Excel@FIT 2015
http://excel.fit.vutbr.cz

Generative Power of Simple Matrix Grammars Depending on the Number of Their Components

Ondřej Soukup*

Abstract
The concept of simple matrix grammars was introduced and first studied in early seventies. All the achieved results were summarized in a few following studies and the simple matrix grammars start disappearing from the forefront, despite their concept may be still actual. This work returns to them and aims to correct some generally accepted historical mistakes. In the following studies, the very first definition of simple matrix grammars has slightly evolved to the modern and currently generally accepted form, nevertheless, the validity of the former related results were not revised. However, these modifications have significant influence on the simple matrix grammars. Despite the existing beliefs that the simple matrix grammars define an infinite hierarchy of languages depending on the number of their components, we show that two components are exactly strong enough and the addition of another component does not increase the generative power. Moreover, we prove that simple matrix grammars with two components are precisely as strong as matrix grammars.

Keywords: simple matrix grammars — generative power — number of components

Supplementary Material: *N/A*

*isoukup@fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Simple matrix grammars were introduced in [1] by O. H. Ibarra and simultaneously in [2] as tuple grammars by W. Kuich and H. A. Maurer in 1970. Several studies on their generative power and closure properties were published during the next years, for instance [3, 4], or on their leftmost variants, which were introduced in [5] by H. A. Maurer. Later on, they started disappearing from the forefront. However, since the concept of simple matrix grammars is in essence nothing but a series of context-free grammars working in parallel, it might be still actual.

The very first definitions of simple matrix grammars were slightly different than the following ones (for instance see [1] and [5]), which are now generally accepted as simple matrix grammars. Especially, the restriction demanding the constantly equal numbers of nonterminal symbols in all components was not required later on. This could obviously significantly influence the properties of simple matrix grammars, however, some former and crucial results were not revised. Afterwards, they were generally accepted, since simple matrix grammars did not play a major role in theoretical computer science, leaving an unpaid debt behind. This study aims to fill in this historical gab bringing updated results on the generative power of simple matrix grammars.

First and primarily, we prove the previous conclusions about the hierarchy of languages depending on the number of components of simple matrix grammars do not hold. Introducing more than two components does not increase the generative power of simple matrix grammars. Then, we examine their generative power more precisely with respect to Chomsky hierarchy of languages. Simple matrix grammar with a single component is in fact just context-free grammar. However, the addition of the second component significantly increases the generative power. We prove, simple matrix grammars with two components characterize precisely the family of matrix languages.

2. Preliminaries

We assume that the reader is familiar with formal language theory (see [6, 7]). For a set W, $\mathrm{card}(W)$ denotes its cardinality. Let V be an alphabet. V^* is the set of all strings over V. Algebraically, V^* represents the free monoid generated by V under the operation of concatenation. The unit of V^* is denoted by ε. The alphabet of w, denoted by $\mathrm{alph}(w)$, is the set of symbols appearing in w.

Let $\Rightarrow$ be a relation over V^*. The kth power, transitive and transitive and reflexive closure of $\Rightarrow$ are denoted $\Rightarrow^k$, $\Rightarrow^+$ and $\Rightarrow^*$, respectively. Unless explicitly stated otherwise, we write $x \Rightarrow y$ instead $(x,y) \in \Rightarrow$.

A *context-free grammar* (CFG for short) G is a quadruple $G = (N,T,P,S)$, where N is an alphabet of nonterminal symbols, T is an alphabet of terminal symbols, where $N \cap T = \emptyset$, P is a finite set of rules of the form $l : A \rightarrow w$, where l is a unique label representing the rule, $A \in N$ and $w \in (N \cup T)^*$, and $S \in N$ is the starting nonterminal. For rule $l : A \rightarrow w$ and uAv, uwv, where $A \in N$, $u,v,w \in (N \cup T)^*$, $uAv \Rightarrow uwv[l]$ or simply $uAv \Rightarrow uwv$ is the relation of the direct derivation. Then, transitive and transitive and reflexive closures are defined as usual and denoted as stated above. For CFG G, $\mathscr{L}(G) = \{w \mid w \in T^*, S \Rightarrow^* w\}$ denotes the language generated by G.

