ELSEVIER

Available online at www.sciencedirect.com
SCIENCE DIRECT

Theoretical Computer Science 330 (2005) 361–373

Theoretical Computer Science

www.elsevier.com/locate/tcs

On the descriptional complexity of some rewriting mechanisms regulated by context conditions

György Vaszil

Computer and Automation Research Institute of the Hungarian Academy of Sciences, Kende utca 13-17, 1111 Budapest, Hungary

Received 24 November 2003; received in revised form 7 April 2004; accepted 4 June 2004

Abstract

We improve the upper bounds of certain descriptional complexity measures of two types of rewriting mechanisms regulated by context conditions. We prove that scattered context grammars having two context sensing productions and five nonterminals are sufficient to generate all recursively enumerable languages and we also show that the same power can be reached by simple semi-conditional grammars having 10 conditional productions with conditions of the length two or eight conditional productions with conditions of length three. The results are based on the common idea of using the so called Geffert normal forms for phrase structure grammars.

Keywords: Regulated rewriting; Descriptional complexity; Simple semi-conditional grammars; Scattered context grammars

1. Introduction

The general idea behind regulated rewriting with context-free grammars can be formulated as follows: Given a context-free grammar, a control mechanism is added which restricts the application of the rules in such a way that some of the derivations possible in the usual context-free derivation process are eliminated. Thus, the generated set of words is a subset of the original context-free language generated by the grammar without the

E-mail address: vaszil@sztaki.hu (Gy. Vaszil).

0304-3975/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2004.06.032

control. Since these generated subsets can be noncontext-free languages, these mechanisms are more powerful than context-free grammars.

The need for such devices is raised by the study of phenomena occurring in several areas of mathematics, linguistics, or even developmental biology, which cannot be described by the capabilities of context-free languages. To study these areas, it is desirable to construct generative mechanisms which have as many context-free-like properties as possible, but are also able to describe the noncontext-free features of the specific languages in question. See [1] for a discussion on the necessity of noncontext-free constructions and also [2] for regulated rewriting in general.

A class of generative devices motivated by the ideas above use the presence/absence of certain symbols or substrings in their current sentential forms to have additional control over the application of their rewriting rules.

Well-known examples of this idea are scattered context grammars [8]. The productions of these grammars are ordered sequences of context-free rewriting rules which have to be applied in parallel on nonterminals appearing in the sentential form in the same order as the nonterminals on the left-hand sides of the rules appear in the production sequence. If a rule sequence contains more than one rule then the production is called context sensing. Scattered context grammars are known to characterize recursively enumerable languages, see [7,11,12,19].

A different way of using the sentential form to control the rule application is realized by conditional grammars, sometimes also called grammars with regular restriction, see [5,18]. The derivations in these mechanisms are controlled by regular languages associated to the context-free rules: The rules can only be applied to the sentential form if it belongs to the language associated to the given rule.

A weaker restriction is used in the generative mechanisms called semi-conditional grammars, see [9,17]. These are context-free grammars with a permitting and a forbidding condition associated to each production. The conditions are given in the form of two words; a production can only be used on a given sentential form if the permitting word is a subword of the sentential form and the forbidding word is not a subword of the sentential form. Note that semi-conditional grammars are special cases of conditional grammars, since the set of those sentential forms which satisfy the conditions associated to a rule is a regular set. Semi-conditional grammars are also known to generate the class of recursively enumerable languages.

Since it has been always important to describe formal languages as concisely and economically as possible, it is of interest to study these mechanisms from the point of view of descriptional complexity: the number of nonterminals, the number, and the complexity of the productions.

Scattered context grammars have been shown to generate all recursively enumerable languages with four nonterminals in [13], with three nonterminals in [14], and then with two context-sensing productions in [4]. In [3], a simultaneous reduction of these measures is attempted, it is shown that any recursively enumerable language can be generated by a scattered context grammar having seven nonterminals and five context sensing productions or six nonterminals and six context sensing productions.

In the case of semi-conditional grammars, the complexity of productions is measured by their degree, defined as the maximal length of the context conditions as a pair of integers,

(i, j), where i and j are the length of the longest permitting and the longest forbidding word, respectively. In [17], recursively enumerable languages were characterized by semi-conditional grammars of degree $(2, 1)$ or $(1, 2)$, while grammars of degree $(1, 1)$ without erasing productions were shown to generate only a subclass of context-sensitive languages. The investigation of the generative power of grammars having only permitting or only forbidding conditions were also started, the language classes generated by grammars of degree (1, 0) or (0, 1) without erasing productions were shown to be strictly included in the class of context-sensitive languages. $E0L$ and $ET0L$ systems with forbidding conditions only are also studied in [16], they are shown to generate any recursively enumerable language with forbidding context conditions of length two.

The study of semi-conditional grammars continued in [10] where simple semi-conditional grammars were introduced. We speak of a simple semi-conditional grammar if each rule has at most one nonempty condition, that is, no controlling context condition at all, or either a permitting, or a forbidding one. In [10], simple semi-conditional grammars of degree $(1, 2)$ and $(2, 1)$ were shown to be able to generate all recursively enumerable languages, but the number of rules necessary to obtain this power was unbounded. In [15], however, the authors prove that the class of recursively enumerable languages can be characterized by simple semi-conditional grammars of degree $(2, 1)$ with only 12 conditional productions.

