VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ONE-SIDED RANDOM CONTEXT GRAMMARS

JEDNOSTRANNÉ GRAMATIKY S NAHODILÝM KONTEXTEM

DISERTAČNÍ PRÁCE
PHD THESIS

AUTOR PRÁCE — Ing. PETR ZEMEK
AUTHOR

VEDOUCÍ PRÁCE — prof. RNDr. ALEXANDER MEDUNA, CSc.
SUPERVISOR

BRNO 2014

Abstract

This thesis introduces the notion of a *one-sided random context grammar* as a context-free-based regulated grammar, in which a set of *permitting symbols* and a set of *forbidding symbols* are attached to every rule, and its set of rules is divided into the set of *left random context rules* and the set of *right random context rules*. A left random context rule can rewrite a nonterminal if each of its permitting symbols occurs to the left of the rewritten symbol in the current sentential form while each of its forbidding symbols does not occur there. A right random context rule is applied analogically except that the symbols are examined to the right of the rewritten symbol.

The thesis is divided into three parts. The first part gives a motivation behind introducing one-sided random context grammars and places all the covered material into the scientific context. Then, it gives an overview of formal language theory and some of its lesser-known areas that are needed to fully grasp some of the upcoming topics.

The second part forms the heart of the thesis. It formally defines one-sided random context grammars and studies them from many points of view. Generative power, relations to other types of grammars, reduction, normal forms, leftmost derivations, generalized and parsing-related versions all belong between the studied topics.

The final part of this thesis closes its discussion by adding remarks regarding its coverage. More specifically, these remarks concern application perspectives, bibliography, and open problem areas.

Keywords

formal language theory, regulated grammars, random context grammars, one-sided random context grammars, permitting grammars, forbidding grammars, generative power, reduction, normal forms, leftmost derivations, generalized versions, LL versions

Bibliographic Citation

Zemek, P.: One-sided random context grammars, Ph.D. thesis, Faculty of Information Technology, Brno University of Technology, Brno, CZ (2014)

Abstrakt

Tato disertační práce zavádí *jednostranné gramatiky s nahodilým kontextem* jako řízené gramatiky založené na bezkontextových gramatikách. V těchto gramatikách je ke každému pravidlu přiřazena množina *povolujících symbolů* a množina *zakazujících symbolů* a množina pravidel je rozdělena na množinu *levých pravidel s nahodilým kontextem* a množinu *pravých pravidel s nahodilým kontextem*. Levým pravidlem s nahodilým kontextem lze přepsat neterminál pokud se všechny povolující symboly vyskytují vlevo od přepisovaného neterminálu a žádný zakazující symbol tam přítomen není. Pravé pravidlo s nahodilým kontextem lze aplikovat analogicky, ale ona kontrola na přítomnost a nepřítomnost symbolů je provedena doprava od přepisovaného neterminálu.

Práce je rozdělena na tři části. První část uvádí motivaci za zavedením jednostranných gramatik s nahodilým kontextem a umisťuje materiál pokrytý v této práci do vědeckého kontextu. Poté dává přehled základů teorie formálních jazyků a některých méně známých oblastí, jejichž znalost je nutná pro pochopení studovaného tématu.

Druhá část tvoří jádro práce. Formálně definuje jednostranné gramatiky s nahodilým kontextem a studuje je z mnoha pohledů. Mezi studovaná témata patří generativní síla, vztah k jiným typům gramatik, redukce, normální formy, nejlevější derivace, zobecněné a LL verze těchto gramatiky.

Třetí část této práce zakončuje diskusi několika poznámkami. Mezi ně patří poznámky týkající se aplikovatelnosti zavedených gramatik v praxi, bibliografie a otevřených problémů.

Klíčová slova

teorie formálních jazyků, řízené gramatiky, gramatiky s nahodilým kontextem, jednostranné gramatiky s nahodilým kontextem, povolující gramatiky, zakazující gramatiky, generativní síla, redukce, normální formy, nejlevější derivace, zobecněné verze, LL verze

Bibliografická citace

ZEMEK, Petr. *One-sided random context grammars*. Brno, 2014. Disertační práce. Fakulta informačních technologií, Vysoké učení technické v Brně.

Declaration

I hereby declare that this thesis is my own work that has been created under the supervision of prof. RNDr. Alexander Meduna, CSc. It is based on the following two books, one book chapter, and nine papers that I have written jointly with my supervisor: [78–84, 86, 87, 109, 110]. Furthermore, Chapter 9 is based on [74], which is a paper written together with Lukáš Vrábel. Where other sources of information have been used, they have been duly acknowledged.

Petr Zemek
March 3, 2014

Acknowledgements

This thesis was supported by several grants—namely, BUT FIT grants FIT-S-11-2 and FIT-S-14-2299, research plan CEZ MŠMT MSM0021630528, European Regional Development Fund in the IT4Innovations Centre of Excellence (MŠMT CZ1.1.00/02.0070), and Visual Computing Competence Center (TE01010415).

I wish to thank prof. RNDr. Alexander Meduna, CSc. for his support during his supervision of this work, for valuable and inspiring consultations, and for his advice and recommendations from which I have benefited greatly. I would also like to thank many colleagues from the university for fruitful discussions about formal languages. My special thanks go to Lukáš Vrábel, Zbyněk Křivka, Jiří Koutný, Ondřej Soukup, and Martin Čermák. Finally, I thank my family for their enthusiastic encouragement; most importantly, I deeply appreciate the great patience and constant support of my girlfriend Daniela.

Contents

Part I
Introduction and Terminology

This part gives an introduction to the present thesis in order to express all its upcoming discussion clearly and precisely. First, it places all the material covered in the thesis into the scientific context. Then, it gives an overview of formal language theory in order to make the entire thesis completely self-contained. This part consists of two chapters.

Chapter 1 represents an introduction to this thesis. First, it argues that regulated rewriting with its regulated formal models forms a very important branch of formal language theory as a whole. Then, it provides a motivation behind introducing and studying one-sided random context grammars—the main topic of this thesis—as a new variant of regulated grammars. After that, it gives the organization of the present thesis.

Chapter 2 gives an overview of formal language theory. Apart from its classical rudiments, it covers several lesser-known areas of this theory, such as fundamentals concerning three types of regulated grammars, because these areas are also needed to fully grasp some upcoming topics included in this thesis.

Readers having solid background in the topics covered in Chapter 2 can only treat it as a reference for the terminology used throughout the rest of this thesis.

Chapter 1
Introduction

Formal Languages and Regulated Grammars

Formal languages, such as programming languages, are applied in a great number of scientific disciplines, ranging from biology through linguistics up to informatics (see [98–100]). As obvious, to use them properly, they have to be precisely specified in the first place. Most often, they are defined by mathematical models with finitely many rules by which they rewrite sequences of symbols, called strings.

Over its history, formal language theory has introduced a great variety of these language-defining models. 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 a rewriting process that starts from these strings and ends in a special set of strings, usually called final configurations.

Concerning grammars, the classical theory of formal languages has often classified all grammars into two fundamental categories—*context-free grammars* and *non-context-free grammars*. As their name suggests, context-free grammars are based upon context-free rules, by which these grammars rewrite symbols regardless of the context surrounding them. As opposed to them, non-context-free grammars rewrite symbols according to context-dependent rules, whose application usually depends on rather strict conditions placed upon the context surrounding the rewritten symbols, and this way of context-dependent rewriting often makes them clumsy and inapplicable in practice. From this point of view, we obviously always prefer using context-free grammars, but they have their drawbacks, too. Perhaps most importantly, context-free grammars are significantly less powerful than non-context-free grammars. Considering all these pros and cons, it comes as no surprise that modern formal language theory has intensively and systematically struggled to come with new types of grammars that are underlined by context-free rules, but which are more powerful than ordinary context-free grammars. Regulated versions of context-free grammars, briefly referred to as *regulated grammars* in this thesis, represent perhaps the most successful and significant achievement in this direction. They are based upon context-free grammars extended by additional regulating mechanisms by which they control the way the language generation is performed.

Over the last four decades, formal language theory has introduced an investigated many regulated grammars (see [16, 79, 87], Chapter 13 of [54], and Chapter 3 of [99] for an overview of the most important results). Arguably, one of the most studied type of regulated grammars are random context grammars, which are central to this thesis.

Random Context Grammars

In essence, *random context grammars* (see Section 1.1 in [16]) 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 symbols do not occur there.

