Fundamenta Informaticae XX (2008) 1–9
IOS Press

On Descriptional Complexity of Partially Parallel Grammars

Tomáš Masopust
Faculty of Information Technology, Brno University of Technology
Božetěchova 2, Brno 61266, Czech Republic
masopust@fit.vutbr.cz

Alexander Meduna
Faculty of Information Technology, Brno University of Technology
Božetěchova 2, Brno 61266, Czech Republic
meduna@fit.vutbr.cz

Abstract. This paper presents some new results concerning the descriptional complexity of partially parallel grammars. Specifically, it proves that every recursively enumerable language is generated (i) by a four-nonterminal scattered context grammar with no more than four non-context-free productions, (ii) by a two-nonterminal multisequential grammar with no more than two selectors, or (iii) by a three-nonterminal multicontinuous grammar with no more than two selectors.

Keywords: formal languages, scattered context grammars, multisequential grammars, multicontinuous grammars, descriptional complexity.

1. Introduction

The descriptional complexity of formal grammars, which studies how to describe grammars in a reduced and succinct way, has always represented an important investigation area of formal language theory. As a central topic, this investigation of the descriptional complexity studies how to reduce the number of grammatical components, such as the number of nonterminals or productions, to obtain the most economical description of languages generated by the given type of grammars. Consult [5, 6] for the results concerning the descriptional complexity of context-free grammars and languages with respect to the number of nonterminals, and [8] and the citations therein for an overview of results concerning the

Address for correspondence: Faculty of Information Technology, Brno University of Technology, Božetěchova 2, Brno 61266, Czech Republic

descriptional complexity of partially parallel grammars and grammars regulated by context conditions with respect to both the number of nonterminals and the number of non-context-free productions (or selectors). Useful information can also be found in the proceedings of the international workshop series *Descriptional Complexity of Formal Systems*, which is held annually (see [3] for its latest edition).

This paper concentrates its investigation on reducing of partially parallel grammars with respect to the number of nonterminals and non-context-free productions. It studies how to achieve this reduction without any modification of their generative power, which coincides with the power of Turing machines. By achieving this reduction, it actually makes the partially parallel rewriting more succinct and economical, and this economization is obviously highly appreciated both from a practical and theoretical standpoint. More specifically, two types of partially parallel context-free grammars are central to this paper—scattered context grammars (see [4]) and multirewriting grammars (see [7]). *Scattered context grammars* are based on sequences of context-free productions, by which these grammars simultaneously rewrite several nonterminals during a derivation step. *Multirewriting grammars* are underlaid by grammars that use context-free-like productions that have a terminal or a nonterminal on their left-hand sides. By using extremely simple regular languages, called *selectors*, they specify sequences of symbols that are rewritten during a derivation step and, in addition, place some slight restrictions on the context appearing between the rewritten symbols. Otherwise, they work by analogy with context-free grammars.

Let us note that we disallow *pseudoterminals*—that is, terminals that do not appear in the generated languages—in any of the grammars under investigation because their existence would significantly simplify the achievement of most results. Specifically, we could obtain the results concerning multirewriting grammars by using selectors in a trivial way as follows from their definition given in Section 2.

It is well-known that every recursively enumerable language is generated by a three-nonterminal scattered context grammar with an unlimited number of non-context-free productions (see [11]), by a five-nonterminal scattered context grammar with no more than two non-context-free productions (see [14]), by a six-nonterminal multisequential grammar (see [9]), or by a six-nonterminal multicontinuous grammar (see [10]). An overview of the previous results can also be found in [12].

This paper presents several new results concerning the descriptional complexity of partially parallel grammars. Specifically, it proves that every recursively enumerable language is generated (i) by a four-nonterminal scattered context grammar with no more than four non-context-free productions, (ii) by a two-nonterminal multisequential grammar with no more than two selectors, or (iii) by a three-nonterminal multicontinuous grammar with no more than two selectors. Result (i) improves the previous result with respect to the number of nonterminals keeping the number of non-context-free productions limited; on the other hand, it increases the number of non-context-free productions. Finally, results (ii) and (iii) improve the previous results with respect to the number of both nonterminals and non-context-free productions.

