(u_nA_nv_n) \Rightarrow u_1x_1v_1h(u_2x_2v_2)\ldots h(u_nx_nv_n)$ in H by the matrix $p_1\overline{p_2}\ldots\overline{p_n} \in M$. Algorithm 3 implies that
$((p_1 : A_1 \to x_1), (p_2 : A_2 \to x_2), \ldots, (p_n : A_n \to x_n)) \in Q$, because
$p_1\overline{p_2}\ldots\overline{p_n} \in M$. Hence,
$(u_1A_1v_1, u_2A_2v_2, \ldots, u_nA_nj_n) \Rightarrow (u_1x_1v_1, u_2x_2v_2, \ldots, u_nx_nj_n)$ in Γ

As a result, we obtain:

$(S_1, S_2, \ldots, S_n) \Rightarrow^k (u_1x_1v_1, u_2x_2v_2, \ldots, u_nx_nj_n)$ in Γ so that
$y = u_1x_1v_1h(u_2x_2v_2)\ldots h(u_nx_nv_n)$. □

Theorem 3.12. Let $\Gamma = (G_1, G_2, \dots G_n, Q)$ be a n-GGR. On input Γ, Algorithm 3 halts and correctly constructs a matrix grammar $H = (G, M)$ such that $L_{\text{first}}(\Gamma) = L(H)$.

Proof. Consider Claim 5 for any $m \geq 0$ and $y_i \in T_i^*$ for all $i = 1, \dots, n$. Notice that $h(a) = \varepsilon$ for all $a \in T_i$. We obtain an implication of the form: if $(S_1, S_2, \dots, S_n) \Rightarrow^* (y_1, y_2, \dots, y_n)$ in Γ, then $S \Rightarrow^* y_1$ in H. Hence, $L_{\text{first}}(\Gamma) \subseteq L(H)$. Consider Claim 6 for any $m \geq 1$ and $y \in T^*$. Notice that $h(a) = \varepsilon$ for all $a \in T_i$. We obtain an implication of the form: if $S \Rightarrow^* y$ in H, then $(S_1, S_2, \dots, S_n) \Rightarrow^* (y_1, y_2, \dots, y_n)$ in Γ, such that $y = y_1$. Hence, $L(H) \subseteq L_{\text{first}}(\Gamma)$. □

Algorithm 3.13. A conversion of a matrix grammar to a 2-GGR

- *Input:* A matrix grammar $H = (G, M)$; string $\overline{w} \in \overline{T}^*$, where $\overline{T}$ is any alphabet.
- *Output:* A 2-GGR $\Gamma = (G_1, G_2, Q)$ satisfying $\{w_1 | (w_1, \overline{w}) \in 2\text{-}L(\Gamma)\} = L(H)$.
- *Method:*
  - Let $G = (N, T, P, S)$. Then:
  - $G_1 = G$;
  - $G_2 = (N_2, T_2, P_2, S_2)$, where
    $N_2 := \{S_2\} \cup \{\langle p_1 p_2 \dots p_k, j\rangle | p_1, p_2 \dots p_k \in P, p_1 p_2 \dots p_k \in M, 1 \leq j \leq k-1\}$;
    $T_2 := \overline{T}$;
    $P_2 := \{S_2 \to \langle p_1 p_2 \dots p_k, 1\rangle | p_1, p_2 \dots p_k \in P, p_1 p_2 \dots p_k \in M, k \geq 2\} \cup$
    $\{\langle p_1 p_2 \dots p_k, j\rangle \to \langle p_1 p_2 \dots p_k, j+1\rangle | p_1 p_2 \dots p_k \in M, k \geq 2, 1 \leq j \leq k-2\} \cup$
    $\{\langle p_1 p_2 \dots p_k, k-1\rangle \to S_2 | p_1, p_2 \dots p_k \in P, p_1 p_2 \dots p_k \in M, k \geq 2\} \cup$
    $\{S_2 \to S_2 | p_1 \in M, |p_1| = 1\} \cup$
    $\{\langle p_1 p_2 \dots p_k, k-1\rangle \to \overline{w} | p_1, p_2 \dots p_k \in P, p_1 p_2 \dots p_k \in M, k \geq 2\} \cup$
    $\{S_2 \to \overline{w} | p_1 \in M, |p_1| = 1\}$;
  - $Q := \{(p_1, S_2 \to \langle p_1 p_2 \dots p_k, 1\rangle) | p_1, p_2 \dots p_k \in P, p_1 p_2 \dots p_k \in M, k \geq 2\} \cup$
    $\{(p_{j+1}, \langle p_1 p_2 \dots p_k, j\rangle \to \langle p_1 p_2 \dots p_k, j+1\rangle) | p_1 p_2 \dots p_k \in M, k \geq 2, 1 \leq j \leq k-2\} \cup$
    $\{(p_k, \langle p_1 p_2 \dots p_k, k-1\rangle \to S_2) | p_1, p_2 \dots p_k \in P, , p_1 p_2 \dots p_k \in M, k \geq 2\} \cup$
    $\{(p_1, S_2 \to S_2) | p_1 \in M, |p_1| = 1\} \cup$
    $\{(p_k, \langle p_1 p_2 \dots p_k, k-1\rangle \to \overline{w}) | p_1, p_2 \dots p_k \in P, p_1 p_2 \dots p_k \in M, k \geq 2\} \cup$
    $\{(p_1, S_2 \to \overline{w}) | p_1 \in M, |p_1| = 1\}$;