In the following, we continue the study of this area. First, by combining the techniques used in [3,15], we improve the results of [3] by showing that any recursively enumerable language can be generated with scattered context grammars having two context sensing rules and five nonterminals. Then, by applying an improved version of the technique used in [15], we reduce further the number of conditional productions in simple semi-conditional grammars: We prove that ten of them are sufficient to generate all recursively enumerable languages with grammars of degree (2,1) or eight of them are sufficient to generate all recursively enumerable languages with grammars of degree (3,1).

The key idea of our proofs is the use of the normal form results for phrase structure grammars by Geffert from [6] which demonstrate that any recursively enumerable language can be generated with a finite but unbounded number of context-free rules plus a bounded and very small number of noncontext-free productions. Thus, it is natural to study the consequences of these normal forms from the point of view of the descriptional complexity of rewriting mechanisms which are regulated by the use of context conditions, such as scattered context or simple semi-conditional grammars.

2. Preliminaries and definitions

The reader is assumed to be familiar with the basic notions of formal language theory, for details refer to [18].

Let T^* denote the set of all words over a finite alphabet T. If the empty word denoted by ε is not included, then we use the notation T^+. The length of a word $w \in T^*$ is denoted by $|w|$, the number of occurrences of a symbol $x \in T$ in w is denoted by $|w|_x$, and the reverse of w is denoted by w^R. The set of subwords of a word $w \in T^+$, $\{y \in T^+ \mid w = xyz, x, z \in T^*\}$, is denoted by $sub(w)$.

A *scattered context grammar* is a construct $G = (N, T, P, S)$, where N is the nonterminal alphabet, T is the terminal alphabet, $S \in N$ is the start symbol, and P is a set of productions of the form $(X_1, \ldots, X_n) \to (\alpha_1, \ldots, \alpha_n)$, where $n \geqslant 1$, $X_i \in N$, $\alpha_i \in (N \cup T)^*$, $1 \leqslant i \leqslant n$. If $n \geqslant 2$, then the production is called *context sensing*.

We say that $x \in (N \cup T)^+$ directly derives $y \in (N \cup T)^*$ according to the rule $(X_1, \ldots, X_n) \to (\alpha_1, \ldots, \alpha_n) \in P$, denoted by $x \Rightarrow y$, if x can be written as $x = x_1 X_1 x_2 \ldots x_n X_n x_{n+1}$ with $x_i \in (N \cup T)^*$, $1 \leqslant i \leqslant n+1$, and $y = x_1 \alpha_1 x_2 \ldots x_n \alpha_n x_{n+1}$.

The language generated by a scattered context grammar G is the set $L(G) = \{w \in T^* \mid S \Rightarrow^* w\}$, where $\Rightarrow^*$ denotes the reflexive and transitive closure of $\Rightarrow$.

A *semi-conditional grammar* is a construct $G = (N, T, P, S)$ with nonterminal alphabet N, terminal alphabet T, a start symbol $S \in N$, and a set of productions of the form $(X \to \alpha, u, v)$ with $X \in N$, $\alpha \in (N \cup T)^*$, and $u, v \in (N \cup T)^+ \cup \{0\}$, where $0 \notin (N \cup T)$ is a special symbol. If $u \neq 0$ or $v \neq 0$, then the production is said to be *conditional*.

A semi-conditional grammar G has *degree* (i, j) if for all productions $(X \to \alpha, u, v)$, $u \neq 0$ implies $|u| \leqslant i$ and $v \neq 0$ implies $|v| \leqslant j$.

We say that $x \in (N \cup T)^+$ directly derives $y \in (N \cup T)^*$ according to the rule $(X \to \alpha, u, v) \in P$, denoted by $x \Rightarrow y$, if $x = x_1 X x_2$, $y = x_1 \alpha x_2$ for some $x_1, x_2 \in (N \cup T)^*$ and furthermore $u \neq 0$ implies that $u \in sub(x)$ and $v \neq 0$ implies that $v \notin sub(x)$.

If we denote the reflexive and transitive closure of $\Rightarrow$ by $\Rightarrow^*$, then the language generated by a semi-conditional grammar G is $L(G) = \{w \in T^* | S \Rightarrow^* w\}$.

We speak of a *simple semi-conditional grammar* if each production has at most one nonempty condition, that is, if $G = (N, T, P, S)$ is a simple semi-conditional grammar, then $(X \to \alpha, u, v) \in P$ implies $0 \in \{u, v\}$.

3. The descriptional complexity of scattered context grammars

Before proceeding to our argument, we need a normal form result from [6] stating that if $L \subseteq T^*$ is a recursively enumerable language, then L can be generated by a grammar

$$G = (N, T, P \cup \{AB \to \varepsilon,\ CD \to \varepsilon\}, S),$$

such that $N = \{S, S', A, B, C, D\}$ and P contains only context-free productions. Furthermore, the context-free rules of G are of the form

$$S \to zSx, \quad \text{where } z \in \{A, C\}^*, \quad x \in T,$$
$$S \to S',$$
$$S' \to uS'v, \quad \text{where } u \in \{A, C\}^*, \quad v \in \{B, D\}^*,$$
$$S' \to \varepsilon.$$

Considering the rules above, we can distinguish three phases in the generation of a terminal word $x_1 \ldots x_n, \quad x_i \in T, \quad 1 \leqslant i \leqslant n$.