2. Preliminaries

We assume that the reader is familiar with formal language theory (see [1, 13]). For an alphabet (finite nonempty set) V, V^* represents the free monoid generated by V. The unit of V^* is denoted by ε. Set $V^+ = V^* - \{\varepsilon\}$. For $w \in V^*$, $|w|$ and w^R denote the length and the mirror image of w, respectively. Let $w \in V^*$. Then, $alph(w) = \{a \in V : a \text{ appears in } w\}$ and for $L \subseteq V^*$, $alph(L) = \bigcup_{w \in L} alph(w)$.

A *scattered context grammar* is a quadruple $G = (N, T, P, S)$, where N is a nonterminal alphabet, T is a terminal alphabet such that $N \cap T = \emptyset$, $V = N \cup T$, $S \in N$ is the start symbol, and P is a finite set of productions such that each production p has the form $(A_1, A_2, \ldots, A_n) \to (x_1, x_2, \ldots, x_n)$, for some $n \geq 1$, where $A_i \in N$ and $x_i \in V^*$, for $i = 1, \ldots, n$. Set $\pi(p) = n$. If $\pi(p) \geq 2$, then p is said to be a *non-context-free* production. If $\pi(p) = 1$, then p is said to be *context-free*. For $u, v \in V^*$, $u \Rightarrow v$ provided that

1. $u = u_1 A_1 u_2 A_2 u_3 \ldots u_n A_n u_{n+1}$,
2. $v = u_1 x_1 u_2 x_2 u_3 \ldots u_n x_n u_{n+1}$, and
3. $(A_1, A_2, \ldots, A_n) \to (x_1, x_2, \ldots, x_n) \in P$,

where $u_i \in (N \cup T)^*$, for $i = 1, \ldots, n$. The language of G is defined as $L(G) = \{w \in T^* : S \Rightarrow^* w\}$, where $\Rightarrow^*$ denotes the reflexive and transitive closure of $\Rightarrow$.

A *multisequential grammar* is a quintuple $G = (N, T, P, S, K)$, where N is a nonterminal alphabet, T is a terminal alphabet such that $N \cap T = \emptyset$, $V = N \cup T$, $S \in N$ is the start symbol, P is a finite set of productions of the form $a \to x$, where $a \in V$ and $x \in V^*$, and K is a finite set of selectors of the form

$$X_1 \mathbf{act}(Y_1) X_2 \ldots X_n \mathbf{act}(Y_n) X_{n+1} ,$$

where n is a positive integer,

1. $X_i \in \{Z^* : Z \subseteq V\}$, for $i = 1, \ldots, n+1$, and
2. $Y_j \in \{Z : Z \subseteq V,\ Z \neq \emptyset\}$, for $j = 1, \ldots, n$.

For $u, v \in V^*$, $u \Rightarrow v$ provided that

1. $u = u_1 a_1 u_2 a_2 u_3 \ldots u_n a_n u_{n+1}$,
2. $v = u_1 x_1 u_2 x_2 u_3 \ldots u_n x_n u_{n+1}$, and
3. K contains a selector $X_1 \mathbf{act}(Y_1) X_2 \ldots X_n \mathbf{act}(Y_n) X_{n+1}$ such that
   (a) $u_i \in X_i$, for $i = 1, \ldots, n+1$,
   (b) $a_j \in Y_j$ and $a_j \to x_j \in P$, for $j = 1, \ldots, n$.

The language of G is defined as $L(G) = \{w \in T^* : S \Rightarrow^* w\}$, where $\Rightarrow^*$ denotes the reflexive and transitive closure of $\Rightarrow$.