Random context grammars are significantly stronger than ordinary context-free grammars. In fact, they characterize the family of recursively enumerable languages (see Theorem 1.2.5 in [16]), and this computational completeness obviously represents their indisputable advantage. Also, *propagating random context grammars*, which do not have any erasing rules—that is, rules with the empty string on their right-hand sides—are stronger than context-free grammars. However, they are strictly less powerful than context-sensitive grammars. Indeed, they generate a language family that is strictly included in the family of context sensitive languages (see Theorem 1.2.4 in [16]).

From a pragmatical standpoint, however, random context grammars have a drawback consisting in the necessity of scanning the current sentential form in its entirety during every single derivation step. From this viewpoint, it is highly desirable to modify these grammars so they scan only a part of the sentential form, yet they keep their computational completeness. *One-sided random context grammars*—the topic of the present thesis—represent a modification like this.

One-Sided Random Context Grammars

Specifically, 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 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. Otherwise, it works just like any ordinary random context grammar.

As the main result of this thesis, we demonstrate that propagating versions of one-sided random context grammars, which possess no erasing rules, characterize the family of context-sensitive languages, and with erasing rules, they characterize the family of recursively enumerable languages.

Furthermore, we discuss the generative power of several special cases of one-sided random context grammars. Specifically, we prove 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 so-called scattered context grammars (see [35, 73]). *One-sided forbidding grammars*, which have only forbidding rules, are equivalent to so-called selective substitution grammars (see [39, 40, 95]). Finally, *left forbidding grammars*, which have only left-sided forbidding rules, are only as powerful as context-free grammars.

Apart from the generative power of one-sided random context grammars and their special cases, we investigate the following aspects of these grammars. First, we establish four normal forms of one-sided random context grammars, in which all rules satisfy some prescribed properties or format. Then, we study a reduction of one-sided random context grammars with respect to the number of nonterminals and rules. After that, we place three leftmost derivation restrictions on one-sided random context grammars and investigate their generative power. We also study generalized versions of one-sided random context grammars, in which strings of symbols rather than single symbols can be required or forbidden. Finally, we study one-sided random context grammars from a more practical viewpoint by investigating their parsing-related variants.

To summarize, this thesis is primarily and principally meant as a theoretical treatment of one-sided random context grammars, which represent a modification of random context grammars. Apart from this theoretical treatment, however, we also cover some application perspectives to give the reader ideas about their applicability in practice.

Motivation

Taking into account the definition of one-sided random context grammars and all the results sketched above, we see that these grammars may fulfill an important role in the language theory and its applications for the following four reasons.

(I) From a practical viewpoint, one-sided random context grammars examine the existence of permitting symbols and the absence of forbidding symbols only within a portion of the current sentential form while ordinary random context grammars examine the entire current sentential form. As a result, the one-sided versions of these grammars work in a more economical and, therefore, efficient way than the ordinary versions. Moreover, one-sided random context grammars provide a finer control over the regulation process. Indeed, the designer of the grammar may select whether the presence or absence of symbols is examined to the left or to the right. In the case of ordinary random context grammars, this selection cannot be done since they scan the sentential forms in their entirety.

(II) The one-sided versions of propagating random context grammars are stronger than ordinary propagating random context grammars. Indeed, the language family defined by propagating random context grammars is properly included in the family of context-sensitive languages (see Theorem 1.2.4 in [16]). One-sided random context grammars are as powerful as ordinary random context grammars. These results come as a surprise because one-sided random context grammars examine only parts of sentential forms as pointed out in (I) above.

(III) Left forbidding grammars were introduced in [31], which also demonstrated that these grammars only define the family of context-free languages (see Theorem 1 in [31]). It is more than natural to generalize left forbidding grammars to one-sided forbidding grammars, which are stronger than left forbidding grammars (see Corollary 4.3.6 in this thesis). As a matter of fact, even *propagating left permitting grammars*, introduced in [10], are stronger than left forbidding grammars because they define a proper superfamily of the family of context-free languages (see Corollary 4.3.4 in this thesis). In the thesis, we also generalize left permitting grammars to one-sided permitting grammars and study their properties.

(IV) In the future, one might find results achieved in this thesis useful when attempting to solve some well-known open problems. Specifically, recall that every propagating scattered context grammar can be turned to an equivalent context-sensitive grammar (see Theorem 3.21 in [73]), but it is a longstanding open problem whether these two kinds of grammars are actually equivalent—the *PSC = CS problem* (see [73]). If in the future one proves that propagating one-sided permitting grammars and propagating one-sided random context grammars are equivalent, then so are propagating scattered context grammars and context-sensitive grammars (see Theorem 4.3.3 in this thesis), so the PSC = CS problem would be solved.

Organization

The text is divided into ten chapters. After this introductory Chapter 1, Chapter 2 briefly reviews formal language theory. It covers all the notions that are necessary to follow the rest of this thesis.

Chapters 3 through 9 represent the heart of this thesis. They introduce one-sided random context grammars and study them from many points of view. In a greater detail, Chapter 3 defines one-sided random context grammars and illustrates them by examples. Chapter 4 studies the generative power of these grammars. Chapter 5 establishes four normal forms of one-sided random context grammars. Chapter 6 investigates their descriptional complexity. Chapter 7 introduces three types of leftmost derivation restrictions placed upon one-sided random context grammars, and studies their effect to the generative power of these grammars. Chapter 8 introduces and investigates generalized versions of one-sided random context grammars. Chapter 9 introduces and investigates parsing-related variants of one-sided random context grammars, which may be applied in practice.

Chapter 10 closes the thesis by making several final remarks concerning the covered material with a special focus on its future developments. It concerns application perspectives of one-sided random context grammars, bibliographic comments and references, and open problem areas.

Chapter 2
Rudiments of Formal Language Theory

The present chapter briefly reviews formal language theory. It covers all the notions that are necessary to follow the rest of this thesis. Apart from well-known essentials of formal language theory, it includes lesser-known notions, such as a variety of regulated grammars. They are also needed to establish several upcoming results, so the reader should pay a special attention to them, too.

This chapter consists of three sections. Section 2.1 gives the used mathematical notation. Section 2.2 covers strings, languages, and operations over them. Section 2.3 concerns grammars and language families.

2.1 Mathematical Notation

For a set Q, $\mathrm{card}(Q)$ denotes the cardinality of Q, and 2^Q denotes the power set of Q. For two sets P and Q, $P \subseteq Q$ denotes that P is a subset of Q; $P \subset Q$ denotes that $A \subseteq B$ and $A \neq B$—that is, P is a proper subset of Q. Set intersection, union, and difference are denoted by $\cap$, $\cup$, and $-$, respectively. The empty set is denoted by $\emptyset$.

For a relation ρ, ρ^+ and ρ^* denote the transitive and transitive-reflexive closure of ρ, respectively.

2.2 Strings and Languages

An *alphabet* Σ is a finite, nonempty set of elements called *symbols*. A *string* over Σ is any finite sequence of symbols from Σ. We omit all separating commas in strings; that is, for a string $a_1, a_2, \ldots, a_n$, for some $n \geq 1$, we write $a_1 a_2 \cdots a_n$ instead. The *empty string*, denoted by ε, is the string that is formed by no symbols—that is, the empty sequence. By Σ^*, we denote the set of all strings over Σ (including ε). Set $\Sigma^+ = \Sigma^* - \{\varepsilon\}$.

Let x be a string over Σ—that is, $x \in \Sigma^*$—and express x as $= a_1a_2 \cdots a_n$, where $a_i \in \Sigma$, for all $i = 1 \ldots, n$, for some $n \geq 0$ (the case when $n = 0$ means that $x = \varepsilon$). The *length* of x, denoted by $|x|$, is defined as $|x| = n$. The *alphabet* of x, denoted by $\mathrm{alph}(x)$, is defined as $\mathrm{alph}(x) = \{a_1, a_2, \ldots, a_n\}$; informally, it is the set of symbols appearing in x. Notice that $|\varepsilon| = 0$ and $\mathrm{alph}(\varepsilon) = \emptyset$.

Let x and y be two strings over Σ. Then, xy is the *concatenation* of x and y. Note that $x\varepsilon = \varepsilon x = x$. If x can be written in the form $x = uv$, for some $u, v \in \Sigma^*$, then u is a *prefix* of x and v is a *suffix* of x. If $0 < |u| < |x|$, then u is a *proper prefix* of x; similarly, if $0 < |v| < |x|$, then v is a *proper suffix* of x. If $x = uvw$, for some $u, v, w \in \Sigma^*$, then v is a *substring* of x. The set of all substrings of x is denoted by $\mathrm{sub}(x)$.