(1) $S \Rightarrow^* z_n \ldots z_1 S x_1 \ldots x_n \Rightarrow z_n \ldots z_1 S' x_1 \ldots x_n$,
where $z_i \in \{A, C\}^*$, $1 \leqslant i \leqslant n$.

(2) $z_n \ldots z_1 S' x_1 \ldots x_n \Rightarrow^* z_n \ldots z_1 u_m \ldots u_1 S' v_1 \ldots v_m x_1 \ldots x_n \Rightarrow$
$z_n \ldots z_1 u_m \ldots u_1 v_1 \ldots v_m x_1 \ldots x_n$,

where $u_j \in \{A, C\}^*$, $v_j \in \{B, D\}^*$, $1 \leqslant j \leqslant m$. Now the terminal string $x_1 \dots x_n$ is generated by G if and only if using $AB \to \varepsilon$ and $CD \to \varepsilon$ the substring $z_n \dots z_1 u_m \dots u_1 v_1 \dots v_m$ can be erased.

(3) $z_n \dots z_1 u_m \dots u_1 v_1 \dots v_m x_1 \dots x_n \Rightarrow^* x_1 \dots x_n$.

This can be successful if and only if $z_n \dots z_1 u_m \dots u_1 = g(v_1 \dots v_m)^R$, where $g : \{B, D\} \to \{A, C\}$ is a morphism with $g(B) = A$ and $g(D) = C$.

We will refer to the three stages of a derivation of a grammar in the normal form above as the *first*, the *second*, and the *third phase*.

Now we state our first result concerning the descriptional complexity of scattered context grammars which considerably improves the bounds presented in [3]. We keep the number of context sensing rules at the currently known minimum of two and at the same time reduce the number of necessary nonterminals to five.

Theorem 1. *Every recursively enumerable language can be generated by a scattered context grammar with not more than two context sensing productions and five or less nonterminals.*

Proof. Let $L \subseteq T^*$ be a recursively enumerable language generated by the grammar

$$G = (N, T, P \cup \{AB \to \varepsilon, CD \to \varepsilon\}, S),$$

such that $N = \{S, S'A, B, C, D\}$ and P contains only context-free productions, as described above. Let us construct a scattered context grammar

$$G' = (N', T, P', S),$$

where $N' = \{S, S', 0, 1, \$\}$,

$$\begin{aligned} P' = \{&(S) \to (h(z)Sx) \mid S \to zSx \in P,\ z \in \{A, C\}^*,\ x \in T\} \\ &\cup\{(S) \to (S')\} \\ &\cup\{(S') \to (h(u)S'h(v)) \mid S' \to uS'v \in P,\ u \in \{A, C\}^*,\ v \in \{B, D\}^*\} \\ &\cup\{(S') \to (\$\$), (\$) \to (\varepsilon)\} \\ &\cup\{(0, \$, \$, 0) \to (\$, \varepsilon, \varepsilon, \$), (1, \$, \$, 1) \to (\$, \varepsilon, \varepsilon, \$)\}, \end{aligned}$$

and $h : \{A, B, C, D\}^* \to \{0, 1\}^*$ is a morphism defined as $h(A) = h(B) = 101$, $h(C) = h(D) = 1001$.

The idea behind this grammar is to generate encodings of the sentential forms produced by the first two phases of the derivations in G, strings of the form $zuvw$, $zu \in \{A, C\}^*$, $v \in \{B, D\}^*$, $w \in T^*$ such that the substring over $\{A, B, C, D\}$ is encoded using the nonterminals 0 and 1 to the string $h(zu)\$\$h(v)w$ and then to simulate the third phase of the derivation by erasing $h(zu)\$\$h(v)$ if and only if $h(zu)$ is equal to the reverse of $h(v)$, that is, if and only if $w \in L$.

By observing the productions it is clear that the first two phases of the derivation of G producing $zuvw$ (z, u, v, w as above) can be simulated by G' producing the encoded string $h(zu)\$\$h(v)w$ and if $h(zu)$ is equal to the reverse of $h(v)$, then $h(zu)\$\$h(v)$ can be erased using the context sensing rules $(0, \$, \$, 0) \to (\$, \varepsilon, \varepsilon, \$)$ and $(1, \$, \$, 1) \to (\$, \varepsilon, \varepsilon, \$)$

which also check the equality of $h(zu)$ and $h(v)^R$ and then the rule (\$) $\to$ (ε) to erase the markers.

Now we show that G' can only generate terminal strings which can also be generated by G. Derivations of G' start with

$$S \Rightarrow^* h(z)Sw \Rightarrow h(z)S'w \Rightarrow^* h(zu)S'h(v)w \Rightarrow h(zu)\$\$h(v)w,$$

where $zu \in \{A, C\}^*,\ v \in \{B, D\}^*,\ w \in T^*$.

