数理解析研究所講究録
第 1964 巻 2015 年 78-87

Regulated Grammars and Automata: The Gist with a Case Study

Alexander Meduna

Brno University of Technology, Faculty of Information Technology,
IT4Innovations Centre of Excellence, Božetěchova 1/2, 612 66 Brno, Czech Republic
meduna@fit.vutbr.cz, http://www.fit.vutbr.cz/~meduna/

Abstract. This paper explains the gist underlying regulated grammars and automata as well as the main purpose of their investigation. This investigation is classified into four major topics—the study of their power, properties, reduction, and mutual convertibility. The paper illustrates this investigation by a case study in terms of one-sided random-context grammars. Most importantly, it points out that propagating versions of one-sided random-context grammars characterize the family of context-sensitive languages, and with erasing rules, they characterize the family of recursively enumerable languages; as a result, they are stronger than ordinary random context grammars. Open problem areas are formulated.

1 Introduction

Formal language theory represents a branch of mathematics that formalizes languages and devices that define them strictly rigorously. This theory defines languages as sets of sequences consisting of symbols. This definition encompasses almost all languages as they are commonly understood. Indeed, natural languages, such as English, are included in this definition. Of course, all artificial languages introduced by various scientific disciplines can be viewed as formal languages as well; perhaps most illustratively, every programming language represents a formal language in terms of this definition. Consequently, formal language theory is important to all the scientific areas that make use of these languages to a certain extent.

The strictly mathematical approach to languages necessitates introducing *formal language models* that define them, and formal language theory have introduced a great variety of them over its history. Most of them are based upon rules by which they repeatedly rewrite sequences of symbols, called strings. Despite their diversity, they can be classified into two basic categories—generative and recognition language models. Generative models, better known as *grammars*, define strings of their language so their rewriting process generates them from a special start symbol. On the other hand, recognition models, better known as *automata*, define strings of their language by rewriting process that starts from these strings and ends in a special set of strings, usually called *final configurations*.

Like any branch of mathematics, formal language theory has defined its language models generally. Unfortunately, from a practical viewpoint, this generality actually means that the models work in a completely non-deterministic way, and as such, they are hardly implementable and, therefore, applicable in practice. Being fully aware of this pragmatic difficulty, formal language theory have introduced fully deterministic versions of these models; sadly, their application-oriented perspectives are also doubtful. First and foremost, in an ever-changing environment in which real language processors work, it is utterly naive, if not absurd, that these deterministic versions might adequately reflect and simulate real language processors applied in such pragmatically oriented areas as various engineering techniques for language analysis. Second, in many case, this determinism decreases the power of their general counterparts—another highly undesirable feature of this strict determinism.

2 The Gist and Purpose of Regulated Grammars and Automata

Considering all the difficulties sketched above, formal language theory have introduced yet another version of language models, generally referred to as *regulated language models*, which formalize real language processors perhaps most adequately. In essence, these models are based upon their general versions extended by an additional mathematical mechanism that prescribes the use of rules during the generation of their languages. From a practical viewpoint, an important advantage of these models consists in controlling their language-defining process and, therefore, operating in a more deterministic way than general models, which perform their derivations in a quite unregulated way. Perhaps even more significantly, the regulated versions of language models are stronger than their unregulated versions. Considering these advantages, it comes as no surprise that formal language theory has paid an incredibly high attention to *regulated grammars and automata.*

Over the past quarter century, literally hundreds studies were written about regulated grammars, and their investigation represents an exciting trend within formal language theory. This investigation has introduced a number of new regulated grammatical concepts and achieved many remarkable results.

From a more practical viewpoint the developers of current and future language processing technologies need a systematized body of mathematically precise knowledge upon which they can rely and build up their methods and techniques.

3 Major Investigation Areas

There exist four crucially important topics concerning regulated grammars and automata—their power, properties, reduction, and convertibility.

As obvious, the *power* of the regulated language models under consideration represents perhaps the most important information about them. Indeed, we always want to know the family of languages that these models define.

A special attention is paid to algorithms that arrange regulated grammars and automata so they satisfy some prescribed *properties* while the generated languages remain unchanged because many language processors strictly require their satisfaction in practice. From a theoretical viewpoint, these properties frequently simplify proofs demonstrating results about these grammars and automata.