The families of context-free and context-sensitive languages are denoted by **CF** and **CS**, respectively.

3. Definitions

In this section, we define matrix grammars and simple matrix grammars.

Definition 1. *A* matrix grammar *(MG for short) is a pair*

$$H = (G,M)$$

where

- $G = (N,T,P,S)$ *is a context-free grammar;*
- M *is a finite language over the alphabet of rules* $(M \subseteq P^*)$.

For $x,y \in (N \cup T)^*$, $m \in M$,

$$x \Rightarrow y[m]$$

in H, *if and only if there are* x_0, x_1, $\dots$, x_n *such that* $x_0 = x$, $x_n = y$, *and*

- $x_0 \Rightarrow x_1[p_1] \Rightarrow x_2[p_2] \Rightarrow \dots \Rightarrow x_n[p_n]$ *in* G, *and*
- $m = p_1p_2\dots p_n$, *where* $p_i \in P$, $1 \leq i \leq n$, *for some* $n \geq 1$.

Then,

$$\mathscr{L}(H) = \{x \in T^* \mid S \Rightarrow^* x\}$$

is the language generated by H. *Let* **MT** *denotes the* family of languages generated by matrix grammars.

Definition 2. *Let* $n \geq 1$. *A* simple matrix grammar of degree n *(${}_n$SMG for short) is an (n+3)-tuple*

$$G_n = (N_1, N_2, \dots, N_n, \Sigma, P, S), \textit{where}$$

1. $N_1, \dots, N_n$ *are finite nonempty pairwise disjoint sets of nonterminal symbols;*
2. Σ *is a finite nonempty set of terminal symbols,* $\Sigma \cap N_i = \emptyset$, *for* $1 \leq i \leq n$;
3. S *is called the* start symbol *such that* $S \notin N_1 \cup \dots \cup N_n \cup \Sigma$;
4. P *is a finite set of rewriting rules of the form:*
   (a) $(S) \rightarrow (v)$, $v \in \Sigma^*$.
   (b) $(S) \rightarrow (v_1v_2\dots v_n)$, $v_i \in (N_i \cup \Sigma)^*$, $\mathrm{alph}(v_i) \cap N_i \neq \emptyset$, *for* $1 \leq i \leq n$.
   (c) $(A_1, A_2, \dots, A_n) \rightarrow (v_1, v_2, \dots, v_n)$, $A_i \in N_i$, $v_i \in (N_i \cup \Sigma)^*$, *for* $1 \leq i \leq n$.

For $x,y \in (N_1 \cup \Sigma)^*(N_2 \cup \Sigma)^* \dots (N_n \cup \Sigma)^*$, $p \in P$,

$$x \Rightarrow y[p]$$

in G_n, *if and only if either* $x = S$ *and* p *satisfies* 4a *or* 4b *or* $p : (A_1, A_2, \dots, A_n) \rightarrow (v_1, v_2, \dots, v_n)$ *satisfies* 4c, $x = u_1A_1w_1 \dots u_nA_nw_n$ *and* $y = u_1v_1w_1 \dots u_nv_nw_n$, $u_i, w_i \in (N_i \cup \Sigma)$, *for* $1 \leq i \leq n$. *Then,*

$$\mathscr{L}(G_n) = \{x \mid x \in \Sigma^*, S \Rightarrow^* x\}$$

is said to be the language that G_n generates. *Set*

$${}_n\mathbf{SM} = \{\mathscr{L}(G_n) \mid G_n \textit{ is a } {}_n\textit{SMG}\}$$

${}_n\mathbf{SM}$ *is said to be the* language family that ${}_n$*SMG*s generate. *Let* **SM** *denotes the* family of languages generated by all simple matrix grammars.