Let n be a nonnegative integer. Then, the *nth power* of x, denoted by x^n, is a string over Σ recursively defined as

$$
\begin{array}{l}
(1)\ x^0 = \varepsilon \\
(2)\ x^n = xx^{n-1} \text{ for } n \geq 1
\end{array}
$$

A *language* L over Σ is any set of strings over Σ—that is, $L \subseteq \Sigma^*$. The set Σ^* is called the *universal language* because it consists of all strings over Σ. If L is a finite set, then it is a *finite language*; otherwise, it is an *infinite language*. The set of all finite languages over Σ is denoted by $\mathrm{fin}(\Sigma)$. For $L \in \mathrm{fin}(\Sigma)$, $\text{max-len}(L)$ denotes the length of the longest string in L. We set $\text{max-len}(\emptyset) = 0$. The *empty language* is denoted by $\emptyset$.

The *alphabet* of L, denoted by $\mathrm{alph}(L)$, is defined as

$$\mathrm{alph}(L) = \bigcup_{x \in L} \mathrm{alph}(x)$$

As all languages are sets, all common operations over sets can be applied to them. Specifically,

$$
\begin{aligned}
L_1 \cup L_2 &= \{x \mid x \in L_1 \text{ or } x \in L_2\} \\
L_1 \cap L_2 &= \{x \mid x \in L_1 \text{ and } x \in L_2\} \\
L_1 - L_2 &= \{x \mid x \in L_1 \text{ and } x \notin L_2\}
\end{aligned}
$$

There are also some special operations which apply only to languages. The *concatenation* of L_1 and L_2, denoted by L_1L_2, is the set

$$L_1L_2 = \big\{x_1x_2 \mid x_1 \in L_1 \text{ and } x_2 \in L_2\big\}$$

Note that $L\{\varepsilon\} = \{\varepsilon\}L = L$. For $n \geq 0$, the *nth power* of L, denoted by L^n, is recursively defined as

$$(1)\ L^0 = \{\varepsilon\}$$
$$(2)\ L^n = L^{n-1}L$$

The *closure* (*Kleene star*) of a language L, denoted by L^*, is the set

$$L^* = \bigcup_{i \geq 0} L^i$$

The *positive closure* of a language L, denoted by L^+, is the set

$$L^+ = \bigcup_{i \geq 1} L^i$$

Let Σ and Γ be two alphabets. A total function σ from Σ^* to 2^{Γ^*} such that $\sigma(uv) = \sigma(u)\sigma(v)$, for every $u, v \in \Sigma^*$, is a *substitution*. By this definition, $\sigma(\varepsilon) = \{\varepsilon\}$ and $\sigma(a_1 a_2 \cdots a_n) = \sigma(a_1)\sigma(a_2) \cdots \sigma(a_n)$, where $n \geq 1$ and $a_i \in \Sigma$, for all $i = 1, 2, \ldots, n$, so σ is completely specified by defining $\sigma(a)$ for each $a \in \Sigma$. If $\sigma(a)$ is finite, for every $a \in \Sigma$, then σ is said to be *finite*. For $L \subseteq \Sigma^*$, we extend the definition of σ to

$$\sigma(L) = \bigcup_{w \in L} \sigma(w)$$

A total function φ from Σ^* to Γ^* such that $\varphi(uv) = \varphi(u)\varphi(v)$, for every $u, v \in \Sigma^*$, is a *homomorphism*. As any homomorphism is a special case of finite substitution, we specify φ by analogy with the specification of σ. For $L \subseteq \Sigma^*$, we extend the definition of φ to

$$\varphi(L) = \{\varphi(w) \mid w \in L\}$$

By analogy with set theory, sets whose members are languages are called *families of languages*. If some language models define the same language family $\mathscr{L}$, we say that they are *equivalent* or, synonymously, *equally powerful*. Regarding $\mathscr{L}$, we say that these models *characterize* or *define* $\mathscr{L}$. For instance, in the next section, we review phrase-structure grammars that characterize the family of recursively enumerable languages.

2.3 Grammars and Language Families

In this section, we define devices that generate languages. These devices are called grammars, and they play a major role in formal language theory.

Definition 2.3.1. A *phrase-structure grammar* is a quadruple

$$G = (N, T, P, S)$$

where

- N is an alphabet of *nonterminals*;
- T is an alphabet of *terminals* such that $N \cap T = \emptyset$;
- P is a finite relation from $(N \cup T)^* N (N \cup T)^*$ to $(N \cup T)^*$;
- $S \in N$ is the *start symbol*.

Pairs $(u, v) \in P$ are called *rewriting rules* (abbreviated *rules*) and are written as $u \to v$. The set $V = N \cup T$ is the *total alphabet* of G. A rewriting rule $u \to v \in P$ satisfying $v = \varepsilon$ is called an *erasing rule*. If there is no such rule in P, then we say that G is *propagating*.

The G-based *direct derivation relation* over V^* is denoted by $\Rightarrow_G$ and defined as

$$x \Rightarrow_G y$$

if and only if $x = x_1 u x_2, y = y_1 v y_2$, and $u \to v \in P$, where $x_1, x_2, y_1, y_2 \in V^*$.

Since $\Rightarrow_G$ is a relation, $\Rightarrow_G^k$ is the kth power of $\Rightarrow_G$, for $k \geq 0$, $\Rightarrow_G^+$ is the transitive closure of $\Rightarrow_G$, and $\Rightarrow_G^*$ is the reflexive-transitive closure of $\Rightarrow_G$. Let $D\colon S \Rightarrow_G^* x$ be a derivation, for some $x \in V^*$. Then, x is a *sentential form*. If $x \in T^*$, then x is a *sentence*. If x is a sentence, then D is a *successful* (or *terminal*) *derivation*.

The *language* of G, denoted by $L(G)$, is the set of all sentences defined as

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

For brevity, we sometimes denote a rule $u \to v$ with a unique label r as $r\colon u \to v$, and instead of $u \to v \in P$, we simply write $r \in P$. When a derivation $u \Rightarrow_G v$ is made according to rule $r \in P$, we sometimes write $u \Rightarrow_G v \; [r]$ to point this out.

In the literature, a phrase-structure grammar is also often defined with rules of the form $x \to y$, where $x \in V^+$ and $y \in V^*$ (see, for instance, [107]). Both definitions are interchangeable in the sense that the grammars defined in these two ways generate the same family of languages—the family of recursively enumerable languages.

Definition 2.3.2. A *recursively enumerable language* is a language generated by a phrase-structure grammar. The family of recursively enumerable languages is denoted by **RE**. □

Definition 2.3.3. A *context-sensitive grammar* is a phrase-structure grammar

$$G = \left(N, T, P, S\right)$$

such that every $u \to v$ in P is of the form

$$u = x_1 A x_2, \; v = x_1 y x_2$$

where $x_1, x_2 \in V^*$, $A \in N$, and $y \in V^+$. A *context-sensitive language* is a language generated by a context-sensitive grammar. The family of context-sensitive languages is denoted by **CS**. □

Definition 2.3.4. A *context-free grammar* is a phrase-structure grammar

$$G = (N, T, P, S)$$

such that every rule in P is of the form

$$A \to x$$

where $A \in N$ and $x \in V^*$. A *context-free language* is a language generated by a context-free grammar. The family of context-free languages is denoted by **CF**. □

Concerning the families of context-free, context-sensitive, and recursively enumerable languages, the next important theorem holds true.

Theorem 2.3.5 (Chomsky Hierarchy, see [6, 7]). CF ⊂ CS ⊂ RE

Next, we recall leftmost derivations in context-free grammars.

Definition 2.3.6. Let $G = (N, T, P, S)$ be a context-free grammar. The relation of a *direct leftmost derivation*, denoted by ${}_{\mathrm{lm}}\Rightarrow_G$, is defined as follows: if $u \in T^*$, $v \in V^*$, and $r\colon A \to x \in P$, then

$$uAv \;{}_{\mathrm{lm}}\Rightarrow_G uxv\; [r]$$

Let ${}_{\mathrm{lm}}\Rightarrow_G^n$ and ${}_{\mathrm{lm}}\Rightarrow_G^*$ denote the nth power of ${}_{\mathrm{lm}}\Rightarrow_G$, for some $n \geq 0$, and the transitive closure of ${}_{\mathrm{lm}}\Rightarrow_G$, respectively. The *language that G generates by using leftmost derivations* is denoted by $L(G, {}_{\mathrm{lm}}\Rightarrow)$ and defined as

$$L(G, {}_{\mathrm{lm}}\Rightarrow) = \{w \in T^* \mid S \;{}_{\mathrm{lm}}\Rightarrow_G^* w\}$$ □

Without any loss of generality, in context-free grammars, we may consider only leftmost derivations, which is formally stated in the following theorem.