Now the derivation can continue with the application of rules to delete \$ or one of the two context sensing erasing rules $(0, \$, \$, 0) \to (\$, \varepsilon, \varepsilon, \$)$, $(1, \$, \$, 1) \to (\$, \varepsilon, \varepsilon, \$)$. Note that if a \$ marker is deleted when there are other nonterminals present, then the derivation cannot be successfully finished. Note also, that if a nonterminal 0 or 1 is placed between the two \$ markers, then it cannot be eliminated, thus, the derivation cannot be successfully finished. This means that the context sensing erasing productions can only be applied to delete those two occurrences of 0s or 1s which appear directly before and after the \$\$ markers.

By the considerations above, $h(zu)\$\$h(v)$ can only be erased with the context sensing productions if $h(zu)$ is identical to the reverse of $h(v)$, thus, if and only if the string zuv can be erased with the rules $AB \to \varepsilon$ and $CD \to \varepsilon$ of G, that is, if and only if $w \in L$. □

4. The number of conditional productions of simple semi-conditional grammars

In [15] the authors show that an erasing rule of the form $XY \to \varepsilon$ (X and Y being two nonterminals) can be simulated by six conditional productions of a simple semi-conditional grammar, thus, to simulate a grammar in the normal form above, simple semi-conditional grammars with 12 conditional productions are sufficient. The proofs of our results follow a similar line of reasoning, but we use another normal form given also in [6]. As shown there, it is not difficult to see that if we encode the nonterminal A to XY, the nonterminal B to Z, the nonterminal C to X, and the nonterminal D to YZ, then one erasing rule of the form $XYZ \to \varepsilon$ is sufficient to have the same effect as $AB \to \varepsilon$ and $CD \to \varepsilon$ together.

Thus, all recursively enumerable languages $L \subseteq T^*$ can be generated by a grammar

$$G = (N, T, P \cup \{ABC \to \varepsilon\}, S),$$

such that $N = \{S, S', A, B, C\}$, and P contains only context-free productions of the form

$$S \to zSx, \quad \text{where } z \in \{A, B\}^*, \quad x \in T,$$
$$S \to S',$$
$$S' \to uS'v, \quad \text{where } u \in \{A, B\}^*, \quad v \in \{B, C\}^*,$$
$$S' \to \varepsilon.$$

Again, three phases can be distinguished in the generation of a terminal word $x_1 \ldots x_n,\ x_i \in T,\ 1 \leqslant i \leqslant n$.

(1) $S \Rightarrow^* z_n \ldots z_1 S x_1 \ldots x_n \Rightarrow z_n \ldots z_1 S' x_1 \ldots x_n$,
where $z_i \in \{A, B\}^*$, $1 \leqslant i \leqslant n$.

(2) $z_n \ldots z_1 S' x_1 \ldots x_n \Rightarrow^* z_n \ldots z_1 u_m \ldots u_1 S' v_1 \ldots v_m x_1 \ldots x_n \Rightarrow$
$z_n \ldots z_1 u_m \ldots u_1 v_1 \ldots v_m x_1 \ldots x_n$,

where $u_j \in \{A, B\}^*$, $v_j \in \{B, C\}^*$, $1 \leqslant j \leqslant m$, and the terminal string $x_1 \dots x_n$ is generated by G if and only if, using the erasing rule $ABC \rightarrow \varepsilon$, the substring $z_n \dots z_1 u_m \dots u_1 v_1 \dots v_m$ can be deleted.

(3) $z_n \dots z_1 u_m \dots u_1 v_1 \dots v_m x_1 \dots x_n \Rightarrow^* x_1 \dots x_n$.

We will refer to the three stages of a derivation of a grammar in the normal form above as the *first*, the *second*, and the *third phase*.

As it is proved in [16], all recursively enumerable languages can be generated by simple semi-conditional grammars of degree (2, 1) with 12 conditional productions. Now we prove that ten productions are sufficient to reach the same power.

Theorem 2. *Every recursively enumerable language can be generated by a simple semi-conditional grammar of degree* (2,1) *having ten or less conditional productions*.

Proof. Let $L \subseteq T^*$ be a recursively enumerable language generated by the grammar

$$G = (N, T, P \cup \{ABC \rightarrow \varepsilon\}, S)$$

as above.

Now we construct G', a simple semi-conditional grammar of degree $(2, 1)$ as follows: Let

$$G' = (N', T, P', S),$$

where $N' = \{S, S', A, A', B, B', B'', C, C', L, L', R\}$ and

$$\begin{aligned} P' = \{&(X \rightarrow \alpha, 0, 0) \mid X \rightarrow \alpha \in P\} \\ \cup\{&(A \rightarrow LA', 0, L), (B \rightarrow B', 0, B'), (C \rightarrow C'R, 0, R), \\ &(A' \rightarrow \varepsilon, A'B', 0), (C' \rightarrow \varepsilon, B'C', 0), (B' \rightarrow B'', LB', 0), \\ &(B'' \rightarrow \varepsilon, B''R, 0), (L \rightarrow L', LR, 0), (R \rightarrow \varepsilon, L'R, 0), \\ &(L' \rightarrow \varepsilon, 0, R)\}. \end{aligned}$$

By observing the productions of P', we can see that the terminal words generated by G can also be generated by the simple semi-conditional grammar G'. In the following, we show that G' cannot generate words that are not generated by G.