4. Results

In this section, we investigate the generative power of simple matrix grammars. We pay a special attention to the number of their components.

Since any context-free grammar is, in fact, ${}_1$*SMG*, we obtain the next theorem.

Theorem 1. ${}_1\mathbf{SM} = \mathbf{CF}$.

In [2] and [1], it was proved, that simple matrix grammars define the infinite hierarchy of languages between the family of context-free and the family of context-sensitive languages based on the number of their components. More precisely, it has been proved

$$\mathbf{CF} = {}_{1}\mathbf{SM} \subset {}_{2}\mathbf{SM} \subset {}_{3}\mathbf{SM} \subset \cdots \subset \mathbf{CS}$$

However, we dispute the results on infinite hierarchy. We disprove it by proving the following theorem.

Theorem 2. ${}_{n}\mathbf{SM} = {}_{2}\mathbf{SM}$, *for* $n \geq 2$.

Proof. Construction. Let $G = (N_1, N_2, \ldots, N_n, \Sigma, P, S)$ be any ${}_{n}SMG$, for some $n \geq 2$. Suppose, the rules in P are in the form

$$r : (A_1, A_2, \ldots, A_n) \to (w_1, w_2, \ldots, w_n)$$

where $1 \leq r \leq \text{card}(P)$ is the unique numeric label. Set $N = N_1 \cup N_2 \cup \cdots \cup N_n$. Define ${}_{2}SMG$ $G' = (N, N', \Sigma, P', S)$. $N' = \{Q^i_j \mid 1 \leq i \leq \text{card}(P), 1 \leq j \leq n\}$. Construct P' as follows. Initially set $P' = \emptyset$. Perform 1 through 3, given next:

1. for each $r : (S) \to (w) \in P$, $w \in \Sigma^*$, add $(S) \to (w)$ to P';
2. for each $r : (S) \to (w) \in P$, $\text{alph}(w) - \Sigma \neq \emptyset$, add $(S) \to (wQ^t_1)$, where $t \in \{1, 2, \ldots, \text{card}(P)\}$, to P';
3. for each $r : (A_1, A_2, \ldots, A_n) \to (w_1, w_2, \ldots, w_n) \in P$, add
   (a) $(A_i, Q^r_i) \to (w_i, Q^r_{i+1})$, for $1 \leq i < n$,
   (b) $(A_n, Q^r_n) \to (w_n, Q^t_1)$, for $t \in \{1, 2, \ldots, \text{card}(P)\}$,
   (c) $(A_n, Q^r_n) \to (w_n, \varepsilon)$, to P'.

Claim 1. $\mathscr{L}(G) = \mathscr{L}(G')$.

Proof. We establish the proof by proving the following two claims.

Claim 2. $\mathscr{L}(G) \subseteq \mathscr{L}(G')$.

Proof. To prove the claim, we show that for any sequence of derivation steps of G, generating the sentence form w, there is the corresponding sequence of derivation steps of G' generating the corresponding sentence form or the same terminal string, if it is successful, by induction on k—the number of the derivation steps of G.

Basis. Let $k = 0$. The correspondence of the sentence forms of G and G' is trivial. Let $k = 1$. Then, some starting rule from P of the form $(S) \Rightarrow (v)$ was applied. However, in P', there is the corresponding rule $(S) \Rightarrow (v)$, in the case the derivation is finished, or $(S) \Rightarrow (vQ^r_1)$, for $1 \leq r \leq \text{card}(P)$, otherwise. Without any loss of generality, suppose r is the label of the next rule applied by G. The basis holds.

Induction Hypothesis. Suppose that there exists $k \geq 1$ such that the assumption of correspondence holds for all sequences of the derivation steps of G of the length m, where $0 \leq m \leq k$.