The *reduction* of regulated grammars and automata also represents an important investigation area of this book because their reduced versions define languages in a succinct and easy-to-follow way. As obvious, this reduction simplifies the development of language processing technologies, which then work economically and effectively.

Of course, the same languages can be defined by different language models. We obviously tend to define them by the most appropriate models under given circumstances. Therefore, whenever discussing different types of equally powerful language models, we also study their mutual *convertibility.* More specifically, given a language model of one type, we study how to convert it to a language model of another equally powerful type so both the original model and the model produced by this conversion define the same language.

4 A Case Study

The present section deals with regulated grammars referred to as *random context grammars* (see Section 1.1 in [2]). In essence, these grammars regulate the language generation process so they require the presence of some prescribed symbols and, simultaneously, the absence of some others in the rewritten sentential forms. More precisely, random context grammars are based upon context-free rules, each of which may be extended by finitely many permitting and forbidding nonterminal symbols. A rule like this can rewrite the current sentential form provided that all its permitting symbols occur in the sentential form while all its forbidding words do not.

This paper concentrates its attention to *one-sided random context grammars* (see [7]) as slightly modified versions of ordinary random context grammars. That is, while random context grammars verify the presence and absence of symbols in sentential forms in their entirety, one-sided random context grammars perform this verification only in their prefixes or suffixes. More precisely, in every one-sided random context grammar, the set of rules is divided into the set of *left random context rules* and the set of *right random context rules*. When applying a left random context rule, the grammar checks the existence and absence of its permitting and forbidding symbols, respectively, only in the prefix to the left of the rewritten nonterminal. Similarly, when applying a right random context rule, it checks the existence and absence of its permitting and forbidding symbols, respectively, only in the suffix to the right of the rewritten nonterminal. Otherwise, it works just like any random context grammar.

The present section demonstrates that propagating versions of one-sided random-context grammars characterize the family of context-sensitive languages, and with erasing rules, they characterize the family of recursively enumerable languages. Furthermore, the section discusses the generative power of special cases of one-sided random-context grammars. Specifically, it proves that *one-sided permitting grammars*, which have only permitting rules, are more powerful than context-free grammars; on the other hand, they are no more powerful than propagating scattered context grammars. *One-sided forbidding grammars*, which have only forbidding rules, are equivalent to selective substitution grammars (see [5, 14]). Finally, *left forbidding grammars*, which have only left forbidding rules, are only as powerful as context-free grammars.

Preliminaries

In this section, we assume that the reader is familiar with formal language theory (see [15]). For a set Q, $\mathrm{card}(Q)$ denotes the cardinality of Q, and 2^Q denotes the power set of Q. For an alphabet (finite nonempty set) V, V^* represents the free monoid generated by V under the operation of concatenation. The unit of V^* is denoted by ε. For $x \in V^*$, $|x|$ denotes the length of x and $\mathrm{alph}(x)$ denotes the set of symbols occurring in x.

A *random context grammar* (see Section 1.1 in [2]) is a quadruple, $G = (N, T, P, S)$, where N and T are two disjoint alphabets of *nonterminals* and *terminals*, respectively, $S \in N$ is the *start symbol*, and $P \subseteq N \times (N \cup T)^* \times 2^N \times 2^N$ is a finite relation, called the set of *rules*. Set $V = N \cup T$. Each rule $(A, x, U, W) \in P$ is written as $(A \to x,\ U,\ W)$ throughout this section. The *direct derivation relation* over V^*, symbolically denoted by $\Rightarrow_G$, is defined as follows: if $u, v \in V^*$, $(A \to x,\ U,\ W) \in P$, $U \subseteq \mathrm{alph}(uAv)$, and $W \cap \mathrm{alph}(uAv) = \emptyset$, then $uAv \Rightarrow_G uxv$. U is called the *permitting context* and W is called the *forbidding context*. Let $\Rightarrow_G^*$ denote the reflexive-transitive closure of $\Rightarrow_G$. The *language of* G is denoted by $L(G)$ and defined as $L(G) = \{w \in T^* \mid S \Rightarrow_G^* w\}$.