We will examine the possible derivations of G' starting with S and leading to a terminal word. The first two phases of a derivation by G can be reproduced using the nonconditional rules of P', the rules of the form $(X \rightarrow \alpha, 0, 0)$ where $X \rightarrow \alpha \in P$. Since the conditional rules do not involve the symbols S and S' neither on the left or right sides nor in the conditions, if we can apply conditional rules before S and S' both disappear then we can also apply them in the same way later. According to this observation, we can assume that the first application of a conditional rule happens when neither S nor S' is present in the sentential form, that is, when the generated word is of the form

$$zuvw, \quad \text{where } z, u \in \{A, B\}^*, v \in \{B, C\}^*, w \in T^*.$$

Now we show that the prefix zuv can be deleted by the conditional rules of G' if and only if it can be deleted by the rule $ABC \rightarrow \varepsilon$ of G.

By continuing the derivation, the application of some of the rules $(A \to LA', 0, L)$, $(B \to B', 0, B')$, or $(C \to C'R, 0, R)$ can follow (one rule at most once). If these rules do not produce any of the subwords $A'B'$ or $B'C'$, the derivation cannot continue, so if $x = zu$ and $y = v$, it is sufficient to check derivation paths starting from the strings
(1) $x_1LA'B'x_2yw$, where $x = x_1ABx_2$,
(2) $x_1LA'B'x_2y_1C'Ry_2w$, where $x = x_1ABx_2$, $y = y_1Cy_2$,
(3) $x_1LA'x_2y_1B'C'Ry_2w$, where $x = x_1Ax_2$, $y = y_1BCy_2$, or
(4) $xy_1B'C'Ry_2w$, where $y = y_1BCy_2$,
because A can only occur in x and C can only occur in y.

Now we show that if we continue the derivation, it either enters a blocking configuration or after deleting one occurrence of the substring ABC, we obtain a string which is either of one of the four types above or a terminal string.

Let us follow the derivations starting with each of these strings. We first assume that the substring $LA'B'C'R$ is not present in any of the above cases, that is, $x_2 \neq \varepsilon$ and/or $y_1 \neq \varepsilon$.
(1) If $x_2 \neq \varepsilon$, then from the first sentential form we obtain either
- $x_1'B'x_1''LB''x_2y'C'Ry''w$, where $x_1'Bx_1'' = x_1$, $y'Cy'' = y$,
- $x_1LB''\alpha'B'\alpha''C'Ry''w$, where $\alpha'B\alpha'' = x_2y'$, $\alpha'' \neq \varepsilon$, and $y'Cy'' = y$,
- $x_1LB''x_2y'B'Ry''w$, where $y'BCy'' = y$, or
- $x_1LB''x_2y'C'R\alpha'B'\alpha''w$, where $\alpha'B\alpha'' = y''$, $y'Cy'' = y$.

(2) If $x_2y_1 \neq \varepsilon$, then from the second sentential form we obtain either
- $x_1'B'x_1''LB''x_2y_1C'Ry_2w$, where $x_1'Bx_1'' = x_1$,
- $x_1LB''\alpha'B'\alpha''C'Ry_2w$, where $\alpha'B\alpha'' = x_2y_1$, $\alpha'' \neq \varepsilon$,
- $x_1LB''x_2y_1'B'Ry_2w$, where $y_1'B = y_1$, or
- $x_1LB''x_2y_1C'Ry_2'B'y_2''w$, where $y_2'By_2'' = y_2$.

(3) From the third sentential form we obtain
- $x_1LA'x_2y_1B'Ry_2w$.

(4) If $y_1 \neq \varepsilon$, then from the fourth sentential form we obtain
- $x_1LA'x_2y_1B'Ry_2w$, where $x_1Ax_2 = x$.

The derivation cannot continue from any of these sentential forms, thus, we need to have a string of the following form:

$$zuvw = xyw = x_1LA'B'C'Ry_2w,$$

where $x = zu \in \{A, B\}^*$, $y = v \in \{B, C\}^*$, and moreover, $x_1AB = x$ and $Cy_2 = y$ or $x_1A = x$ and $BCy_2 = y$. In two derivation steps we might obtain the following two strings:

$$x_1LA'B'C'Ry_2w \Rightarrow^2 x_1LB''C'Ry_2w, \quad \text{or } x_1LA'B'C'Ry_2w \Rightarrow^2 x_1LB'Ry_2w.$$

The derivation from the first string cannot be continued, so let us consider the second possibility and follow each derivation path starting with this string. First the rule $(B' \to B'', LB', 0)$ must be used producing $x_1LB''Ry_2w$. Now observe that independently of the substring $LB''R$. There is the possibility of rewriting one B to B' in x_1 or in y_2, so let us denote by $\bar{x}_1$ and $\bar{y}_2$ the strings with $g(\bar{x}_1\bar{y}_2) = x_1y_2$ and $|\bar{x}_1\bar{y}_2|_{B'} \leqslant 1$, where $g(B') = B$