Induction Step. Consider any sequence of moves

$$S \Rightarrow^{k+1} w$$

Since $k + 1 \geq 1$, this sequence can be written in as

$$S \Rightarrow^k v_1 A_1 u_1 \ldots v_n A_n u_n \Rightarrow v_1 x_1 u_1 \ldots v_n x_n u_n$$

where $A_i \in N_i$, $v_i, u_i \in (N_i \cup \Sigma)^*$ and the last derivation step was performed by some rule

$$r : (A_1, \ldots, A_n) \to (x_1, \ldots, x_n) \in P$$

Then, there exits a sequence of derivation steps of G'

$$S \Rightarrow^k v_1 A_1 u_1 \ldots v_n A_n u_n Q^r_1$$

From the construction of G', there exist rules

$$(A_1, Q^r_1) \to (x_1, Q^r_2), (A_2, Q^r_2) \to (x_2, Q^r_3), \ldots, \\ (A_{n-1}, Q^r_{n-1}) \to (x_{n-1}, Q^r_n), \\ (A_n, Q^r_n) \to (x_n, Q^t_1), (A_n, Q^r_n) \to (x_n, \varepsilon)$$

for $1 \leq t \leq \text{card}(P)$. Therefore, in G', there is the sequence of derivation steps

$$\begin{aligned} S &\Rightarrow^* v_1 A_1 u_1 v_2 A_2 u_2 \ldots v_n A_n u_n Q^r_1 \\ &\Rightarrow v_1 x_1 u_1 v_2 A_2 u_2 \ldots v_n A_n u_n Q^r_2 \\ &\Rightarrow v_1 x_1 u_1 v_2 x_2 u_2 \ldots v_n A_n u_n Q^r_3 \\ &\Rightarrow^* v_1 x_1 u_1 v_2 x_2 u_2 \ldots v_{n-1} x_{n-1} u_{n-1} v_n A_n u_n Q^r_n \end{aligned}$$

Next, there are two possible situations.

First, suppose the derivation of G is not finished and the rule labeled by some t is applied next. Then,

$$v_1 x_1 u_1 \ldots v_n A_n u_n Q^r_n \Rightarrow v_1 x_1 u_1 \ldots v_n x_n u_n Q^t_1$$

in G', generating the sentence form corresponding to the new sentence form of G, where

$$v_1 x_1 u_1 v_2 x_2 u_2 \ldots v_n x_n u_n Q^t_1 = w Q^t_1$$

Second, suppose $w \in \Sigma^*$. By the last of the highlighted rules of G'

$$v_1 x_1 u_1 v_2 x_2 u_2 \ldots v_n A_n u_n Q^r_n \Rightarrow v_1 x_1 u_1 v_2 x_2 u_2 \ldots v_n x_n u_n$$

where $v_1 x_1 u_1 v_2 x_2 u_2 \ldots v_n x_n u_n = w$ and the claim holds. □

Claim 3. $\mathscr{L}(G) \supseteq \mathscr{L}(G')$.

Proof. To prove the claim, we show that for any sequence of derivation steps of G', generating the sentence form w, there is the corresponding sequence of derivation steps of G generating the corresponding sentence form or the same terminal string, if it is successful, by induction on k—the number of the derivation steps of G'.

First, assume that in G' the symbols Q^x_y of the second component serve as a states of computation. With the application of the starting rule some symbol Q^r_1 is inserted. It means, G' is about to simulate the application of the rule r of G. The only applicable rule is then the rule rewriting Q^r_1 to Q^r_2. And it holds equally for any Q^x_y, where $y < n$. Therefore, except the starting rules, these rules have always to be applied in sequence. We use this assumption in the next proof.

Basis. Let $k = 0$. The sentence forms of G and G' correspond. Let $k = 1$. Then, some starting rule from P' of the form $(S) \to (v)$ or $(S) \to (vQ^r_1)$, for some $r \in \{1, 2, \dots, \text{card}(P)\}$ was applied. However, this rule was introduced by the rule $(S) \to (v)$ of P. Therefore G can make the derivation step corresponding to the one made by G' and the basis holds.

Induction Hypothesis. Suppose that there exists $k \geq 1$ such that the assumption of correspondence holds for all sequences of the derivation steps of G' of the length m, where $0 \leq m \leq k$.