Let $G = (N, T, P, S)$ be a random context grammar. Rules of the form $(A \to \varepsilon,\ U,\ W)$ are called *erasing rules*. If $(A \to x,\ U,\ W) \in P$ implies that $|x| \geq 1$, then G is a *propagating random context grammar*. If $(A \to x,\ U,\ W) \in P$ implies that $W = \emptyset$, then G is a *permitting grammar*. If $(A \to x,\ U,\ W) \in P$ implies that $U = \emptyset$, then G is a *forbidding grammar*. By analogy with propagating random context grammars, we define a *propagating permitting grammar* and a *propagating forbidding grammar*, respectively.

Denotation of Language Families

Throughout the rest of this section, the language families under discussion are denoted in the following way. **RC**, **P**, and **F** denote the language families generated by random context grammars, permitting grammars, and forbidding grammars, respectively. The notation with the upper index $-\varepsilon$ stands for the corresponding propagating family. For example, $\mathbf{RC}^{-\varepsilon}$ denotes the family of languages generated by propagating random context grammars. **CF**, **CS**, and **RE** denote the families of context-free languages, context-sensitive languages, and recursively enumerable languages, respectively. $\mathbf{SC}^{-\varepsilon}$, **S**, and $\mathbf{S}^{-\varepsilon}$ denote the language families generated by propagating scattered context languages (see [4]), selective substitution grammars (see [5, 14]), and propagating selective substitution grammars—that is, selective substitution grammars without erasing rules, see [5, 14]—, respectively.

Definitions and Examples

Next, we formally define one-sided random context grammars and their variants. In addition, we illustrate them by examples.

Definition 1. *A* one-sided random context grammar *is a quintuple*

$$G = \big(N, T, P_L, P_R, S\big)$$

where N and T are two disjoint alphabets, $S \in N$, and

$$P_L, P_R \subseteq N \times \big(N \cup T\big)^* \times 2^N \times 2^N$$

are two finite relations. Set $V = N \cup T$. The components V, N, T, P_L, P_R, and S are called the total alphabet, *the alphabet of* nonterminals, *the alphabet of* terminals, *the set of* left random context rules, *the set of* right random context rules, *and the* start symbol, *respectively. Each $(A, x, U, W) \in P_L \cup P_R$ is written as*

$$(A \rightarrow x, U, W)$$

throughout this section. For $(A \rightarrow x,\ U,\ W) \in P_L$, U and W are called the left permitting context *and the* left forbidding context, *respectively. For $(A \rightarrow x,\ U,\ W) \in P_R$, U and W are called the* right permitting context *and the* right forbidding context, *respectively.*

When applying a left random context rule, the grammar checks the existence and absence of its permitting and forbidding symbols, respectively, only in the prefix to the left of the rewritten nonterminal in the current sentential form. Analogously, when applying a right random context rule, it checks the existence and absence of its permitting and forbidding symbols, respectively, only in the suffix to the right of the rewritten nonterminal. The following definition states this formally.

Definition 2. *Let $G = (N,\ T,\ P_L,\ P_R,\ S)$ be a one-sided random context grammar. The* direct derivation relation *over V^* is denoted by $\Rightarrow_G$ and defined as follows. Let $u, v \in V^*$ and $(A \rightarrow x,\ U,\ W) \in P_L \cup P_R$. Then,*

$$uAv \Rightarrow_G uxv$$

if and only if

$$(A \to x, U, W) \in P_L, U \subseteq \text{alph}(u), \textit{ and } W \cap \text{alph}(u) = \emptyset$$

or

$$(A \to x, U, W) \in P_R, U \subseteq \text{alph}(v), \textit{ and } W \cap \text{alph}(v) = \emptyset$$

Let $\Rightarrow_G^*$ *denote the reflexive-transitive closure of* $\Rightarrow_G$.

The language generated by a one-sided random context grammar is defined as usual—that is, it consists of strings over the terminal alphabet that can be generated from the start symbol.

Definition 3. *Let* $G = (N, T, P_L, P_R, S)$ *be a one-sided random context grammar. The* language *of* G *is denoted by* $L(G)$ *and defined as*

$$L(G) = \{w \in T^* \mid S \Rightarrow_G^* w\}$$

Next, we define several special variants of one-sided random context grammars.