and $g(X) = X$ for all $X \in N' \cup T$, $X \neq B$. Then we might obtain: $\bar{x}_1 L B'' R \bar{y}_2 w \Rightarrow \bar{x}_1 L R \bar{y}_2 w \Rightarrow \bar{x}_1 L' R \bar{y}_2 w \Rightarrow$
(1) $\bar{x}_1' L A' \bar{x}_1'' L' R \bar{y}_2 w$, where $\bar{x}_1' A \bar{x}_1'' = \bar{x}_1$ or
(2) $\bar{x}_1 L' \bar{y}_2 w$.
Note that these cases do not distinguish between sentential forms with different $\bar{x}_1$ and $\bar{y}_2$ as long as $g(\bar{x}_1 \bar{y}_2) = x_1 y_2$.
Consider the derivations starting from (1) by distinguishing two cases. If $\bar{x}_1'' \neq B' \bar{x}_1'''$, then we might obtain
(1) $\bar{x}_1' L A' \bar{x}_1'' L' R \bar{y}_2 w \Rightarrow \bar{x}_1' L A' \bar{x}_1'' L' \bar{y}_2 w \Rightarrow$
- $\bar{x}_1' L A' \bar{x}_1'' \bar{y}_2 w$,
- $\bar{x}_1' L A' \bar{x}_1'' L' \bar{y}_2' C' R \bar{y}_2'' w \Rightarrow \bar{x}_1' L A' \bar{x}_1'' L' \bar{y}_2''' B' R \bar{y}_2'' w$, where $\bar{y}_2' = B' \bar{y}_2'''$.

In the first case, we obtain a string, where the ABC substring is successfully erased and the result is either a string of one of the four forms at the beginning of the proof or the derivation will not be able to continue. Just as in the case of the second string, in which the derivation cannot continue.
If $\bar{x}_1'' = B' \bar{x}_1'''$, then we might obtain in a few derivation steps
(1) $\bar{x}_1' L A' B' \bar{x}_1''' L' R \bar{y}_2 w \Rightarrow \ldots \Rightarrow$
- $\bar{x}_1' L B'' \bar{x}_1''' L' \bar{y}_2' C' R \bar{y}_2'' w$,
- $\bar{x}_1' L B'' \bar{x}_1''' \bar{y}_2' C' R \bar{y}_2'' w$, or
- $\bar{x}_1' L B'' R \bar{y}_2'' w \Rightarrow \bar{x}_1' L R \bar{y}_2'' w$.

In the first two cases the derivation cannot be continued, in the third case another occurrence of the substring ABC was erased and we have a string of the same form as above.
If we consider the derivations starting from (2), we either have
(2) $\bar{x}_1 L' \bar{y}_2 w \Rightarrow \bar{x}_1 \bar{y}_2 w$
or we obtain a string of the form already considered under the previous case. The word $\bar{x}_1 \bar{y}_2 w$ is either terminal, or we can obtain from it a string of one of the four forms at the beginning of the proof, or the derivation is blocked.

We have seen that the derivations starting with the sentential form $zuvw$, as above, either enter a blocking configuration or exactly one occurrence of the substring ABC can be deleted by the rules of P'. If we note that P' contains ten conditional productions and that the degree of G' is (2, 1), then the proof is complete. □

Now we continue by studying one of the open problems from [15] where the authors propose to investigate simple semi-conditional grammars having a degree different from (2,1). In the next theorem we show that the number of conditional productions can be decreased further if we allow permitting conditions of length three, that is, grammars of degree (3,1).

Theorem 3. *Every recursively enumerable language can be generated by a simple semi-conditional grammar of degree* (3,1) *having eight or less conditional productions*.

Proof. Let $L \subseteq T^*$ be a recursively enumerable language generated by the grammar

$$G = (N, T, P \cup \{ABC \to \varepsilon\}, S)$$

as above.

Now we construct G', a simple semi-conditional grammar of degree (3, 1) as follows: Let

$$G' = (N', T, P', S),$$

where $N' = \{S, S', A, A', A'', B, B', B'', C, C', C''\}$ and

$$\begin{aligned} P' = \{&(X \to \alpha, 0, 0) \mid X \to \alpha \in P\} \\ \cup\{&(X \to X', 0, X') \mid X \in \{A, B, C\}\} \\ \cup\{&(C' \to C'', A'B'C', 0), (A' \to A'', A'B'C'', 0), (B' \to B'', A''B'C'', 0), \\ &(A'' \to \varepsilon, 0, C''), (C'' \to \varepsilon, 0, B'), (B'' \to \varepsilon, 0, 0)\}. \end{aligned}$$

The first two phases of generating a terminal word with the grammar G can be reproduced by G' using the rules of P', the rules of the form $(X \to \alpha, 0, 0)$, $X \to \alpha \in P$. The third phase, the application of the erasing production $ABC \to \varepsilon$, is simulated by the additional rules. By observing these additional rules, we can see that all words generated by G can also be generated by G'. In the following, we show that G' does not generate words that cannot be generated by G.

Let us follow the possible paths of derivation of G' generating a terminal word. The derivations start with S. While the sentential form contains S or S', it is of the form zSw or $zuS'vw$, $z, u, v \in \{A, B, C, A', B', C'\}^*$, $w \in T^*$, where if $g(X') = X$ for $X \in \{A, B, C\}$ and $g(X) = X$ for all other symbols of $N \cup T$, then $g(zSw)$ or $g(zuS'vw)$ are valid sentential forms of G. Furthermore, zu contains at most one occurrence of A', v contains at most one occurrence of C', and the whole sentential form, or to put it in another way, zuv contains at most one occurrence of B'. (To see this, note the forbidding conditions on the rules $(X \to X', 0, X')$, $X \in \{A, B, C\}$.) After the rule $S' \to \varepsilon$ is used, we get a sentential form $zuvw$ with z, u, v, and w as above and $g(zuvw)$ being a valid sentential form of G.