Theorem 2.3.7 (see [69]). *Let G be a context-free grammar. Then,*

$$L(G, {}_{\mathrm{lm}}\Rightarrow) = L(G)$$

Normal Forms

Next, we review several normal forms of context-sensitive and phrase-structure grammars, namely the Penttonen and Geffert normal forms.

Definition 2.3.8. Let $G = (N, T, P, S)$ be a phrase-structure grammar. G is in the *Penttonen normal form* (see [91]) if every rule in P is in one of the following four forms

(i) $AB \to AC$ (ii) $A \to BC$ (iii) $A \to a$ (iv) $A \to \varepsilon$

where $A, B, C \in N$, and $a \in T$. □

Theorem 2.3.9 (see [91]). *For every phrase-structure grammar G, there is a phrase-structure grammar G' in the Penttonen normal form such that $L(G') = L(G)$.*

Theorem 2.3.10 (see [91]). *For every context-sensitive grammar G, there is a context-sensitive grammar G' in the Penttonen normal form such that $L(G') = L(G)$.*

Observe that if G is a context-sensitive grammar in the Pentonnen normal form, then none of its rules is of the form (iv), which is not context-sensitive.

The next normal form limits the number of nonterminals and non-context-sensitive rules in phrase-structure grammars.

Definition 2.3.11. Let G be a phrase-structure grammar. G is in the *Geffert normal form* (see [29]) if it is of the form

$$G = \big(\{S, A, B, C, D\}, T, P \cup \{ABC \to \varepsilon\}, S\big)$$

where P contains context-free rules of the following three forms

$$\text{(i) } S \to uSa \qquad \text{(ii) } S \to uSv \qquad \text{(iii) } S \to uv$$

with $u \in \{A, AB\}^*$, $v \in \{BC, C\}^*$, and $a \in T$. □

Theorem 2.3.12 (see [29]). *For every recursively enumerable language K, there exists a phrase-structure grammar G in the Geffert normal form such that $L(G) = K$. In addition, every successful derivation in G is of the form $S \Rightarrow_G^* w_1w_2w$ by rules from P, where $w_1 \in \{A, AB\}^*$, $w_2 \in \{BC, C\}^*$, $w \in T^*$, and $w_1w_2w \Rightarrow_G^* w$ is derived by $ABC \to \varepsilon$.*

In the remainder of the present chapter, we recall the definition of several other types of grammars that are needed throughout this thesis.

Random Context Grammars

In essence, random context grammars (see Section 1.1 in [16]) 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 symbols do not occur there.

Definition 2.3.13. A *random context grammar* is a quadruple

$$G = \big(N, T, P, S\big)$$

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

$$P \subseteq N \times \big(N \cup T\big)^* \times 2^N \times 2^N$$

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

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

For $(A \to x, U, W) \in P$, U and W are called the *permitting context* and the *forbidding context*, respectively. If $(A \to x, U, W) \in P$ implies that $|x| \geq 1$, then G is said to be *propagating*.

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.

The *direct derivation relation* over V^* is denoted by $\Rightarrow_G$ and defined as follows. Let $u, v \in V^*$ and $(A \to x, U, W) \in P$. Then,

$$uAv \Rightarrow_G uxv$$

if and only if

$$(A \to x, U, W) \in P, U \subseteq \mathrm{alph}(uAv), \text{ and } W \cap \mathrm{alph}(uAv) = \emptyset$$

Let $\Rightarrow_G^n$ and $\Rightarrow_G^*$ denote the nth power of $\Rightarrow_G$, for some $n \geq 0$, and the reflexive-transitive closure of $\Rightarrow_G$, respectively.

The *language* of G is denoted by $L(G)$ and defined as

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

The families of languages defined by permitting grammars, forbidding grammars, and random context grammars are denoted by **Per**, **For**, and **RC**, respectively. To indicate that only propagating grammars are considered, we use the upper index $-\varepsilon$. That is, $\mathbf{Per}^{-\varepsilon}$, $\mathbf{For}^{-\varepsilon}$, and $\mathbf{RC}^{-\varepsilon}$ denote the families of languages defined by propagating permitting grammars, propagating forbidding grammars, and propagating random context grammars, respectively.

Theorem 2.3.14. $\mathbf{CF} \subset \mathbf{Per}^{-\varepsilon} = \mathbf{Per} \subset \mathbf{RC}^{-\varepsilon} \subset \mathbf{CS} \subset \mathbf{RC} = \mathbf{RE}$

Proof. $\mathbf{CF} \subset \mathbf{Per}^{-\varepsilon} \subset \mathbf{RC}^{-\varepsilon}$ follows from Theorem 2.7 in Chapter 3 of [99]. $\mathbf{Per}^{-\varepsilon} = \mathbf{Per}$ follows from Theorem 1 in [111]. $\mathbf{RC}^{-\varepsilon} \subset \mathbf{CS} \subset \mathbf{RC} = \mathbf{RE}$ follows from Theorems 1.2.4, 1.2.5, and 1.4.5 in [16]. □

Theorem 2.3.15 (see Theorem 16 in [77]). $\mathbf{CF} \subset \mathbf{For}^{-\varepsilon} \subseteq \mathbf{For} \subset \mathbf{CS}$

Selective Substitution Grammars

Selective substitution grammars (see [39, 40, 95]) use context-free-like rules that have a terminal or a nonterminal on their left-hand sides. By using extremely simple languages, referred to as *selectors*, they specify symbols where one of them is rewritten during a derivation step and, in addition, place some restrictions on the context appearing before and after the rewritten symbol. Otherwise, they work by analogy with context-free grammars.

Definition 2.3.16. A *selective substitution grammar* (an *s-grammar* for short) is a quintuple

$$G = \big(V, T, P, S, K\big)$$

where V is the *total alphabet*, $T \subseteq V$ is an alphabet of *terminals*,

$$P \subseteq V \times V^*$$

is a finite relation called the set of *rules*, $S \in V - T$ is the *start symbol*, and K is a finite set of *selectors* of the form

$$X^*\overline{Y}Z^*$$

where $X, Y, Z \subseteq V$; in words, the barred symbols are said to be *activated*. If $A \to x \in P$ implies that $|x| \geq 1$, then G is said to be *propagating*.

The *direct derivation relation* over V^*, symbolically denoted by $\Rightarrow_G$, is defined as follows:

$$uAv \Rightarrow_G uxv$$

if and only if $A \to x \in P$ and $X^*\overline{Y}Z^* \in K$ such that $u \in X^*$, $A \in Y$, and $v \in Z^*$. Let $\Rightarrow_G^n$ and $\Rightarrow_G^*$ denote the nth power of $\Rightarrow_G$, for some $n \geq 0$, and the reflexive-transitive closure of $\Rightarrow_G$, respectively.

The *language* of G is denoted by $L(G)$ and defined as

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

The families of languages generated by s-grammars and propagating s-grammars are denoted by $\mathbf{S}$ and $\mathbf{S}^{-\varepsilon}$, respectively.

Scattered Context Grammars

The notion of a scattered context grammar G is based on sequences of context-free rules, according to which G can simultaneously rewrite several nonterminals during a single derivation step.

Definition 2.3.17. A *scattered context grammar* (see [35, 73]) is a quadruple

$$G = \big(N, T, P, S\big)$$

where N is an alphabet of *nonterminals*, T an alphabet of *terminals* ($N \cap T = \emptyset$), P is a finite set of *rules* of the form

$$(A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n)$$

where $n \geq 1$, $A_i \in N$, and $x_i \in (N \cup T)^*$, for all $i = 1, 2, \ldots, n$ (each rule may have different n), and $S \in N$ is the *start symbol*. Set $V = N \cup T$ to be the *total alphabet*. When every $(A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n) \in P$ satisfies that $|x_i| \geq 1$, for all $i = 1, 2, \ldots, n$, then G is said to be *propagating*.

The *direct derivation relation* over V^*, symbolically denoted by $\Rightarrow_G$, is defined as follows:

$$u \Rightarrow_G v$$

if and only if $(A_1, \ldots, A_n) \rightarrow (x_1, \ldots, x_n) \in P$ and

$$u = u_1 A_1 \ldots u_n A_n u_{n+1}$$
$$v = u_1 x_1 \ldots u_n x_n u_{n+1}$$

where $u_i \in V^*$, for all $i = 1, 2, \ldots, n+1$. Let $\Rightarrow_G^n$ and $\Rightarrow_G^*$ denote the nth power of $\Rightarrow_G$, for some $n \geq 0$, and the reflexive-transitive closure of $\Rightarrow_G$, respectively.