Definition 4. *Let* $G = (N, T, P_L, P_R, S)$ *be a one-sided random context grammar. Rules of the form* $(A \to \varepsilon, U, W)$ *are called* erasing rules. *If* $(A \to x, U, W) \in P_L \cup P_R$ *implies that* $|x| \geq 1$, *then* G *is a* propagating one-sided random context grammar. *If* $(A \to x, U, W) \in P_L \cup P_R$ *implies that* $W = \emptyset$, *then* G *is a* one-sided permitting grammar. *If* $(A \to x, U, W) \in P_L \cup P_R$ *implies that* $U = \emptyset$, *then* G *is a* one-sided forbidding grammar. *By analogy with propagating one-sided random context grammars, we define a* propagating one-sided permitting grammar *and a* propagating one-sided forbidding grammar, *respectively.*

Definition 5. *Let* $G = (N, T, P_L, P_R, S)$ *be a one-sided random context grammar. If* $P_R = \emptyset$, *then* G *is a* left random context grammar. *By analogy with one-sided permitting and forbidding grammars, we define a* left permitting grammar *(see [1]) and a* left forbidding grammar *(see [3]). Their propagating versions are defined analogously as well.*

Next, we illustrate the above definitions by three examples.

Example 1. Consider the one-sided random context grammar

$$G = \big(\{S, A, B, \bar{A}, \bar{B}\}, \{a, b, c\}, P_L, P_R, S\big)$$

where P_L contains the following four rules

$$\begin{array}{ll} (S \to AB, \emptyset, \emptyset) & (\bar{B} \to B, \{A\}, \emptyset) \\ (B \to b\bar{B}c, \{\bar{A}\}, \emptyset) & (B \to \varepsilon, \emptyset, \{A, \bar{A}\}) \end{array}$$

and P_R contains the following three rules

$$\begin{array}{lll} (A \to a\bar{A}, \{B\}, \emptyset) & (\bar{A} \to A, \{\bar{B}\}, \emptyset) & (A \to \varepsilon, \{B\}, \emptyset) \end{array}$$

It is rather easy to see that every derivation that generates a nonempty string of $L(G)$ is of the form

$$\begin{aligned} S &\Rightarrow_G AB \\ &\Rightarrow_G a\bar{A}B \\ &\Rightarrow_G a\bar{A}b\bar{B}c \\ &\Rightarrow_G aAb\bar{B}c \\ &\Rightarrow_G aAbBc \\ &\Rightarrow_G^* a^nAb^nBc^n \\ &\Rightarrow_G a^nb^nBc^n \\ &\Rightarrow_G a^nb^nc^n \end{aligned}$$

where $n \geq 1$. The empty string is generated by

$$S \Rightarrow_G AB \Rightarrow_G B \Rightarrow_G \varepsilon$$

Based on the previous observations, we see that G generates the non-context-free language $\{a^nb^nc^n \mid n \geq 0\}$.

Example 2. Consider $K = \{a^nb^mc^m \mid 1 \leq m \leq n\}$. This non-context-free language is generated by the one-sided permitting grammar

$$G = \big(\{S, A, B, X, Y\}, \{a, b, c\}, P_L, \emptyset, S\big)$$

with P_L containing the following seven rules

$(S \to AX,\ \emptyset,\ \emptyset)$

$(A \to a,\ \emptyset,\ \emptyset)$
$(A \to aB,\ \emptyset,\ \emptyset)$
$(B \to A,\ \emptyset,\ \emptyset)$

$(X \to bc,\ \emptyset,\ \emptyset)$
$(X \to bYc,\ \{B\},\ \emptyset)$
$(Y \to X,\ \{A\},\ \emptyset)$

Notice that G is, in fact, a propagating left permitting grammar. Observe that $(X \to bYc,\ \{B\},\ \emptyset)$ is applicable if B, produced by $(A \to aB,\ \emptyset,\ \emptyset)$, occurs to the left of X in the current sentential form. Similarly, $(Y \to X,\ \{A\},\ \emptyset)$ is applicable if A, produced by $(B \to A,\ \emptyset,\ \emptyset)$, occurs to the left of Y in the current sentential form. Consequently, it is rather easy to see that every derivation that generates $w \in L(G)$ is of the form

$$\begin{aligned} S &\Rightarrow_G AX \\ &\Rightarrow_G^* a^uAX \\ &\Rightarrow_G a^{u+1}BX \\ &\Rightarrow_G a^{u+1}BbYc \\ &\Rightarrow_G a^{u+1}AbYc \\ &\Rightarrow_G^* a^{u+1+v}AbYc \\ &\Rightarrow_G a^{u+1+v}AbXc \\ &\ \ \vdots \\ &\Rightarrow_G^* a^{n-1}Ab^{m-1}Xc^{m-1} \\ &\Rightarrow_G^2 a^nb^mc^m = w \end{aligned}$$

where $u, v \geq 0$, $1 \leq m \leq n$. Hence, $L(G) = K$.