Now we show that zuv can be erased by G' if and only if $g(zuv)$ can be erased by G. We do this by showing that if we start from a sentential form $zuvw$ containing single occurrences of each primed symbol A', B', C', then in the next at most nine derivation steps, the derivation either enters a blocking configuration or the three primed symbols formed a substring $A'B'C'$ which is erased and nothing else is erased. (Thus, the conditional rules of P' really simulate the rule $ABC \to \varepsilon$ of P.)

If we start with a sentential form $zuvw$ containing single occurrences of each primed symbol, then to be able to continue the derivation these symbols must form a substring $A'B'C'$, so the sentential form must be of the form $z\bar{u}A'B'C'\bar{v}$, where either $u = \bar{u}A'B'$ and $v = C'\bar{v}$ or $u = \bar{u}A'$ and $v = B'C'\bar{v}$.

Until B' does not disappear (or equivalently, until B'' is not introduced), none of the erasing productions can be applied, so after the first use of the production $(B' \to B'', A''B'C'', 0)$ we have a sentential form of one of the following forms:

- $z\bar{u}A''B''C''\bar{v}w$,
- $zu_1A'u_2A''B''C''\bar{v}w$, where $u_1Au_2 = \bar{u}$,
- $z\bar{u}A''B''C''v_1C'v_2w$, where $v_1Cv_2 = \bar{v}$, or
- $zu_1A'u_2A''B''C''v_1C'v_2w$, where $u_1Au_2 = \bar{u}$, $v_1Cv_2 = \bar{v}$.

Now we denote by $xA''B''C''yw$ one of the sentential forms above and observe all possible derivations.

The first step can be taken in four different ways:
$xA''B''C''yw \Rightarrow$
(1) $x_1B'x_2A''B''C''yw \Rightarrow x_1B'x_2A''C''yw$, where $x_1Bx_2 = x$,
(2) $xA''B''C''y_1B'y_2w \Rightarrow xA''C''y_1B'y_2w$, where $y_1By_2 = y$,
(3) $xA''C''yw$, or
(4) $xA''B''yw$.

In cases (1) and (2), the derivation cannot continue because B' is present, so no erasing production can be applied and because it is impossible to have $A'B'C'$ or $A'B'C''$ as a substring. The derivation paths starting from (3) are as follows:

(3) $xA''C''yw \Rightarrow$
- (a) $x_1B'x_2A''C''yw$,
- (b) $xA''C''y_1B'y_2w$,
- (c) $xA''yw \Rightarrow$
  - (i) $xyw \Rightarrow$
    - (A) $x_1B'x_2yw$,
    - (B) $xy_1B'y_2w$,
  - (ii) $x_1B'x_2A''yw \Rightarrow x_1B'x_2yw$,
  - (iii) $xA''y_1B'y_2w \Rightarrow xy_1B'y_2w$,

where $x_1Bx_2 = x$ and $y_1By_2 = y$. In cases (3)(a) and (3)(b), the derivation cannot be continued, in the sentential forms of cases (3)(c)(i)(A), (3)(c)(i)(B), (3)(c)(ii), and (3)(c)(iii), the substring $A''B''C''$ is removed and they contain at most one occurrence of A', B', and C'.

Let us now consider the derivation paths starting from (4).

(4) $xA''B''yw \Rightarrow$
- (a) $xA''yw \Rightarrow$
  - (i) $xyw \Rightarrow$
    - (A) $x_1B'x_2yw$,
    - (B) $xy_1B'y_2w$,
  - (ii) $x_1B'x_2A''yw \Rightarrow x_1B'x_2yw$,
  - (iii) $xA''y_1B'y_2w \Rightarrow xy_1B'y_2w$,
- (b) $xB''yw \Rightarrow$
  - (i) $xyw \Rightarrow$
    - (A) $x_1B'x_2yw$,
    - (B) $xy_1B'y_2w$,
  - (ii) $x_1B'x_2B''yw \Rightarrow x_1B'x_2yw$,
  - (iii) $xB''y_1B'y_2w \Rightarrow xy_1B'y_2w$,
- (c) $x_1B'x_2A''B''yw \Rightarrow$
  - (i) $x_1B'x_2B''yw \Rightarrow x_1B'x_2yw$,
  - (ii) $x_1B'x_2A''yw \Rightarrow x_1B'x_2yw$,
- (d) $xA''B''y_1B'y_2w \Rightarrow$
  - (i) $xB''y_1B'y_2w \Rightarrow xy_1B'y_2w$,
  - (ii) $xA''y_1B'y_2w \Rightarrow xy_1B'y_2w$,

where $x_1Bx_2 = x$ and $y_1By_2 = y$. The substring $A''B''C''$ is erased from all of the strings produced along these paths. These strings contain at most one occurrence of the symbols A', B', and C'.