Induction Step. Recall the previous assumption. Let us consider any sequence of moves

$$S \Rightarrow^{k+n} w$$

This sequence can be written in the form

$$\begin{aligned} S &\Rightarrow^* v_1A_1u_1v_2A_2u_2 \dots v_nA_nu_nQ^r_1 \\ &\Rightarrow v_1x_1u_1v_2A_2u_2 \dots v_nA_nu_nQ^r_2 \\ &\Rightarrow v_1x_1u_1v_2x_2u_2 \dots v_nA_nu_nQ^r_3 \\ &\dots \\ &\Rightarrow^* v_1x_1u_1v_2x_2u_2 \dots v_nx_nu_nX \end{aligned}$$

where $A_i \in N_i$, $v_i, u_i \in (N_i \cup \Sigma)^*$ and $X = Q^t_1$, for $1 \leq t \leq \text{card}(P)$, or $X = \varepsilon$. There were n rules from P' used

$$(A_1, Q^r_1) \to (x_1, Q^r_2), (A_2, Q^r_2) \to (x_2, Q^r_3), \dots, \\ (A_n, Q^r_n) \to (x_n, X)$$

These rules were introduced by the rule of G

$$r : (A_1, A_2, \dots, A_n) \to (x_1, x_2, \dots, x_n) \in P$$

By the induction hypothesis, in the terms of G

$$S \Rightarrow^* v_1A_1u_1v_2A_2u_2 \dots v_nA_nu_n$$

Thus using the rule r

$$v_1A_1u_1 \dots v_nA_nu_n \Rightarrow v_1x_1u_1 \dots v_nx_nu_n = w$$

If $X = \varepsilon$, $w \in \Sigma^*$ and G' generates the terminal string, G generates the same terminal string and the claim holds.

□

We proved $\mathscr{L}(G) \subseteq \mathscr{L}(G')$ and $\mathscr{L}(G) \supseteq \mathscr{L}(G')$, therefore $\mathscr{L}(G) = \mathscr{L}(G')$.

□

Since G is any $_nSMG$, for some $n \geq 2$, and G' is $_2SMG$, Theorem 2 holds.

□

We have proved

$$\mathbf{CF} = {}_\mathbf{1}\mathbf{SM} \subseteq {}_\mathbf{2}\mathbf{SM} = \mathbf{SM} \subset \mathbf{CS}$$

however, this classification of simple matrix languages is not very precise. Let us recall (see [8]) that

$$\mathbf{CF} \subset \mathbf{MT} \subset \mathbf{CS}$$

Next, we prove the following theorem.

Theorem 3. **SM** = **MT**.

Proof. Without any loss of generality, we consider only $_2SMG$s.

Claim 4. $_\mathbf{2}\mathbf{SM} \subseteq \mathbf{MT}$.

Proof. Construction. Let $G = (N_1, N_2, \Sigma, P, S)$ be any $_2SMG$. Suppose that the rules in P are of the form

$$r : (A_1, A_2) \to (w_1, w_2)$$

where $1 \leq r \leq \text{card}(P)$ is a unique numerical label. Define $MG\ H = (G', M)$, where $G' = (N_1 \cup N_2 \cup \{S\}$, Σ, P', $S)$. Construct P' and M as follows. Initially set $P' = M = \emptyset$. Perform 1 and 2, given next:

1. for each $r : (S) \to (w) \in P$,
   (a) add $r : S \to w$ to P',
   (b) add r to M;
2. for each $r : (A_1, A_2) \to (w_1, w_2) \in P$,
   (a) add $r_1 : A_1 \to w_1$ and
   (b) add $r_2 : A_2 \to w_2$ to P',
   (c) add r_1r_2 to M.

Claim 5. $\mathscr{L}(G) = \mathscr{L}(H)$.

Proof. Basic idea. Every ${}_2SMG$ first applies some starting rule. Next, in every derivation step simultaneously in each component there is rewritten a nonterminal to some string. This process can be simulated by MG as follows.