Example 3. Consider the one-sided forbidding grammar

$$G = \big(\{S, A, B, A', B', \bar{A}, \bar{B}\}, \{a, b, c\}, P_L, P_R, S\big)$$

where P_L contains the following five rules

$$(S \to AB, \emptyset, \emptyset) \qquad (B \to bB'c, \emptyset, \{A, \bar{A}\}) \qquad (B' \to B, \emptyset, \{A'\})$$
$$(B \to \bar{B}, \emptyset, \{A, A'\}) \qquad (\bar{B} \to \varepsilon, \emptyset, \{\bar{A}\})$$

and P_R contains the following four rules

$$(A \to aA', \emptyset, \{B'\}) \qquad (A' \to A, \emptyset, \{B\})$$
$$(A \to \bar{A}, \emptyset, \{B'\}) \qquad (\bar{A} \to \varepsilon, \emptyset, \{B\})$$

Notice that every derivation that generates a nonempty string of $L(G)$ is of the form

$$\begin{aligned} S &\Rightarrow_G AB \\ &\Rightarrow_G aA'B \\ &\Rightarrow_G aA'bB'c \\ &\Rightarrow_G aAbB'c \\ &\Rightarrow_G aAbBc \\ &\Rightarrow_G^* a^nAb^nBc^n \\ &\Rightarrow_G a^n\bar{A}b^nBc^n \\ &\Rightarrow_G a^n\bar{A}b^n\bar{B}c^n \\ &\Rightarrow_G a^nb^n\bar{B}c^n \\ &\Rightarrow_G a^nb^nc^n \end{aligned}$$

where $n \geq 1$. The empty string is generated by

$$S \Rightarrow_G AB \Rightarrow_G \bar{A}B \Rightarrow_G \bar{A}\bar{B} \Rightarrow_G \bar{B} \Rightarrow_G \varepsilon$$

Based on the previous observations, we see that G generates the non-context-free language $\{a^nb^nc^n \mid n \geq 0\}$.

Denotation of Language Families

Throughout the rest of this section, the language families under discussion are denoted in the following way. **ORC**, **OP**, and **OF** denote the language families generated by one-sided random context grammars, one-sided permitting grammars, and one-sided forbidding grammars, respectively. **LRC**, **LP**, and **LF** denote the language families generated by left random context grammars, left permitting grammars, and left forbidding grammars, respectively.

The notation with the upper index $-\varepsilon$ stands for the corresponding propagating family. For example, $\mathbf{ORC}^{-\varepsilon}$ denotes the family of languages generated by propagating one-sided random context grammars.

Results

In this section, we give an overview of the established results concerning one-sided random context grammars. More details can be found in the cited papers and in Chapter 6 of [13].

Generative Power

First, we investigate the generative power of one-sided random context grammars. In [7], it is proved that one-sided random context grammars characterize the family of recursively enumerable languages, and that their propagating versions characterize the family of context-sensitive languages.

Theorem 1 (see Theorem 2 in [7]). $\mathbf{ORC} = \mathbf{RE}$ □

Theorem 2 (see Theorem 1 in [7]). $\mathbf{ORC}^{-\varepsilon} = \mathbf{CS}$ □

Since $\mathbf{RC}^{-\varepsilon} \subset \mathbf{CS}$ and $\mathbf{RC} = \mathbf{RE}$ (see [2]), we have that one-sided random context grammars are equally powerful as random context grammars, while propagating one-sided random context grammars are more powerful than propagating random context grammars.

Theorem 3 (see Corollary 3 in [7]). $\mathbf{RC}^{-\varepsilon} \subset \mathbf{ORC}^{-\varepsilon} \subset \mathbf{RC} = \mathbf{ORC}$ □

The power of one-sided forbidding grammars is investigated in [9]. In there, it is proved that they have the same power as selective substitution grammars (see [5, 14]).