Claim 3.14. Let $x \Rightarrow y$ in H, where $x, y \in (N \cup T)^*$ Then, $(x, S_2) \Rightarrow^* (y, S_2)$ and $(x, S_2) \Rightarrow^* (y, \overline{w})$ in Γ.

P r o o f . In this proof, we distinguish two cases – I and II. In I, we consider a derivation step $x \Rightarrow y$ in H by a matrix consisting of a single rule. In II, we consider $x \Rightarrow y$ by a matrix consisting of several rules

I. Consider a derivation step $x \Rightarrow y$ in H by a matrix, which contains only one rule $(p_1 : A_1 \to x_1)$. It implies that $uA_1v \Rightarrow ux_1v[p_1]$ in G, where $uA_1v = x, ux_1v = y$. Algorithm 4 implies
$(A_1 \to x_1, S_2 \to S_2) \in Q$ and $(A_1 \to x_1, S_2 \to \overline{w}) \in Q$. Hence,
$(uA_1v, S_2) \Rightarrow^1 (ux_1v, S_2)$ and $(uA_1v, S_2) \Rightarrow^1 (ux_1v, \overline{w})$ in Γ.

II. Let $x \Rightarrow y$ in H by a matrix of the form $p_1p_2 \dots p_k$, where $p_i, \dots, p_k \in P, k \geq 2$. It implies that $x \Rightarrow y_1[p_1] \Rightarrow y_2[p_2] \Rightarrow \dots \Rightarrow y_{k-1}[p_{k_1}] \Rightarrow y_k[p_k]$, in G, where $y_k = y$. Algorithm 4 implies
$(p_1, S_2 \to \langle p_1p_2 \dots p_k, 1\rangle) \in Q$,
$(p_{j+1}, \langle p_1p_2 \dots p_k, j\rangle \to \langle p_1p_2 \dots p_k, j+1\rangle) \in Q$, where $j = 1, \dots, k-2$,
$(p_k, \langle p_1p_2 \dots p_k, k-1\rangle \to S_2) \in Q$,
$(p_k, \langle p_1p_2 \dots p_k, k-1\rangle \to \overline{w}) \in Q$.
Hence,
$(x, S_2) \Rightarrow (y_1, \langle p_1p_2 \dots p_k, 1\rangle) \Rightarrow (y_2, \langle p_1p_2 \dots p_k, 2\rangle) \Rightarrow \dots \Rightarrow (y_{k-1}, \langle p_1p_2 \dots p_k, k-1\rangle) \Rightarrow (y_k, S_2)$, where $y_k = y$ and $(x, S_2) \Rightarrow (y_1, \langle p_1p_2 \dots p_k, 1\rangle) \Rightarrow (y_2, \langle p_1p_2 \dots p_k, 2\rangle) \Rightarrow \dots \Rightarrow (y_{k-1}, \langle p_1p_2 \dots p_k, k-1\rangle) \Rightarrow (y_k, \overline{w})$, where $y_k = y$. □

Claim 3.15. Let $x \Rightarrow^m y$ in H, where $m \geq 1, y \in (N \cup T)^*$. Then, $(x, S_2) \Rightarrow^* (y, \overline{w})$ in Γ.