The *language* of G is denoted by $L(G)$ and defined as

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

The families of languages generated by scattered context grammars and propagating scattered context grammars are denoted by **SC** and $\mathbf{SC}^{-\varepsilon}$, respectively.

Theorem 2.3.18 (see Theorems 3.20, 3.21 in [73]). $\mathbf{CF} \subset \mathbf{SC}^{-\varepsilon} \subseteq \mathbf{CS} \subset \mathbf{SC} = \mathbf{RE}$

It is not known whether the inclusion $\mathbf{SC}^{-\varepsilon} \subseteq \mathbf{CS}$ is, in fact, an identity.

Part II
One-Sided Random Context Grammars

This part forms the heart of the thesis. It introduces a new variant of random context grammars, called *one-sided random context grammars*, and studies this variant from many points of view. Generative power, reduction, normal forms, leftmost derivations, generalized and parsing-related versions all belong between the studied topics. The present part consists of seven chapters.

Chapter 3 defines one-sided random context grammars and their variants, like one-sided permitting grammars and one-sided forbidding grammars, and illustrates them by examples.

Chapter 4 establishes relations between families of languages defined by one-sided random context grammars and some well-known language families. It, therefore, studies the generative power of these grammars.

Chapter 5 establishes four normal forms of one-sided random context grammars, in which rules satisfy some prescribed properties or format.

Chapter 6 studies the descriptional complexity of one-sided random context grammars. More specifically, it shows how to reduce the number of nonterminals and rules in these grammars without affecting their generative power.

Chapter 7 introduces three types of leftmost derivation restrictions placed upon one-sided random context grammars, and studies their effect to the generative power of these grammars.

Chapter 8 introduces and investigates generalized versions of one-sided random context grammars. More specifically, it studies one-sided forbidding grammars that may forbid occurrences of strings rather than just occurrences of single symbols.

Chapter 9 closes this part by introducing and investigating parsing-related variants of one-sided random context grammars, which may be applied in practice.

Chapter 3
Definitions and Examples

This three-section chapter defines one-sided random context grammars and their variants, and illustrates them by examples. More specifically, Section 3.1 gives formal definitions of these grammars, Section 3.2 illustrates them by several examples, and Section 3.3 presents a denotation of language families generated by these grammars.

3.1 Definitions

Without further ado, let us define one-sided random context grammars formally.

Definition 3.1.1. 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

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

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 3.1.2. 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 $(A \to x, U, W) \in P_L \cup P_R$. Then,

$$uAv \Rightarrow_G uxv$$

if and only if

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

or

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

Let $\Rightarrow_G^n$ and $\Rightarrow_G^*$ denote the nth power of $\Rightarrow_G$, for some $n \geq 0$, and the reflexive-transitive closure of $\Rightarrow_G$, respectively. □

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 3.1.3. 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 3.1.4. Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. 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 3.1.5. 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*. If $P_R = \emptyset$ and $(A \to x, U, W) \in P_L$ implies that $W = \emptyset$, then G is a *left permitting grammar* (see [10]). If $P_R = \emptyset$ and $(A \to x, U, W) \in P_L$ implies that $U = \emptyset$, then G is a *left forbidding grammar* (see [31]). Their propagating versions are defined analogously as the propagating version of one-sided random context grammars. □

3.2 Examples

Next, we illustrate the above definitions by three examples.

Example 3.2.1. 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

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

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

$$\big\{a^nb^nc^n \mid n \geq 0\big\}$$

□

Example 3.2.2. 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

$$(S \to AX, \emptyset, \emptyset) \qquad (A \to a, \emptyset, \emptyset) \qquad (X \to bc, \emptyset, \emptyset)$$
$$(A \to aB, \emptyset, \emptyset) \qquad (X \to bYc, \{B\}, \emptyset)$$
$$(B \to A, \emptyset, \emptyset) \qquad (Y \to X, \{A\}, \emptyset)$$

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[1]

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

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

Example 3.2.3. 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

[1] Notice that after X is rewritten to bc by $(X \to bc, \emptyset, \emptyset)$, more as can be generated by $(A \to aB, \emptyset, \emptyset)$. However, observe that this does not affect the generated language.

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

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

$$\left\{a^n b^n c^n \mid n \geq 0\right\}$$ □

3.3 Denotation of Language Families

Throughout the rest of this thesis, the language families under discussion are denoted in the following way. **ORC**, **OPer**, and **OFor** denote the language families generated by one-sided random context grammars, one-sided permitting grammars, and one-sided forbidding grammars, respectively. **LRC**, **LPer**, and **LFor** 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.

Chapter 4
Generative Power

In this chapter, we establish relations between the language families defined in the previous chapter and some well-known language families from Chapter 2. Most importantly, we show that one-sided random context grammars are equally powerful as random context grammars, and that propagating one-sided random context grammars are more powerful than propagating random context grammars.

The present chapter consists of three sections. First, Section 4.1 studies the generative power of one-sided random context grammars. Then, Section 4.2 investigates the power of one-sided forbidding grammars. Finally, Section 4.3 discusses one-sided permitting grammars and their generative power.

4.1 One-Sided Random Context Grammars

First, we consider one-sided random context grammars and their propagating versions. We prove that $\mathbf{ORC}^{-\varepsilon} = \mathbf{CS}$ and $\mathbf{ORC} = \mathbf{RE}$.

Lemma 4.1.1. $\mathbf{CS} \subseteq \mathbf{ORC}^{-\varepsilon}$

Proof. Let $G = (N, T, P, S)$ be a context-sensitive grammar. Without any loss of generality, making use of Theorem 2.3.10, we assume that G is in the Penttonen normal form. We next construct a propagating one-sided random context grammar H such that $L(H) = L(G)$. Set $\bar{N} = \{\bar{A} \mid A \in N\}$, $\hat{N} = \{\hat{A} \mid A \in N\}$, and $N' = N \cup \bar{N} \cup \hat{N}$. Define H as

$$H = \left(N', T, P_L, P_R, S\right)$$

with P_L and P_R constructed as follows:

(1) for each $A \to a \in P$, where $A \in N$ and $a \in T$, add $(A \to a, \emptyset, N')$ to P_L;
(2) for each $A \to BC \in P$, where $A, B, C \in N$, add $(A \to BC, \emptyset, \bar{N} \cup \hat{N})$ to P_L;
(3) for each $AB \to AC \in P$, where $A, B, C \in N$, add $(B \to C, \{\hat{A}\}, N)$ to P_L;
(4) for each $A \in N$, add $(A \to \bar{A}, \emptyset, N \cup \hat{N})$ and $(A \to \hat{A}, \emptyset, N \cup \hat{N})$ to P_L;

(5) for each $A \in N$, add $(\bar{A} \to A,\ \emptyset,\ \bar{N} \cup \hat{N})$ and $(\hat{A} \to A,\ \emptyset,\ \bar{N} \cup \hat{N})$ to P_R.

Before proving that $L(H) = L(G)$, we give an insight into the construction. The simulation of context-free rules of the form $A \to BC$, where $A, B, C \in N$, is performed directly by rules introduced in (2). H simulates context-sensitive rules—that is, rules of the form $AB \to AC$, where $A, B, C \in N$—as follows. H first rewrites all nonterminals to the left of an occurrence of A to their barred versions by rules from (4), starting from the leftmost nonterminal of the current sentential form. Then, it rewrites A to $\hat{A}$ by $(A \to \hat{A},\ \emptyset,\ N \cup \hat{N})$ from (4). After this, it rewrites B to C by $(B \to C,\ \{\hat{A}\},\ N)$ from (3). Finally, H rewrites $\hat{A}$ back to A and all barred nonterminals back to their corresponding original versions by rules from (5) in the right-to-left way.

To prevent $AAB \Rightarrow_H AaB \Rightarrow_H \hat{A}aB \Rightarrow_H AaC$, rules simulating $A \to a$, where $A \in A$ and $a \in T$, introduced in (1), can be used only if there are no nonterminals to the left of A. Therefore, a terminal can never appear between two nonterminals. Consequently, every sentential form generated by H is of the form $x_1 x_2$, where $x_1 \in T^*$ and $x_2 \in N'^*$.

To prove that $L(H) = L(G)$, we first prove three claims. The first claim shows that every $y \in L(G)$ can be generated by G in two stages; first, only nonterminals are generated, and then, all nonterminals are rewritten to terminals. To prove that $L(G) \subseteq L(H)$, it then suffices to show how H simulates these derivations of G.