A *multicontinuous grammar* is a quintuple $G = (N, T, P, S, K)$, where N, T, P, and S have the same meaning as in a multisequential grammar, $V = N \cup T$, and K is a finite set of selectors of the form

$$X_1 \mathbf{act}(Y_1) X_2 \ldots X_n \mathbf{act}(Y_n) X_{n+1} ,$$

where n is a positive integer,

1. $X_i \in \{Z^* : Z \subseteq V\}$, for $i = 1, \ldots, n+1$, and
2. $Y_j \in \{Z^+ : Z \subseteq V,\ Z \neq \emptyset\}$, for $j = 1, \ldots, n$.

For every $v \in V^+$, where $v = a_1a_2 \dots a_{|v|}$ with $a_i \in V$, for $i = 1, \dots, |v|$, define the language $ContRewriting(v) \subseteq V^*$ by this equivalence: for every $z \in V^*$, $z \in ContRewriting(v)$ if and only if $a_i \to x_i \in P$, for $i = 1, \dots, |v|$, and $z = x_1x_2 \dots x_{|v|}$. For $u, v \in V^*$, $u \Rightarrow v$ provided that

1. $u = u_1y_1u_2y_2u_3 \dots u_ny_nu_{n+1}$,
2. $v = u_1z_1u_2z_2u_3 \dots u_nz_nu_{n+1}$, and
3. K contains a selector $X_1\mathbf{act}(Y_1)X_2 \dots X_n\mathbf{act}(Y_n)X_{n+1}$ such that
   (a) $u_i \in X_i$, for $i = 1, \dots, n+1$,
   (b) $y_j \in Y_j$ and $z_j \in ContRewriting(y_j)$, for $j = 1, \dots, n$.

As usual, the language of G is defined as $L(G) = \{w \in T^* : S \Rightarrow^* w\}$, where $\Rightarrow^*$ denotes the reflexive and transitive closure of $\Rightarrow$.

3. Main Results

This section presents the main results concerning the descriptional complexity of scattered context grammars, multisequential grammars, and multicontinuous grammars.

3.1. Scattered Context Grammars

Recall that every recursively enumerable language is generated by a grammar, G_1, in the first Geffert normal form, where

$$G_1 = (\{S, A, B, C, D\}, T, P \cup \{AB \to \varepsilon, CD \to \varepsilon\}, S),$$

and P contains only context-free productions (see [2]). Moreover, the context-free productions are of the form

$S \to zSa$,
$S \to uSv$,
$S \to uv$,

where $z, u \in \{A, C\}^*$, $v \in \{B, D\}^*$, and $a \in T$. In addition, any terminal derivation in G_1 is of the form $S \Rightarrow^* w_1w_2w$ by productions from P, where $w_1 \in \{A, C\}^*$, $w_2 \in \{B, D\}^*$, $w \in T^*$, and $w_1w_2w \Rightarrow^* w$ by $AB \to \varepsilon$ and $CD \to \varepsilon$.

Theorem 3.1. Every recursively enumerable language is generated by a scattered context grammar containing four nonterminals and four non-context-free productions.

Proof:
Let L be a recursively enumerable language. Then, there is a grammar $G_1 = (\{S', A, B, C, D\}, T, P' \cup \{AB \to \varepsilon, CD \to \varepsilon\}, S')$ in the first Geffert normal form such that $L(G_1) = L$. Define the homomorphism $h : \{A, B, C, D\}^* \to \{0, 1\}^*$ so that $h(A) = h(B) = 00$, $h(C) = 10$, and $h(D) = 01$. Set $N = \{S, 0, 1, \$\}$. Define the scattered context grammar $G = (N, T, P, S)$ with P constructed as follows:

1. $(S) \rightarrow (h(z)S1a1)$, where $S' \rightarrow zS'a \in P'$;
2. $(S) \rightarrow (h(u)Sh(v))$, where $S' \rightarrow uS'v \in P'$;
3. $(S) \rightarrow (11S)$;
4. $(S) \rightarrow (h(u)\$\$h(v))$, where $S' \rightarrow uv \in P'$;
5. $(\$) \rightarrow (\varepsilon)$;
6. $(0, 0, \$, \$, 0, 0) \rightarrow (\$, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \$)$;
7. $(1, 0, \$, \$, 0, 1) \rightarrow (\$, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \$)$;
8. $(1, 1, \$, \$, 1, 1) \rightarrow (11\$, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \$)$;
9. $(1, 1, \$, \$, 1, 1) \rightarrow (\varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon)$.

To give an insight into the proof, consider a derivation of G_1 that generates a nonempty string. Then, G simulates the derivation of G_1 so that it starts with one application of production 3 and if $S' \rightarrow zS'a$, $S' \rightarrow uS'v$, or $S' \rightarrow uv$ is used in G_1, the corresponding production is used in G, see productions 1, 2, or 4, respectively. In addition, A, B, C, and D are encoded by strings of 0s and 1s of length two. If G_1 uses $AB \rightarrow \varepsilon$ or $CD \rightarrow \varepsilon$, then G uses production 6 or 7, respectively. After removing all the symbols A, B, C, and D, G_1 finishes. At this moment, G starts to remove the symbols 1 that enclose terminal symbols. This is done by production 8, and the derivation finishes by production 9. If the derivation generates an empty string, the applications of productions 3 and 8 are omitted, and production 5 is used instead of production 9. The following formal proof demonstrates that this is the only way G derives terminal strings.

Consider a derivation of the form

$$S' \Rightarrow^* \alpha\beta a_1 a_2 \dots a_n \Rightarrow^* a_1 a_2 \dots a_n\,,$$

where $\alpha \in \{A, C\}^*$, $\beta \in \{B, D\}^*$, $a_i \in T$, for $i = 1, \dots, n$, and neither $AB \rightarrow \varepsilon$ nor $CD \rightarrow \varepsilon$ has been used in $S' \Rightarrow^* \alpha\beta a_1 a_2 \dots a_n$. Moreover, only productions $AB \rightarrow \varepsilon$ or $CD \rightarrow \varepsilon$ have been used in $\alpha\beta a_1 a_2 \dots a_n \Rightarrow^* a_1 a_2 \dots a_n$. If $a_1 a_2 \dots a_n \neq \varepsilon$, then G can derive

$$S \Rightarrow^* 11h(\alpha)\$\$h(\beta)1a_1 11a_2 1 \dots 1a_n 1$$

and, by productions constructed in 6 and 7, eliminate $h(\alpha\beta)$. Thus,

$$S \Rightarrow^* 11\$\$1a_1 11a_2 1 \dots 1a_n 1.$$

By productions constructed in 8 and 9, G eliminates all nonterminals 1 and \$. If $a_1 a_2 \dots a_n = \varepsilon$, then G can derive $S \Rightarrow^* h(\alpha)\$\$h(\beta)$; then, by productions constructed in 6 and 7, G eliminates $h(\alpha\beta)$. Thus, $S \Rightarrow^* \$\$$ in G. By the production constructed in 5, G eliminates both nonterminals \$. Therefore,

$$S' \Rightarrow^* a_1 a_2 \dots a_n \text{ implies } S \Rightarrow^* a_1 a_2 \dots a_n.$$

On the other hand, let

$$S \Rightarrow^* \alpha\$\$\beta \Rightarrow^* a_1 a_2 \dots a_n$$

be a derivation, where $\alpha \in \{00, 01, 11\}^*$, $\beta \in (\{00, 01\} \cup \{1\}T\{1\})^*$, $a_i \in T$, for $i = 1, \dots, n$, and none of the non-context-free productions has been used in $S \Rightarrow^* \alpha\$\$\beta$.

Notice that if a nonterminal occurs between the first \$ and the second \$, then the nonterminal cannot be removed, so the derivation cannot generate a word of terminals.