P r o o f . This claim is proved by induction on $m \geq 1$.

Basis:
Let $m = 1$ and let $x \Rightarrow^1 y$ in H. Claim 7 implies that $(x, S_2) \Rightarrow^* (y, \overline{w})$ in Γ.

Induction hypothesis:
Assume that the claim holds for all m-step derivations, where $m = 1, \dots, k$, for some $k \geq 1$.

Induction step:
Consider $S \Rightarrow^{k+1} y$ in H. Then, there exists w such that $S \Rightarrow w \Rightarrow^k y$ in H, where $w, y \in (N \cup T)^*$.

First, observe that $w \Rightarrow^k y$ in H implies that $(w, S_2) \Rightarrow^* (y, \overline{w})$ in Γ by the induction hypothesis.

Furthermore, let $x \Rightarrow w$ in H. Claim 7 implies that $(x, S_2) \Rightarrow^* (w, S_2)$ in Γ.

As a result, we obtain: $(x, S_2) \Rightarrow^* (y, \overline{w})$. □

Claim 3.16. Let $(y_0, S_2) \Rightarrow (y_1, z_1) \Rightarrow (y_2, z_2) \Rightarrow \dots \Rightarrow (y_{k-1}, z_{k-1}) \Rightarrow (y_k, S_2)$ or $(y_0, S_2) \Rightarrow (y_1, z_1) \Rightarrow (y_2, z_2) \Rightarrow \dots \Rightarrow (y_{k-1}, z_{k-1}) \Rightarrow (y_k, \overline{w})$ in Γ, where $z_i \neq S_2$ for all $i = 1, \dots, k-1$. Then, there exists a direct derivation step $y_0 \Rightarrow y_k$ in H.

P r o o f . In this proof, we distinguish two cases – I and II. In I, we consider a derivation step $x \Rightarrow y$ in H by a matrix consisting of a single rule. In II, we consider $x \Rightarrow y$ by a matrix consisting of several rules.

I. Let there exists only one derivation step of the form $(uA_1v, S_2) \Rightarrow (ux_1v, S_2)$ or $(uA_1v, S_2) \Rightarrow (ux_1v, \overline{w})$ in Γ, where $uA_1v = y_0, ux_1v = y_1$. Then, $(A_1 \rightarrow x_1, S_2 \rightarrow S_2) \in Q$ or $(A_1 \rightarrow x_1, S_2 \rightarrow \overline{w}) \in Q$. Algorithm 4 implies that there exists a matrix of the form $(p_1 : A_1 \rightarrow x_1) \in M$. Hence, $uA_1v \Rightarrow^1 ux_1v$ in H.

II. Let $(y_0, S_2) \Rightarrow (y_1, z_1) \Rightarrow (y_2, z_2) \Rightarrow \ldots \Rightarrow (y_{k-1}, z_{k-1}) \Rightarrow (y_k, S_2)$ or $(y_0, S_2) \Rightarrow (y_1, z_1) \Rightarrow (y_2, z_2) \Rightarrow \ldots \Rightarrow (y_{k-1}, z_{k-1}) \Rightarrow (y_k, \overline{w})$ in Γ, where $z_i \neq S_2$ for all $i = 1, \ldots, k-1$ and $k \geq 2$. Algorithm 4 implies that there exists a matrix $p_1p_2 \ldots p_k \in M$ and holds $z_i = \langle p_1p_2 \ldots p_k, i\rangle$ for all $i = 1, \ldots k-1$. Hence, $y_0 \Rightarrow y_k$ in H. □

Claim 3.17. Let $(y_0, S_2) \Rightarrow (y_1, z_1) \Rightarrow (y_2, z_2) \Rightarrow \ldots \Rightarrow (y_{r-1}, z_{r-1}) \Rightarrow (y_r, \overline{w})$ in Γ. Set $m = \mathrm{Card}(\{i | 1 \leq i \leq r-1, z_i = S_2\})$. Informally, m is number of z_i of the form S_2. Then, $y_0 \Rightarrow^{m+1} y_r$ in H.