To summarize the considerations above, we can say that until the disappearance of all double primed symbols, A'', B'', and C'', only the erasing rules and the rule $(B \rightarrow B', 0, B')$ can be applied. We can see that the derivation either enters a blocking configuration or the substring $A'B'C'$ and only this substring, is completely erased, while the resulting sentential form again contains at most one occurrence of each primed symbol.

This means that the additional conditional productions and the production $(B'' \rightarrow \varepsilon, 0, 0)$ of P' correctly simulate the application of the erasing rule $ABC \rightarrow \varepsilon$. If we note that P' contains eight conditional productions and that the degree of G' is $(3, 1)$, then the proof is complete. □

5. Conclusion and open problems

In the present paper, we have studied descriptional complexity measures of scattered context grammars and simple semi-conditional grammars.

In Theorem 1, we have improved the result of [3] by showing that scattered context grammars with two context sensing productions and five nonterminals generate all recursively enumerable languages. This result is especially interesting since reducing the nonterminal complexity usually increases the necessary amount of "context sensitivity", but two as the number of context sensing productions is the best bound known so far, even conjectured to be optimal in [4].

Concerning the number of nonterminals, further reduction might be possible since the best bound known is three [12] (with an arbitrary number of context sensing productions).

In Theorem 2, we have improved the result of [15] by showing that simple semi-conditional grammars of degree $(2, 1)$ generate any recursively enumerable language with not more than ten conditional productions and in Theorem 3, we have continued the investigation of the area into one of the directions proposed in [15]. We have shown that allowing longer words as context conditions may help to reduce the number of conditional productions necessary to generate all recursively enumerable languages: we have proved that simple semi-conditional grammars of degree $(3, 1)$ generate any recursively enumerable language with not more than eight conditional productions.

We already know, see [17], that semi-conditional grammars (and thus, also simple semi-conditional grammars) of degree $(1, 1)$ without erasing rules generate only a subclass of context-sensitive languages, but we do not know what happens if we have simple semi-conditional grammars of degree $(1, 1)$ with erasing rules.

Furthermore, concerning the number of conditional productions of any degree, the optimality of our results is still to be investigated.

References

[1] J. Dassow, Gh. Păun, Regulated Rewriting in Formal Language Theory, Springer, Berlin, 1989.

[2] J. Dassow, Gh. Păun, A. Salomaa, Grammars with controlled derivations, in: G. Rozenberg, A. Salomaa (Eds.), Linear Modeling: Background and Application, Handbook of Formal Languages, vol. II, Springer, Berlin-Heidelberg, 1997, pp. 101–154.

[3] H. Fernau, A. Meduna, A simultaneous reduction of several measures of descriptional complexity in scattered context grammars, Inform. Process. Lett. 86 (2003) 235–240.

[4] H. Fernau, A. Meduna, On the degree of scattered context-sensitivity, Theoret. Comput. Sci. 290 (2003) 2121–2124.
[5] I. Frřs, Grammars with partial ordering of rules, Inform. and Control 12 (1968) 415–425.
[6] V. Geffert, Context-free-like forms for the phrase structure grammars, in: M.P. Chytil, L. Janiga, V. Koubek (Eds.), Proc. Mathematical Foundations of Computer Science 1988, 13th Symp. Carlsbad, Czechoslovakia, August 29–September 2, 1988. Lecture Notes in Computer Science, vol. 324, Springer, Berlin, 1988, pp. 309–317.
[7] J. Gonczarowski, M.K. Warmuth, Scattered and context-sensitive rewriting, Acta Inform. 20 (1989) 391–411.
[8] S.A. Greibach, J.E. Hopcroft, Scattered context grammars, J. Comput. Syst. Sci. 3 (1969) 232–247.
[9] J. Kelemen, Conditional grammars: Motivations, definitions, and some properties, in: Proc. Conf. Automata, Languages and Mathematical Sciences, Salgótarján, 1984, pp. 110–123.
[10] A. Meduna, A. Gopalaratnam, On semi-conditional grammars with productions having either forbidding or permitting conditions, Acta Cybernet. 11 (4) (1994) 307–323.
[11] A. Meduna, A trivial method of characterizing the family of recursively enumerable languages by scattered context grammars, Bull. EATCS 56 (1995) 104–106.
[12] A. Meduna, Syntactic complexity of scattered context grammars, Acta Inform. 32 (1995) 285–298.
[13] A. Meduna, Four nonterminal scattered context grammars characterize the family of recursively enumerable languages, Internat. J. Comput. Math. 63 (1997) 67–83.
[14] A. Meduna, Generative power of three-nonterminal scattered context grammars, Theoret. Comput. Sci. 246 (2000) 279–284.
[15] A. Meduna, M. Švec, Reduction of simple semi-conditional grammars with respect to the number of conditional productions, Acta Cybernet. 15 (2002) 353–360.
[16] A. Meduna, M. Švec, Forbidding $ET0L$ systems, Theoret. Comput. Sci. 306 (2003) 449–469.
[17] Gh. Păun, A variant of random context grammars: semi-conditional grammars, Theoret. Comput. Sci. 41 (1985) 1–17.
[18] A. Salomaa, Formal Languages, Academic Press, New York, 1973.
[19] V. Virkkunen, On scattered context grammars, Acta Univ. Oulu. Sci. Rerum Natur. Ser. A, Math. 6 (1973) 75–82.