If $a_1 a_2 \dots a_n = \varepsilon$, then $\beta \in \{00, 01\}^*$, β does not contain 11 as a substring. Therefore, productions constructed in 8 and 9 cannot be used in the derivation. Thus, neither can production 3 be used, so α does not contain 11 as a substring. As the other productions simulate the productions from G_1, $S' \Rightarrow^* \varepsilon$ in G_1.

If $a_1 a_2 \dots a_n \neq \varepsilon$, then $\beta = \beta_1 1 a_1 1 \beta_2$, where $\beta_1 \in \{00, 01\}^*$ and $\beta_2 \in (\{00, 01\} \cup \{1\}T\{1\})^*$. After deleting β_1 by productions constructed in 6 and 7, the production constructed in 8 or 9 has to be used. Therefore, $\alpha = \alpha_2 11 \alpha_1$, where $\alpha_1 = \beta_1^R$ and $\alpha_2 \in \{0, 1\}^*$. Thus,

$$S \Rightarrow^* \alpha\$\$\beta \Rightarrow^* \alpha_2 11\$\$1 a_1 1 \beta_2.$$

We prove that $\alpha_2 = \varepsilon$ and $\beta_2 \in (\{1\}T\{1\})^*$ (by induction on $|\beta_2| \geq 0$). At this point, the only productions that can be used are productions constructed in 8 and 9. By using the production constructed in 9, G makes

$$S \Rightarrow^* \alpha\$\$\beta \Rightarrow^* \alpha_2 11\$\$1 a_1 1 \beta_2 \Rightarrow \alpha_2 a_1 \beta_2.$$

Therefore, $\alpha_2 a_1 \beta_2 \in T^*$ if and only if $\alpha_2 = \beta_2 = \varepsilon$. By using the production constructed in 8, G makes

$$S \Rightarrow^* \alpha\$\$\beta \Rightarrow^* \alpha_2 11\$\$1 a_1 1 \beta_2 \Rightarrow \alpha_2 11\$ a_1 \$\beta_2.$$

Therefore, if $\beta_2 = 00\beta_2'$, the prefix 00 can be removed only by the production constructed in 6. However, after using this production, the substring 11 attached to \$\$ appears between the two \$s, so it cannot be removed after that. The same is true for $\beta_2 = 01\beta_2''$. Thus, $\beta_2 = 1a_2 1\beta_3$. Then, by induction,

$$S \Rightarrow^* \alpha\$\$\beta = 11\gamma^R\$\$\gamma 1 a_1 1 1 a_2 1 \dots 1 a_n 1$$

where $\gamma \in \{00, 01\}^*$. Since $h(A) = h(B) = 00$, $h(C) = 10$, and $h(D) = 01$, we get $S' \Rightarrow^* \delta_1 \delta_2 a_1 a_2 \dots a_n \Rightarrow^* a_1 a_2 \dots a_n$, where $\delta_1 \in \{A, C\}^*$, $\delta_2 \in \{B, D\}^*$, $h(\delta_1) = \gamma^R$, and $h(\delta_2) = \gamma$. $\square$

Notice that Theorem 3.1 is strongly related to Theorem 1 in [14] that demonstrates that every recursively enumerable language can be generated by a scattered context grammar containing five nonterminals and two non-context-free productions.

3.2. Multisequential Grammars

Recall that every recursively enumerable language is generated by a grammar, G_2, in the second Geffert normal form, where

$$G_2 = (\{S, A, B\}, T, P \cup \{ABBBA \to \varepsilon\}, S)\,,$$

and P contains only context-free productions (see [2]). Moreover, the context-free productions are of the form

$S \to zSa$,
$S \to uSv$,
$S \to uv$,

where $z, u \in \{AB, ABB\}^*$, $v \in \{BA, BBA\}^*$, and $a \in T$. In addition, any terminal derivation in G_2 is of the form $S \Rightarrow^* w_1w_2w$ by productions from P, where $w_1 \in \{AB, ABB\}^*$, $w_2 \in \{BA, BBA\}^*$, $w \in T^*$, and $w_1w_2w \Rightarrow^* w$ by $ABBBA \to \varepsilon$.