Proof. This claim is proved by induction on $m \geq 0$.

Basis:

Let $m = 0$. Then, $z_i \neq S_2$ for all $i = 1, \ldots, k-1$. Claim 9 implies that there exists a derivation step $y_0 \Rightarrow^1 y_r$ in H.

Induction hypothesis:

Assume that the claim holds for all m-step derivations, where $m = 0, \ldots, k$, for some $k \geq 0$.

Induction step:

Consider $(y_0, S_2) \Rightarrow (y_1, z_1) \Rightarrow (y_2, z_2) \Rightarrow \ldots \Rightarrow (y_{r-1}, z_{r-1}) \Rightarrow (y_r, \overline{w})$ in Γ, where $\mathrm{Card}(\{i | 1 \leq i \leq r-1, z_i = S_2\}) = k+1$ Then, there exists $p \in \{1, \ldots, r-1\}$ such that $z_p = S_2$, $\mathrm{Card}(\{i | 1 \leq i \leq p-1, z_i = S_2\}) = 0$, $\mathrm{Card}(\{i | p+1 \leq i \leq r-1, z_i = S_2\}) = k$ and $(y_0, z_0) \Rightarrow \ldots \Rightarrow (y_p, z_p) \Rightarrow \ldots \Rightarrow (y_{r-1}, z_{r-1}) \Rightarrow (y_r, \overline{w})$ in Γ.

First, observe that $(y_p, z_p) \Rightarrow \ldots \Rightarrow (y_{r-1}, z_{r-1}) \Rightarrow (y_r, \overline{w})$, where $z_p = S_2$ and $\mathrm{Card}(\{i | p+1 \leq i \leq r-1, z_i = S_2\}) = k$ implies that $y_p \Rightarrow^{k+1} y_r$ in H by the induction hypothesis.

Furthermore, let $(y_0, z_0) \Rightarrow \ldots \Rightarrow (y_p, z_p)$. $\mathrm{Card}(\{i | 1 \leq i \leq p-1, z_i = S_2\}) = 0$ implies $z_i \neq S_2$ for all $i = 1, \ldots, p$. Claim 9 implies that there exists a derivation step $y_0 \Rightarrow^1 y_p$ in H.

As a result, we obtain: $y_0 \Rightarrow^{k+2} y_r$. □

Theorem 3.18. Let H be a matrix grammar and $\overline{w}$ be a word. On input H and $\overline{w}$, Algorithm 4 halts and correctly constructs a 2-GGR $\Gamma = (G_1, G_2, Q)$ such that $\{w_1 | (w_1, \overline{w}) \in 2\text{-}L(\Gamma)\} = L(H)$.

Proof. To establish this theorem, we next prove:

1. $\{w_1 | (w_1, \overline{w}) \in 2\text{-}L(\Gamma)\} = L(H)$.
   Consider Claim 8 for any $m \geq 1$, $x = S$ and $y \in T^*$. We obtain an implication of the form: if $S \Rightarrow^* y$ in H, then $(S, S_2) \Rightarrow^* (y, \overline{w})$ in Γ. Hence, $L(H) \subseteq \{w_1 | (w_1, \overline{w}) \in 2\text{-}L(\Gamma)\}$. Consider Claim 10 for any $m \geq 1$, $y_0 = S$ and $y_r \in T^*$. We see that if $(S, S_2) \Rightarrow^* (y_r, \overline{w})$ in Γ, then $S \Rightarrow^* y_r$ in H. Hence, $\{w_1 | (w_1, \overline{w}) \in 2\text{-}L(\Gamma)\} \subseteq L(H)$.