For every starting rule of G, there is the same and independently applicable rule of H. They both generate the equal terminal strings or a sentence forms w_1w_2, where $\text{alph}(w_1)\cap N_2 = \emptyset$ and $\text{alph}(w_2)\cap N_1 = \emptyset$. Then, for every rule $(A_1,A_2) \to (x_1,x_2)$ in G, there are two rules $A_1 \to x_1$ and $A_2 \to x_2$ in H, which must be applied consecutively. Since $N_1 \cap N_2 = \emptyset$, every nonterminal sentence form remains of the form w_1w_2, where $\text{alph}(w_1)\cap N_2 = \emptyset$ and $\text{alph}(w_2)\cap N_1 = \emptyset$. Thus, H simulates both separated components of G.

A detailed version of the proof is left to the reader. □

Since for any ${}_2SMG$ we can construct an MG generating the same language, Claim 4 holds.

□

Claim 6. ${}_2\mathbf{SM} \supseteq \mathbf{MT}$.

Proof. Construction. Let $H = (G,M)$, where $G = (N,\Sigma,P,S)$, $S \in N$, be any MG. Without any loss of generality, suppose every $m \in M$ has its unique label. Define ${}_2SMG$ $G' = (N,N',\Sigma,P',S')$, where $S' \notin N$,

$$N' = \{Q_i^r \mid r : p_1p_2 \dots p_k \in M, i \in \{1,2,\dots,k\}, k \geq 1\}.$$

Construct P' as follows. Initially set $P' = \emptyset$. Perform 1 given next:

1. for each $r : p_1p_2 \dots p_k \in M$, $k \geq 1$, where $p_i : A_i \to w_i$, $i \in \{1,2,\dots,k\}$,
   (a) add $(S') \to (SQ_1^r)$,
   (b) $(A_i,Q_i^r) \to (w_i,Q_{i+1}^r)$, for $i < k$,
   (c) $(A_k,Q_k^r) \to (w_k,Q_1^t)$, for all $t : v \in M$,
   (d) $(A_k,Q_k^r) \to (w_k,\varepsilon)$ to P'.

Claim 7. $\mathscr{L}(G) = \mathscr{L}(H)$.

Proof. We establish Claim 7 by proving the following two claims.

Claim 8. $\mathscr{L}(G) \supseteq \mathscr{L}(H)$.

Proof. We prove the claim by the induction on the number of derivation steps k.

Basis. Let $k = 0$. By the application of the rule $(S') \to (SQ_1^r)$, G' generates the sentence form corresponding to the starting sentence form of H. Without any loss of generality, suppose r is the label of the next matrix applied by H. The basis holds.

Induction Hypothesis. Suppose that there exists $k \geq 0$ such that the claim holds for all sequences of the derivation steps of the length m, where $0 \leq m \leq k$.

Induction Step. Consider any sequence of moves

$$S \Rightarrow^{k+1} w'$$

where $w' \in (N \cup \Sigma)^*$. Since $k+1 \geq 1$,

$$S \Rightarrow^k w \Rightarrow w'$$

for some $w \in (N \cup \Sigma)^*$. Then, the last derivation step was performed by some matrix

$$r : p_1p_2 \dots p_n$$

where $n \geq 1$. By the induction hypothesis

$$S' \Rightarrow SQ_1^s \Rightarrow^* wQ_1^r$$

in G', for some s. By the construction of G', there exists the sequence of rules corresponding to the matrix r

$$(A_1,Q_1^r) \to (w_1,Q_2^r), \dots, (A_n,Q_n^r) \to (w_n,X)$$

where $X = Q_1^t$, where t is the next matrix applied by H, if $\text{alph}(w')\cap N \neq \emptyset$, or $X = \varepsilon$ otherwise. By the application of this sequence of the rules

$$wQ_1^r \Rightarrow^* w'X,$$

which completes the proof.