As there is no more than one substring of the form $ABBBA$ in any sentential form of G_2, the multisequential grammar can activate exactly these symbols.

Lemma 3.1. Every recursively enumerable language is generated by a multisequential grammar containing three nonterminals and two selectors.

Proof:
Let L be a recursively enumerable language and let $G_2 = (\{S, A, B\}, T, P \cup \{ABBBA \to \varepsilon\}, S)$ be a grammar in the second Geffert normal form such that $L(G_2) = L$. Define the multisequential grammar $G = (\{S, A, B\}, T, P \cup \{A \to \varepsilon,\ B \to \varepsilon\}, S, K)$ with K containing these two selectors:

1. $\{A, B\}^*\mathbf{act}(S)(\{A, B\} \cup T)^*$,
2. $\{A, B\}^*\mathbf{act}(A)\mathbf{act}(B)\mathbf{act}(B)\mathbf{act}(B)\mathbf{act}(A)(\{A, B\} \cup T)^*$.

Observe that $L(G) = L(G_2)$. □

Moreover, as T is an alphabet, i.e. a nonempty set, there is a terminal symbol in T. Let $a \in T$ be such a symbol. Then, either of symbols A and B can be encoded by symbols A and a.

Theorem 3.2. Every recursively enumerable language is generated by a multisequential grammar containing two nonterminals and two selectors.

Proof:
Consider the multisequential grammar G constructed in the proof of Lemma 3.1. Let $a \in T$ be a terminal symbol and define the homomorphism $h : (\{S, A, B\} \cup T)^* \to (\{S, A\} \cup T)^*$ as $h(b) = b$, for $b \in T$, $h(S) = S$, $h(A) = aAa$, and $h(B) = aAAa$. Define the multisequential grammar $G' = (\{S, A\}, T, \{S \to h(\alpha) : S \to \alpha \in P\} \cup \{A \to \varepsilon, a \to \varepsilon\}, S, K)$ with K containing these two selectors:

1. $\{A, a\}^*\mathbf{act}(S)(\{A\} \cup T)^*$,
2. $\{A, a\}^*\mathbf{act}(a)\mathbf{act}(A)\mathbf{act}(a)\mathbf{act}(a)\mathbf{act}(A)\mathbf{act}(A)\mathbf{act}(a)$
   $\mathbf{act}(a)\mathbf{act}(A)\mathbf{act}(A)\mathbf{act}(a)\mathbf{act}(a)\mathbf{act}(A)\mathbf{act}(A)\mathbf{act}(a)$
   $\mathbf{act}(a)\mathbf{act}(A)\mathbf{act}(a)(\{A\} \cup T)^*$.

Observe that $S \to \alpha$ is a production in G if and only if $S \to h(\alpha)$ is a production in G'. If $uABBBAv \Rightarrow uv$ in G, where $u \in \{A, B\}^*$ and $v \in \{A, B\}^*T^*$, then $h(u)aAaaAAaaAAaaAAaaAah(v) \Rightarrow h(uv)$ in G' (by selector 2), and vice versa. Hence, the theorem holds. □

3.3. Multicontinuous Grammars

Analogously as in the previous section, we prove the following result.

Theorem 3.3. Every recursively enumerable language is generated by a multicontinuous grammar containing three nonterminals and two selectors.