2. $\{(w_1, w_2)|(w_1, w_2) \in 2\text{-}L(\Gamma), w_2 \neq \overline{w}\} = \emptyset$.
   Notice that Algorithm 4 implies that grammar $G_2 = (N_2, T_2, P_2, S_2)$ contains only rules of the form $A \to B$ and $A \to \overline{w}$, where $A, B \in N_2$. Hence, G_2 generates $\emptyset$ or $\{\overline{w}\}$. Γ contains G_2 as a second component, hence $\{(w_1, w_2)|(w_1, w_2) \in 2\text{-}L(\Gamma), w_2 \neq \overline{w}\} = \emptyset$. □

Theorem 3.19. For every matrix grammar H, there is a 2-GGR Γ such that $L(H) = L_{\text{union}}(\Gamma)$.

Proof. We use Algorithm 4 with matrix grammar H and $\overline{w}$ as input, where $\overline{w}$ is any string in $L(H)$, provided that $L(H)$ is nonempty. Otherwise, $\overline{w}$ is any string. We prove that $L(H) = L_{\text{union}}(\Gamma)$.

1. If $L(H) = \emptyset$, take any word $\overline{w}$ and use Algorithm 4 to construct G. Observe that $L_{\text{union}}(\Gamma) = \emptyset = L(H)$.
2. If $L(H) \neq \emptyset$, take any $\overline{w} \in L(H)$ and use Algorithm 4 to construct Γ. As obvious, $L_{\text{union}}(\Gamma) = L(H) \cup \overline{w} = L(H)$. □

Theorem 3.20. For every matrix grammar H, there is a 2-GGR Γ such that $L(H) = L_{\text{conc}}(\Gamma)$.

Proof. We use Algorithm 4 with the matrix grammar H and $\overline{w} = \varepsilon$ as input. We prove that $L(H) = L_{\text{conc}}(\Gamma)$. Theorem 4 says $\{w_1|(w_1, \overline{w}) \in 2\text{-}L(\Gamma)\} = L(H)$ and $\{(w_1, w_2)|(w_1, w_2) \in 2\text{-}L(\Gamma), w_2 \neq \overline{w}\} = \emptyset$. $L_{\text{conc}}(\Gamma) = \{w_1 w_2|(w_1, w_2) \in 2\text{-}L(\Gamma)\} = \{w_1 w_2|(w_1, w_2) \in 2\text{-}L(\Gamma), w_2 = \overline{w}\} \cup \{w_1 w_2|(w_1, w_2) \in 2\text{-}L(\Gamma), w_2 \neq \overline{w}\} = \{w_1 \overline{w}|(w_1, \overline{w}) \in 2\text{-}L(\Gamma)\} \cup \emptyset = \{w_1 \overline{w}|(w_1, \overline{w}) \in 2\text{-}L(\Gamma)\} = L(H)$, because $\overline{w} = \varepsilon$. □

Theorem 3.21. For every matrix grammar H, there is a 2-GGR Γ such that $L(H) = L_{\text{first}}(\Gamma)$.

Proof. We use Algorithm 4 with matrix grammar H and any $\overline{w}$ as input. We prove that $L(H) = L_{\text{first}}(\Gamma)$. Theorem 4 says $\{w_1|(w_1, \overline{w}) \in 2\text{-}L(\Gamma)\} = L(H)$ and $\{(w_1, w_2)|(w_1, w_2) \in 2\text{-}L(\Gamma), w_2 \neq \overline{w}\} = \emptyset$. $L_{\text{first}}(\Gamma) = \{w_1|(w_1, w_2) \in n\text{-}L(\Gamma)\} = \{w_1|(w_1, w_2) \in 2\text{-}L(\Gamma), w_2 = \overline{w}\} \cup \{w_1|(w_1, w_2) \in 2\text{-}L(\Gamma), w_2 \neq \overline{w}\} = \{w_1|(w_1, \overline{w}) \in 2\text{-}L(\Gamma)\} \cup \emptyset = \{w_1|(w_1, \overline{w}) \in 2\text{-}L(\Gamma)\} = L(H)$. □

4. CONCLUSION

Let $L_{GGR_{n,X}}$ denote the language families defined by n-GGR in the X mode, where $X \in$ union, conc, first, let L_H denotes the family of languages generated by the matrix grammars. From the previous results, we obtain:

$$L_H = L_{GGR_{n,X}},\ n \geq 2,\ X \in \{\text{union}, \text{conc}, \text{first}\}.$$

To summarize all the results, multigenerative grammar systems with any number of grammatical components are equivalent with two-component versions of these systems. Perhaps even more importantly, these systems are equivalent with matrix grammars, which generate a proper subfamily of the family of recursively enumerable languages (see [7]). Consequently, the general versions of multigenerative grammar systems are less powerful than their leftmost versions, which characterize the family of recursively enumerable languages (see [14]).