□

Claim 9. $\mathscr{L}(G) \subseteq \mathscr{L}(H)$.

Proof. We prove the claim by the induction on the number of derivation steps k.

Basis. Let $k = 1$. Then, G' applied some starting rule $(S') \to (SQ_1^r)$. The resulting sentence form corresponds to the starting sentence form of H. The basis holds.

Induction Hypothesis. Suppose that there exists $k \geq 1$ such that the claim holds for all sequences of the derivation steps of the length m, where $1 \leq m \leq k$.

Induction Step. Notice that the rules of G', except the starting ones, form sequences

$$(A_1,Q_1^r) \to (w_1,Q_2^r), \dots, (A_n,Q_n^r) \to (w_n,X)$$

where $X = Q_1^t, \varepsilon$, for some t. These sequences must be always fully applied. Therefore, consider any sequence of moves

$$S' \Rightarrow S \Rightarrow^{k+l} w'$$

where $w' \in (N \cup \Sigma)^*$ and $l \geq 1$ is the length of any such sequence of the rules, which was applied last. Then,

$$S' \Rightarrow S \Rightarrow^k w \Rightarrow^l w'$$

The used sequence of the rules was introduced by some matrix $r : v \in M$. By the induction hypothesis

$$S \Rightarrow^* w$$

in H. However, by the matrix r

$$w \Rightarrow w'[r]$$

which completes the proof.

□

Since $\mathscr{L}(G) \supseteq \mathscr{L}(H)$ and $\mathscr{L}(G) \subseteq \mathscr{L}(H)$, $\mathscr{L}(G) = \mathscr{L}(H)$.

□

Since for any MG we can construct a $_2SMG$ generating the same language, Claim 6 holds.

□

On the basis of Claim 4 and Claim 6 Theorem 3 holds.

□

5. Conclusion

Despite the concept of simple matrix grammars is nothing new in the field of theoretical computer science, we showed that there was still serious outstanding historical debt, which we aimed to repay. Let us state all the achieved results.

$$\mathbf{CF} = {}_1\mathbf{SM} \subset {}_2\mathbf{SM} = \mathbf{SM} = \mathbf{MT} \subset \mathbf{CS}$$

We proved, simple matrix grammars with two components are exactly as strong as matrix grammars. Additionally, we disprove mistaken belief that they form an infinite hierarchy of languages based on the number of their components proving that a presence of more than two components has no influence on their generative power.

This paper is a part of a larger study, which additionally focuses on several types of leftmost derivations of simple matrix grammars showing their significant influence on the generative power, which is currently being reviewed in *Semigroups, Languages and Algebras* journal.

Acknowledgments

This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070), the TAČR grant TE01020415, and the BUT grant FIT-S-14-2299.

Significant thanks belong to my supervisor professor Alexander Meduna for his professional guidance, support and many pieces of advice.

References

[1] O. H. Ibarra. Simple matrix languages. *Information and Control*, 17:359–394, 1970.

[2] W. Kuich and H. A. Maurer. Tuple languages. In *International Computing Symposium, Bonn*, pages 881–891, 1970.

[3] W. Kuich and H. A. Maurer. On the inherent ambiguity of simple tuple languages. *Computing*, 7:194–203, 1971.

[4] W. Kuich and H. A. Maurer. The structure generating function and entropy of tuple languages. *Information and Control*, Vol. 19(No. 3):194–203, October 1971.

[5] H. A. Maurer. Simple matrix languages with a leftmost restriction. *Information and Control*, 23:128–139, 1973.

[6] A. Salomaa. *Formal Languages*. Academic Press, London, 1973.

[7] G. Rozenberg and A. Salomaa, editors. *Handbook of Formal Languages, Vol. 1: Word, Language, Grammar*. Springer, New York, 1997.

[8] J. Dassow and G. Păun, editors. *Regulated Rewriting in Formal Language Theory*. Akademie-Verlag, Berlin, 1989. ISBN: 978-3-642-74934-6.