Proof:
Let L be a recursively enumerable language and let $G_2 = (\{S, A, B\}, T, P \cup \{ABBBA \to \varepsilon\}, S)$ be a grammar in the second Geffert normal form such that $L(G_2) = L$. Let $b \in T$ be a terminal symbol and define the homomorphism $h : (\{S, A, B\} \cup T)^* \to (\{S, X, Y\} \cup T)^*$ as $h(a) = a$, for $a \in T$, $h(S) = S$, $h(A) = XY$, and $h(B) = XbY$. Define the multicontinuous grammar $G = (\{S, X, Y\}, T, \{S \to h(\alpha) : S \to \alpha \in P\} \cup \{X \to \varepsilon, Y \to \varepsilon, b \to \varepsilon\}, S, K)$ with K containing these two selectors:

1. $\{X, Y, b\}^*\mathbf{act}(S^+)(\{X, Y\} \cup T)^*$,
2. $\{X, Y, b\}^*\mathbf{act}(X^+)\mathbf{act}(Y^+)\mathbf{act}(X^+)\mathbf{act}(b^+)\mathbf{act}(Y^+)$
   $\mathbf{act}(X^+)\mathbf{act}(b^+)\mathbf{act}(Y^+)\mathbf{act}(X^+)\mathbf{act}(b^+)\mathbf{act}(Y^+)$
   $\mathbf{act}(X^+)\mathbf{act}(Y^+)(\{X, Y\} \cup T)^*$.

At the beginning of any derivation, only selector 1 is applicable. After eliminating S, the other selector is applicable. Moreover, as there is no more than one substring of the form $h(ABBBA) = XYXbYXbYXbYXY$ in each derivation (see [2]), selector 2 is applicable only on no more than one substring. As there is no occurrence of substrings XX or YY in any derivation, this theorem holds. $\square$

Acknowledgements

Supported by the Czech Ministry of Education under the Research Plan No. MSM 0021630528 and the Czech Grant Agency project No. 201/07/0005.

References

[1] Dassow, J., Păun, G.: *Regulated Rewriting in Formal Language Theory*, Springer-Verlag, Berlin, 1989.

[2] Geffert, V.: Context-Free-Like Forms for the Phrase-Structure Grammars, *MFCS* (M. Chytil, L. Janiga, V. Koubek, Eds.), 324, Springer, 1988.

[3] Geffert, V., Pighizzini, G., Eds.: *Proceedings of the 9th International Workshop on Descriptional Complexity of Formal Systems, DCFS 2007*, High Tatras, Slovakia, July 20 – 22, 2007.

[4] Greibach, S., Hopcroft, J.: Scattered Context Grammars, *Journal of Computer and System Sciences*, **3**, 1969, 233–247.

[5] Gruska, J.: On a Classification of Context-Free Languages, *Kybernetika*, **13**, 1967, 22–29.

[6] Gruska, J.: The Descriptional Complexity of Context-Free Languages, *Proceedings of the 2nd MFCS Conference*, 1973.

[7] Kleijn, H. C. M., Rozenberg, G.: Multi Grammars, *International Journal of Computer Mathematics*, **12**, 1983, 177–201.

[8] Masopust, T.: *Formal Models: Regulation and Reduction*, Ph.D. Thesis, Faculty of Information Technology, Brno, 2007.

[9] Meduna, A.: Six-Nonterminal Multi-Sequential Grammars Characterize the Family of Recursively Enumerable Languages, *International Journal of Computer Mathematics*, **65**, 1997, 179–189.

[10] Meduna, A.: Descriptional Complexity of Multi-Continuous Grammars, *Acta Cybernetica*, **13**, 1998, 375–384.

[11] Meduna, A.: Generative Power of Three-Nonterminal Scattered Context Grammars, *Theoretical Computer Science*, **246**, 2000, 279–284.

[12] Meduna, A.: Descriptional Complexity of Scattered Rewriting and Multirewriting: An Overview, *Journal of Automata, Languages and Combinatorics*, **7**, 2002, 571–577.

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

[14] Vaszil, G.: On the Descriptional Complexity of Some Rewriting Mechanisms Regulated by Context Conditions, *Theoretical Computer Science*, **330**, 2005, 361–373.