Claim 1. Let $y \in L(G)$. Then, there exists a derivation $S \Rightarrow_G^* x \Rightarrow_G^* y$, where $x \in N^+$, and during $x \Rightarrow_G^* y$, G applies only rules of the form $A \to a$, $A \in N$, where $a \in T$.

Proof. Let $y \in L(G)$. Since there are no rules in P with symbols from T on their left-hand sides, we can always rearrange all the applications of the rules occurring in $S \Rightarrow_G^* y$ so the claim holds. □

The second claim shows how certain derivations of G are simulated by H. Together with the previous claim, it is used to demonstrate that $L(G) \subseteq L(H)$ later in the proof.

Claim 2. If $S \Rightarrow_G^n x$, where $x \in N^+$, for some $n \geq 0$, then $S \Rightarrow_H^* x$.

Proof. This claim is established by induction on $n \geq 0$.

Basis. Let $n = 0$. Then, for $S \Rightarrow_G^0 S$, there is $S \Rightarrow_H^0 S$, so the basis holds.

Induction Hypothesis. Suppose that there exists $m \geq 0$ such that the claim holds for all derivations of length ℓ, where $0 \leq \ell \leq m$.

Induction Step. Consider any derivation of the form

$$S \Rightarrow_G^{n+1} w$$

where $w \in N^+$. Since $n+1 \geq 1$, this derivation can be expressed as

$$S \Rightarrow_G^n x \Rightarrow_G w$$

for some $x \in N^+$. By the induction hypothesis, $S \Rightarrow_H^* x$.

Next, we consider all possible forms of $x \Rightarrow_G w$, covered by the following two cases—(i) and (ii).

(i) Let $A \to BC \in P$ and $x = x_1Ax_2$, where $A, B, C \in N$, and $x_1, x_2 \in N^*$. Then,

$$x_1Ax_2 \Rightarrow_G x_1BCx_2$$

By (2), $(A \to BC, \emptyset, \bar{N} \cup \hat{N}) \in P_L$, so

$$x_1Ax_2 \Rightarrow_H x_1BCx_2$$

which completes the induction step for (i).

(ii) Let $AB \to AC \in P$ and $x = x_1ABx_2$, where $A, B, C \in N$, and $x_1, x_2 \in N^*$. Then,

$$x_1ABx_2 \Rightarrow_G x_1BCx_2$$

Let $x_1 = X_1X_2 \cdots X_k$, where $X_i \in N$, for all i, $1 \leq i \leq k$, $k = |x_1|$. By (4), $(X_i \to \bar{X}_i, \emptyset, N \cup \hat{N}) \in P_L$, for all i, $1 \leq i \leq k$, so

$$\begin{aligned} X_1X_2 \cdots X_kABx_2 &\Rightarrow_H \bar{X}_1X_2 \cdots X_kABx_2 \\ &\Rightarrow_H \bar{X}_1\bar{X}_2 \cdots X_kABx_2 \\ &\;\;\vdots \\ &\Rightarrow_H \bar{X}_1\bar{X}_2 \cdots \bar{X}_kABx_2 \end{aligned}$$

Let $\bar{x}_1 = \bar{X}_1\bar{X}_2 \cdots \bar{X}_k$. By (4), $(A \to \hat{A}, \emptyset, N \cup \hat{N}) \in P_L$, so

$$\bar{x}_1ABx_2 \Rightarrow_H \bar{x}_1\hat{A}Bx_2$$

By (3), $(B \to C, \{\hat{A}\}, N) \in P_L$, so

$$\bar{x}_1\hat{A}Bx_2 \Rightarrow_H \bar{x}_1\hat{A}Cx_2$$

Finally, by (5), $(\hat{A} \to A, \emptyset, \bar{N} \cup \hat{N}), (\bar{X}_i \to X_i, \emptyset, \bar{N} \cup \hat{N}) \in P_R$, for all i, $1 \leq i \leq k$, so

$$\begin{aligned} \bar{X}_1\bar{X}_2 \cdots \bar{X}_k\hat{A}Cx_2 &\Rightarrow_H \bar{X}_1\bar{X}_2 \cdots \bar{X}_kACx_2 \\ &\Rightarrow_H \bar{X}_1\bar{X}_2 \cdots X_kACx_2 \\ &\;\;\vdots \\ &\Rightarrow_H \bar{X}_1X_2 \cdots X_kACx_2 \\ &\Rightarrow_H X_1X_2 \cdots X_kACx_2 \end{aligned}$$

which completes the induction step for (ii).

Observe that cases (i) and (ii) cover all possible forms of $x \Rightarrow_G w$. Thus, Claim 2 holds. □

Next, we prove how G simulates derivations of H. The following claim is used to prove that $L(H) \subseteq L(G)$ later in the proof. Set $V = N \cup T$ and $V' = N' \cup T$. Define the homomorphism τ from V'^* to V^* as $\tau(\bar{A}) = A$, $\tau(\hat{A}) = A$, and $\tau(A) = A$, for all $A \in N$, and $\tau(a) = a$, for all $a \in T$.

Claim 3. If $S \Rightarrow_H^n x$, where $x \in V'^+$, for some $n \geq 0$, then $S \Rightarrow_G^* \tau(x)$ and x is of the form $x'X_1X_2 \cdots X_h$, where $x' \in T^*$ and $X_i \in N'$, for all i, $1 \leq i \leq h$, for some $h \geq 0$. Furthermore, if $X_j \in \hat{N}$, for some j, $1 \leq j \leq h$, then $X_k \in \bar{N}$, for all k, $1 \leq k < j$, and $X_l \in N$, for all l, $j < l \leq h$.

Proof. This claim is established by induction on $n \geq 0$.

Basis. Let $n = 0$. Then, for $S \Rightarrow_H^0 S$, there is $S \Rightarrow_G^0 S$, so the basis holds.

Induction Hypothesis. Suppose that there exists $m \geq 0$ such that the claim holds for all derivations of length ℓ, where $0 \leq \ell \leq m$.

Induction Step. Consider any derivation of the form

$$S \Rightarrow_H^{n+1} w$$

where $w \in V'^+$. Since $n+1 \geq 1$, this derivation can be expressed as

$$S \Rightarrow_H^n x \Rightarrow_H w$$

for some $x \in V'^+$. By the induction hypothesis, $S \Rightarrow_G^* \tau(x)$ and x is of the form $x'X_1X_2 \cdots X_h$, where $x' \in T^*$ and $X_i \in N'$, for all i, $1 \leq i \leq h$, for some $h \geq 0$. Furthermore, if $X_j \in \hat{N}$, for some j, $1 \leq j \leq h$, then $X_k \in \bar{N}$, for all k, $1 \leq k < j$, and $X_l \in N$, for all l, $j < l \leq h$.

Next, we consider all possible forms of $x \Rightarrow_H w$, covered by the following four cases—(i) through (iv). That is, we show how rules introduced to P_L and P_R in (1) through (5) are simulated by G. The last case covers the simulation of rules from both (4) and (5) because these rules are simulated in the same way.

(i) Let $x'X_1X_2 \cdots X_h \Rightarrow_H x'aX_2 \cdots X_h$ by $(X_1 \rightarrow a,\ \emptyset,\ N') \in P_L$, introduced in (1) from $X_1 \rightarrow a \in P$, where $a \in T$. Then,

$$x'\tau(X_1X_2 \cdots X_h) \Rightarrow_G x'\tau(aX_2 \cdots X_h)$$

which completes the induction step for (i).

(ii) Let $x'X_1X_2 \cdots X_{j-1}X_jX_{j+1} \cdots X_h \Rightarrow_H x'X_1X_2 \cdots X_{j-1}BCX_{j+1} \cdots X_h$ by $(X_j \rightarrow BC,\ \emptyset,\ \bar{N} \cup \hat{N}) \in P_L$, introduced in (2) from $X_j \rightarrow BC \in P$, for some j, $1 \leq j \leq h$, and $B, C \in N$. Then,

$$x'\tau(X_1X_2 \cdots X_{j-1}X_jX_{j+1} \cdots X_h) \Rightarrow_G x'\tau(X_1X_2 \cdots X_{j-1}BCX_{j+1} \cdots X_h)$$

which completes the induction step for (ii).

(iii) Let $x'X_1X_2\cdots X_{j-1}X_jX_{j+1}\cdots X_h \Rightarrow_H x'X_1X_2\cdots X_{j-1}CX_{j+1}\cdots X_h$ by $(X_j \to C, \{\hat{A}\}, N) \in P_L$, introduced in (3) from $AX_j \to AC \in P$, for some j, $2 \le j \le h$, and $A, C \in N$. By the induction hypothesis, $X_{j-1} = \hat{A}$. Therefore,

$$x'\tau(X_1X_2\cdots X_{j-1}X_jX_{j+1}\cdots X_h) \Rightarrow_G x'\tau(X_1X_2\cdots X_{j-1}CX_{j+1}\cdots X_h)$$

which completes the induction step for (iii).