Theorem 4 (see Theorem 3.7 in [9]). $\mathbf{OF} = \mathbf{S}$ □

Theorem 5 (see Theorem 3.8 in [9]). $\mathbf{OF}^{-\varepsilon} = \mathbf{S}^{-\varepsilon}$ □

It is not known whether one-sided forbidding grammars or selective substitution grammars characterize the family of recursively enumerable languages. Also, it is not known whether these grammars without erasing rules characterize the family of context-sensitive languages.

Moreover, [9] proves the following two results concerning the generative power of one-sided forbidding grammar where the set of left random context rules coincide with the set of right random context rules.

Theorem 6 (see Theorem 3.11 in [9]). *A language K is context-free if and only if there is a one-sided forbidding grammar, $G = (N, T, P_L, P_R, S)$, satisfying $K = L(G)$ and $P_L = P_R$.* □

Theorem 7 (see Corollary 3.12 in [9]). *Let $G = (N, T, P_L, P_R, S)$ be a one-sided forbidding grammar satisfying $P_L = P_R$. Then, there is a propagating one-sided forbidding grammar H such that $L(H) = L(G) - \{\varepsilon\}$.* □

One-sided forbidding grammars are at least as powerful as forbidding grammars. This is stated in the next two theorems.

Theorem 8 (see Theorem 5 in [7]). $\mathbf{F} \subseteq \mathbf{OF}$ □

Theorem 9 (see Corollary 1 in [7]). $\mathbf{F}^{-\varepsilon} \subseteq \mathbf{OF}^{-\varepsilon}$ □

In terms of left forbidding grammars and their power, [3] proves that they are no more powerful than context-free grammars.

Theorem 10 (see Theorem 1 in [3]). $\mathbf{LF}^{-\varepsilon} = \mathbf{LF} = \mathbf{CF}$ □

From Theorems 9 and 10 above and from the fact that $\mathbf{CF} \subset \mathbf{F}^{-\varepsilon}$ (see [2]), we obtain the following corollary, which relates the language families generated by left forbidding grammars, one-sided forbidding grammars, and forbidding grammars.

Corollary 1. $\mathbf{LF}^{-\varepsilon} = \mathbf{LF} \subset \mathbf{F}^{-\varepsilon} \subseteq \mathbf{OF}^{-\varepsilon} \subseteq \mathbf{OF}$ □

Finally, the following two theorems relate the language families generated propagating one-sided permitting grammars and propagating left permitting grammars to other families of languages.

Theorem 11 (see Theorem 7 in [7]). $\mathbf{CF} \subset \mathbf{OP}^{-\varepsilon} \subseteq \mathbf{SC}^{-\varepsilon} \subseteq \mathbf{CS} = \mathbf{ORC}^{-\varepsilon}$ □

Theorem 12 (see Corollary 2 in [7]). $\mathbf{CF} \subset \mathbf{LP}^{-\varepsilon} \subseteq \mathbf{SC}^{-\varepsilon} \subseteq \mathbf{CS} = \mathbf{ORC}^{-\varepsilon}$ □

Recall that it is not known whether propagating scattered context grammars characterize the family of context-sensitive languages—that is, whether the inclusion $\mathbf{SC}^{-\varepsilon} \subseteq \mathbf{CS}$ above is, in fact, an identity (see [6]).

Reduction

Recall that one-sided random context grammars characterize the family of recursively enumerable languages (see Theorem 1). Of course, it is more than natural to ask whether the family of recursively enumerable languages is characterized by one-sided random context grammars with a limited number of nonterminals or rules. The present section gives an affirmative answer to this question.

The next theorem states that ten nonterminals suffice to generate any recursively enumerable language by a one-sided random context grammar.

Theorem 13 (see Theorem 1 in [8]). *For every recursively enumerable language K, there exists a one-sided random context grammar, $H = (N, T, P_L, P_R, S)$, such that $L(H) = K$ and* $\mathrm{card}(N) = 10$. □

Other Topics of Investigation

We conclude this section by briefly mentioning other topics related to one-sided random context grammars that have been investigated.