ACKNOWLEDGEMENT

This work was supported by the Czech Science Foundation under grant 201/07/0005 and the MSM 0021630528 grant of the Ministry of Education, Youth and Sports of the Czech Republic. The authors thank the referee of this paper and Jiří Koutný for their helpful comments and suggestions.

(Received February 4, 2008)

REFERENCES

[1] E. Csuhaj-Varju, J. Dassow, J. Kelemen, and Ch. Păun: Grammar Systems: A Grammatical Approach to Distribution and Cooperation. Gordon and Breach, London 1994.

[2] E. Csuhaj-Varju and G. Vaszil: On context-free parallel communicating grammar systems: Synchronization, communication, and normal forms. Theoret. Comput. Sci. *255* (2001), 511–538.

[3] E. Csuhaj-Varju and G. Vaszil: Parallel communicating grammar systems with incomplete information communication. Develop. Language Theory (2001), 381–392.

[4] J. Dassow: On cooperating distributed grammar systems with competence based start and stop conditions. Fund. Inform. *76* (2007), 293–304.

[5] J. Dassow and G. Păun: Regulated Rewriting in Formal Language Theory. Springer-Verlag, New York 1989.

[6] J. Dassow, G. Păun, and G. Rozenberg: Grammar systems. In: Handbook of Formal Languages (G. Rozenberg and A. Salomaa, eds.), Springer, Berlin 1997.

[7] J. Dassow, G. Păun, and A. Salomaa: Grammars with controlled derivations. In: Handbook of Formal Languages (G. Rozenberg and A. Salomaa, eds.), Springer, Berlin (1997).

[8] H. Fernau: Parallel communicating grammar systems with terminal transmission. Acta Inform. *37* (2001), 511–540.

[9] H. Fernau and M. Holzer: Graph-controlled cooperating distributed grammar systems with singleton components. In: Proc. Third Internat. Workshop on Descriptional Complexity of Automata, Grammars, and Related Structures, Vienna 2001, pp. 79–90.

[10] J. Gaso and M. Nehez: Stochastic cooperative distributed grammar systems and random graphs. Acta Inform. *39* (2003), 119–140.

[11] M. A. Harrison: Introduction to Formal Language Theory. Addison-Wesley, London 1978.

[12] A. Meduna: Automata and Languages: Theory and Applications. Springer, London 2000.

[13] A. Meduna: Two-way metalinear PC grammar systems and their descriptional complexity. Acta Cybernet. *16* (2003), 126–137.

[14] A. Meduna and R. Lukas: Multigenerative grammar systems. Schedae Inform. *15* (2006), 175–188.

[15] G. Păun, A. Salomaa, and S. Vicolov: On the generative capacity of parallel communicating grammar systems. Internat. J. Comput. Math. *45* (1992), 45–59.

[16] G. Rozenberg and A. Salomaa, eds.: Handbook of Formal Languages. Springer, Berlin 1997.

[17] A. Salomaa: Formal Languages. Academic Press, New York 1973.

[18] G. Vaszil: On simulating non-returning PC grammar systems with returning systems. Theoret. Comput. Sci. *209* (1998), 1–2, 319–329.

Roman Lukáš, Brno University of Technology, Faculty of Information Technology, Department of Information Systems, Božetěchova 2, 612 66 Brno. Czech Republic.
e-mail: lukas@fit.vutbr.cz

Alexander Meduna, Brno University of Technology, Faculty of Information Technology, Department of Information Systems, Božetěchova 2, 612 66 Brno. Czech Republic.
e-mail: meduna@fit.vutbr.cz