(iv) Let $x'X_1X_2\cdots X_{j-1}X_jX_{j+1}\cdots X_h \Rightarrow_H x'X_1X_2\cdots X_{j-1}X_j'X_{j+1}\cdots X_h$ by $(X_j \to X_j', \emptyset, W) \in P_L \cup P_R$, introduced in (4) or (5), for some j, $1 \le j \le h$, where X_j' depends on the particular rule that was used. Then,

$$x'\tau(X_1X_2\cdots X_h) \Rightarrow_G^0 x'\tau(X_1X_2\cdots X_h)$$

which completes the induction step for (iv).

Observe that cases (i) through (iv) cover all possible forms of $x \Rightarrow_H w$. Thus, Claim 3 holds. □

We next prove that $L(H) = L(G)$. Let $y \in L(G)$. Then, by Claim 1, there is $S \Rightarrow_G^* x \Rightarrow_G^* y$ such that $x \in N^+$ and during $x \Rightarrow_G^* y$, G uses only rules of the form $A \to a$, where $A \in N$, $a \in T$. By Claim 2, $S \Rightarrow_H^* x$. Let $x = A_1A_2\cdots A_k$ and $y = a_1a_2\cdots a_k$, where $A_i \in N$, $A_i \to a_i \in P$, $a_i \in T$, for all i, $1 \le i \le k$, $k = |x|$. By (1), $(A_i \to a_i, \emptyset, N') \in P_L$, for all i, $1 \le i \le k$, so

$$\begin{aligned} A_1A_2\cdots A_k &\Rightarrow_H a_1A_2\cdots A_k \\ &\Rightarrow_H a_1a_2\cdots A_k \\ &\;\;\vdots \\ &\Rightarrow_H a_1a_2\cdots a_k \end{aligned}$$

Consequently, $y \in L(G)$ implies that $y \in L(H)$, so $L(G) \subseteq L(H)$.

Consider Claim 3 for $x \in T^+$. Then, $x \in L(H)$ implies that $\tau(x) = x \in L(G)$, so $L(H) \subseteq L(G)$. Since $L(G) \subseteq L(H)$ and $L(H) \subseteq L(G)$, $L(H) = L(G)$, so Lemma 4.1.1 holds. □

Lemma 4.1.2. $\mathbf{ORC}^{-\varepsilon} \subseteq \mathbf{CS}$

Proof. Since the length of sentential forms in derivations of propagating one-sided random context grammars is nondecreasing, propagating one-sided random context grammars can be simulated by context-sensitive grammars. A rigorous proof of this lemma is left to the reader. □

Theorem 4.1.3. $\mathbf{ORC}^{-\varepsilon} = \mathbf{CS}$

Proof. This theorem follows from Lemmas 4.1.1 and 4.1.2. □

Theorem 4.1.4. ORC = RE

Proof. The inclusion **ORC** ⊆ **RE** follows from Church's thesis. **RE** ⊆ **ORC** can be proved by analogy with the proof of Lemma 4.1.1. Observe that by Theorem 2.3.9, G can additionally contain rules of the form $A \to \varepsilon$, where $A \in N$. We can simulate these context-free rules in the same way we simulate $A \to BC$, where $A, B, C \in N$—that is, for each $A \to \varepsilon \in P$, we introduce $(A \to \varepsilon, \emptyset, \bar{N} \cup \hat{N})$ to P_L. □

4.2 One-Sided Forbidding Grammars

Next, we consider one-sided forbidding grammars. First, we prove that **OFor** = **S** and $\mathbf{OFor}^{-\varepsilon} = \mathbf{S}^{-\varepsilon}$. Then, we show that one-sided forbidding grammars with the set of left forbidding rules coinciding with the set of right forbidding rules characterize only the family of context-free languages. This characterization also holds in terms of left forbidding grammars. Indeed, it holds that $\mathbf{LFor}^{-\varepsilon} = \mathbf{LFor} = \mathbf{CF}$.

Lemma 4.2.1. *For every s-grammar G, there is a one-sided forbidding grammar H such that* $L(H) = L(G)$.

Proof. Let $G = (V, T, P, S, K)$ be an s-grammar. We next construct a one-sided forbidding grammar H such that $L(H) = L(G)$. Set $N = V - T$, $\hat{T} = \{\hat{a} \mid a \in T\}$, $T_1 = \{\langle a, 1\rangle \mid a \in T\}$, $T_2 = \{\langle a, 2\rangle \mid a \in T\}$, $T_{12} = T_1 \cup T_2$, and

$$M_{12} = \left\{\langle r, s, i\rangle \mid r \in P, s = (X^* \overline{Y} Z^*) \in K, i = 1, 2\right\}$$

Without any loss of generality, we assume that N, T, $\hat{T}$, T_1, T_2, and M_{12} are pairwise disjoint. Construct

$$H = \left(N', T, P_L, P_R, S\right)$$

as follows. Initially, set

$$N' = N \cup \hat{T} \cup T_{12} \cup M_{12}$$
$$P_L = \emptyset$$
$$P_R = \emptyset$$

Define the homomorphism τ from V^* to N'^* as $\tau(A) = A$, for all $A \in N$, and $\tau(a) = \hat{a}$, for all $a \in T$. Define the function $\mathscr{T}$ from 2^V to $2^{N'}$ as $\mathscr{T}(\emptyset) = \emptyset$ and

$$\mathscr{T}\left(\{A_1, \ldots, A_n\}\right) = \left\{\tau(A_1), \ldots, \tau(A_n)\right\}$$

Perform (1) and (2), given next.

(1) For each $s = (X^* \overline{Y} Z^*) \in K$ and each $A \in Y$ such that $r = (A \to y) \in P$,

(1.1) add $(\tau(A) \to \langle r, s, 1\rangle, \emptyset, N' - \mathscr{T}(X))$ to P_L;

(1.2) add $(\langle r,s,1\rangle \to \langle r,s,2\rangle, \emptyset, N' - \mathscr{T}(Z))$ to P_R;
(1.3) add $(\langle r,s,2\rangle \to \tau(y), \emptyset, T_{12} \cup M_{12})$ to P_L.

(2) For each $a \in T$,

(2.1) add $(\hat{a} \to \langle a,1\rangle, \emptyset, N' - \hat{T})$ to P_L;
(2.2) add $(\langle a,1\rangle \to \langle a,2\rangle, \emptyset, N' - T_2)$ to P_R;
(2.3) add $(\langle a,2\rangle \to a, \emptyset, N')$ to P_L.

Before proving that $L(H) = L(G)$, let us informally describe (1) and (2). Let $X^*\overline{Y}Z^* \in K$ and $x_1Ax_2 \in V^*$, where $x_1, x_2 \in V^*$ and $A \in Y$. Observe that $x_1 \in X^*$ if and only if $(V-X) \cap \text{alph}(x_1) = \emptyset$, and $x_2 \in Z^*$ if and only if $(V-Z) \cap \text{alph}(x_2) = \emptyset$. Therefore, to simulate the application of $A \to y \in P$ in H, we first check the absence of all symbols from $V - X$ to the left of A, and then, we check the absence of all symbols from $V - Z$ to the right of A.

We need to guarantee the satisfaction of the following two conditions. First, we need to make sure that only a single rule is simulated at a time. For this purpose, we have the three-part construction of rules in (1). Examine it to see that whenever H tries to simultaneously simulate more than one rule of G, the derivation is blocked. Second, as opposed to s-grammars, one-sided forbidding grammars can neither rewrite terminals nor forbid their occurrence. To circumvent this restriction, rules introduced in (1) rewrite and generate hatted terminals which act as nonterminals. For example, $a \to bDc \in P$, where $a,b,c \in T$ and $D \in N$, is simulated by $\hat{a} \to \hat{b}D\hat{c}$ in H.

Hatted terminals can be rewritten to terminals by rules introduced in (2). Observe that this can be done only if there are no symbols from $N' - \hat{T}$ present in the current sentential form; otherwise, the derivation is blocked. Furthermore, observe that after a rule from (2.1) is applied, no rule of G can be simulated anymore. Based on these observations, we see that every successful derivation of $a_1a_2\cdots a_h$ in H is of the form

$$S \Rightarrow_H^* \hat{a}_1\hat{a}_2\cdots\hat{a}_h \Rightarrow_H^* a_1a_2\cdots a_h$$

and during $S \Rightarrow_H^* \hat{a}_1\hat{a}_2\cdots\hat{a}_h$, no sentential form contains any symbols from T.