Leftmost Derivations By analogy with the three well-known types of leftmost derivations in regulated grammars (see [2]), three types of leftmost derivation restrictions placed upon one-sided random context grammars have been defined and studied in [10]. In the *type-*1 *derivation restriction*, during every derivation step, the leftmost occurrence of a nonterminal has to be rewritten. In the *type-*2 *derivation restriction*, during every derivation step, the leftmost occurrence of a nonterminal which can be rewritten has to be rewritten. In the *type-*3 *derivation restriction*, during every derivation step, a rule is chosen, and the leftmost occurrence of its left-hand side is rewritten. In [10], the following three results are demonstrated.

(I) One-sided random context grammars with type-1 leftmost derivations characterize the family of context-free languages.
(II) One-sided random context grammars with type-2 and type-3 leftmost derivations characterize the family of recursively enumerable languages.
(III) Propagating one-sided random context grammars with type-2 and type-3 leftmost derivations characterize the family of context-sensitive languages.

Generalized One-Sided Random Context Grammars We may generalize the concept of one-sided context from symbols to strings. Obviously, as one-sided random context grammars already characterize the family of recursively languages, such a generalization cannot increase their strength. However, a generalization likes this makes sense in terms of variants of one-sided random context grammars. In [11], one-sided forbidding grammars that can forbid strings instead of single symbols are studied, and it has been proved that they are computationally complete, even if all strings are formed by at most two symbols.

One-Sided Versions of Other Formal Models One-sided random context grammars are based upon context-free grammars. It is only natural to consider other types of grammars and equip them with one-sided random context. Some preliminary results in this direction have been achieved in [12], where ET0L grammars (see [16]) and their variants enhanced with left random context are studied.

References

[1] Csuhaj-Varju, T. E., Masopust, and G. Vaszil. Cooperating distributed grammar systems with permitting grammars as components. *Romanian Journal of Information Science and Technology*, 12(2):175–189, 2009.

[2] J. Dassow and G. Paun. *Regulated Rewriting in Formal Language Theory.* Springer, 1989.

[3] F. Goldefus, T. Masopust, and A. Meduna. Left-forbidding cooperating distributed grammar systems. *Theoretical Computer Science*, 20(3):1–11, 2010.

[4] S.A. Greibach and J.E. Hopcroft. Scattered context grammars. *Journal of Computer and System Sciences*, 3(3):233–247, 1969.

[5] H.C.M. Kleijn. Basic ideas of selective substitution grammars. In *Kelemenova, A., Kelemen, J. (eds.) Trends, Techniques, and Problems in Theoretical Computer Science, Lecture Notes in Computer Science*, volume 281, pages 75–95. Springer, 1987.

[6] A. Meduna and J. Techet. *Scattered Context Grammars and their Applications.* WIT Press, Southampton, 2010.

[7] A. Meduna and P. Zemek. One-sided random context grammars. *Acta Informatica*, 48(3):149–163, 2011.

[8] A. Meduna and P. Zemek. Nonterminal complexity of one-sided random context grammars. *Acta Informatica*, 49(2):55–68, 2012.

[9] A. Meduna and P. Zemek. One-sided forbidding grammars and selective substitution grammars. *International Journal of Computer Mathematics*, 89(5):586–596, 2012.

[10] A. Meduna and P. Zemek. One-sided random context grammars with leftmost derivations. In *LNCS Festschrift Series: Languages Alive*, volume 7300, pages 160–173. Springer Verlag, 2012.

[11] A. Meduna and P. Zemek. Generalized one-sided forbidding grammars. *International Journal of Computer Mathematics*, 90(2):127–182, 2013.

[12] A. Meduna and P. Zemek. Left random context et0l grammars. *Fundamenta Informaticae*, 123(3):289–304, 2013.

[13] A. Meduna and P. Zemek. *Regulated Grammars and Automata.* Springer, New York, 2014.

[14] G. Rozenberg. Selective substitution grammars (towards a framework for rewriting systems). part 1: Definitions and examples. *Elektronische Informationsverarbeitung und Kybernetik*, 13(9):455–463, 1977.

[15] G. Rozenberg and A. Salomaa. *Handbook of Formal Languages.* Springer, 1977.

[16] G. Rozenberg and A. Salomaa. *Mathematical Theory of L Systems.* Academic Press, 1980.