To establish the identity $L(H) = L(G)$, we prove two claims. First, Claim 1 shows how derivations of G are simulated by H. Then, Claim 2 demonstrates the converse—that is, it shows how derivations of H are simulated by G.

Claim 1. If $S \Rightarrow_G^n w \Rightarrow_G^* z$, where $w \in V^*$ and $z \in T^*$, for some $n \geq 0$, then $S \Rightarrow_H^* \tau(w)$.

Proof. This claim is established by induction on $n \geq 0$.

Basis. For $n = 0$, this claim obviously holds.

Induction Hypothesis. Suppose that there exists $n \geq 0$ such that the claim holds for all derivations of length ℓ, where $0 \leq \ell \leq n$.

Induction Step. Consider any derivation of the form

$$S \Rightarrow_G^{n+1} w \Rightarrow_G^* z$$

where $w \in V^*$ and $z \in T^*$. Since $n+1 \geq 1$, this derivation can be expressed as

$$S \Rightarrow_G^n x \Rightarrow_G w \Rightarrow_G^* z$$

for some $x \in V^+$. Let $x = x_1Ax_2$ and $w = x_1yx_2$ so $s = (X^*\overline{Y}Z^*) \in K$ such that $x_1 \in X^*$, $A \in Y$, $x_2 \in Z^*$, and $r = (A \to y) \in P$.

By the induction hypothesis, $S \Rightarrow_H^* \tau(x)$. By (1),

$$\begin{aligned}
&(\tau(A) \to \langle r,s,1\rangle, \emptyset, N' - \mathscr{T}(X)) \in P_L\\
&(\langle r,s,1\rangle \to \langle r,s,2\rangle, \emptyset, N' - \mathscr{T}(Z)) \in P_R\\
&(\langle r,s,2\rangle \to \tau(y), \emptyset, T_{12} \cup M_{12}) \in P_L
\end{aligned}$$

By the induction hypothesis and by $x_1Ax_2 \Rightarrow_G x_1yx_2$, $\tau(x) = \tau(x_1)\tau(A)\tau(x_2)$, $(N' - \mathscr{T}(X)) \cap \mathrm{alph}(\tau(x_1)) = \emptyset$, and $(N' - \mathscr{T}(Z)) \cap \mathrm{alph}(\tau(x_2)) = \emptyset$, so

$$\begin{aligned}
\tau(x_1)\tau(A)\tau(x_2) &\Rightarrow_H \tau(x_1)\langle r,s,1\rangle\tau(x_2)\\
&\Rightarrow_H \tau(x_1)\langle r,s,2\rangle\tau(x_2)\\
&\Rightarrow_H \tau(x_1)\tau(y)\tau(x_2)
\end{aligned}$$

Since $\tau(w) = \tau(x_1)\tau(y)\tau(x_2)$, the induction step is completed. □

Set $V' = N' \cup T$. Define the homomorphism ψ from V'^* to V^* as $\psi(A) = A$, for all $A \in N$, $\psi(\langle r,s,1\rangle) = \psi(\langle r,s,2\rangle) = A$, for all $r = (A \to x) \in P$ and all $s = (X^*\overline{Y}Z^*) \in K$, and $\psi(\langle a,1\rangle) = \psi(\langle a,2\rangle) = \psi(\hat{a}) = a$, for all $a \in T$.

Claim 2. If $S \Rightarrow_H^n w \Rightarrow_H^* z$, where $w \in V'^*$ and $z \in T^*$, for some $n \geq 0$, then $S \Rightarrow_G^* \psi(w)$.

Proof. This claim is established by induction on $n \geq 0$.

Basis. For $n = 0$, this claim obviously holds.

Induction Hypothesis. Suppose that there exists $n \geq 0$ such that the claim holds for all derivations of length ℓ, where $0 \leq \ell \leq n$.

Induction Step. Consider any derivation of the form

$$S \Rightarrow_H^{n+1} w \Rightarrow_H^* z$$

where $w \in V'^*$ and $z \in T^*$. Since $n+1 \geq 1$, this derivation can be expressed as

$$S \Rightarrow_H^n x \Rightarrow_H w \Rightarrow_H^* z$$

for some $x \in V'^{+}$. By the induction hypothesis, $S \Rightarrow_G^* \psi(x)$. Observe that if $x \Rightarrow_H w$ is derived by a rule introduced in (1.1), (1.2), or in (2), then the induction step follows directly from the induction hypothesis. Therefore, assume that $x = x_1\langle r,s,2\rangle x_2$, $w = x_1\tau(y)x_2$, and $(\langle r,s,2\rangle \rightarrow \tau(y), \emptyset, T_{12} \cup M_{12}) \in P_L$, introduced in (1.3) from $s = (X^*\overline{Y}Z^*) \in K$ such that $A \in Y$ and $r = (A \rightarrow y) \in P$. Recall that $(T_{12} \cup M_{12}) \cap \mathrm{alph}(x_1) = \emptyset$ has to hold; otherwise, the rule is not applicable. Next, we argue that $\psi(x) \Rightarrow_G \psi(w)$.

Observe that the two other rules from (1.1) and (1.2) have to be applied before $(\langle r,s,2\rangle \rightarrow \tau(y), \emptyset, T_{12} \cup M_{12})$ is applicable. Therefore, $S \Rightarrow_H^* x$ has to be of the form

$$S \Rightarrow_H^* v_1\tau(A)v_2 \Rightarrow_H v_1\langle r,s,1\rangle v_2 \Rightarrow_H^* x_1\langle r,s,2\rangle x_2 = x$$

where $v_1, v_2 \in V'^*$ and $x_1\langle r,s,2\rangle x_2$ is the first sentential form in $S \Rightarrow_H^* x$, where an occurrence of $\langle r,s,1\rangle$ is rewritten to $\langle r,s,2\rangle$. We next argue, by contradiction, that (i) $v_1 = x_1$ and (ii) $v_2 = x_2$. We then use (i) and (ii) to show that $\psi(x) \Rightarrow_G \psi(w)$.

(i) Assume that $v_1 \neq x_1$. The only possible way this case could happen is that after

$$S \Rightarrow_H^* v_1\tau(A)v_2 \Rightarrow_H v_1\langle r,s,1\rangle v_2$$

v_1 is rewritten by a rule t. Since $\langle r,s,1\rangle$ occurs to the right of v_1 and since it is generated by $(\tau(A) \rightarrow \langle r,s,1\rangle, \emptyset, N' - \mathscr{T}(X)) \in P_L$ from (1.1), t is necessarily a rule from (1.1) or (2.1). However, in either case, we cannot rewrite the compound nonterminal generated by t because $\langle r,s,1\rangle$ is still present to the right of v_1. Although we can rewrite $\langle r,s,1\rangle$ to $\langle r,s,2\rangle$ by $(\langle r,s,1\rangle \rightarrow \langle r,s,2\rangle, \emptyset, N' - \mathscr{T}(Z)) \in P_R$, we cannot rewrite $\langle r,s,2\rangle$ because of the generated nonterminal. This contradicts the assumption that $v_1 \neq x_1$. Therefore, $v_1 = x_1$.

(ii) Assume that $v_2 \neq x_2$. The only possible way this case could happen is that after

$$S \Rightarrow_H^* v_1\tau(A)v_2 \Rightarrow_H v_1\langle r,s,1\rangle v_2$$

v_2 is rewritten by a rule t. Since $\langle r,s,1\rangle$ occurs to the left of v_2, t is necessarily a rule from (1.2) or (2.2). However, in either case, we cannot rewrite the compound nonterminal generated by t because $\langle r,s,1\rangle$ is still present to the left of v_2. Furthermore, we cannot rewrite $\langle r,s,1\rangle$ by $(\langle r,s,1\rangle \rightarrow \langle r,s,2\rangle, \emptyset, N' - \mathscr{T}(Z)) \in P_R$ from (1.2) because of the generated nonterminal. This contradicts the assumption that $v_2 \neq x_2$. Therefore, $v_2 = x_2$.

In a similar way, $\mathrm{alph}(x) \cap T = \emptyset$ can be demonstrated. Consequently, $(V' - \mathscr{T}(X)) \cap \mathrm{alph}(x_1) = \emptyset$ and $(V' - \mathscr{T}(Z)) \cap \mathrm{alph}(x_2) = \emptyset$. Recall that $X^*\overline{Y}Z^* \in K$, $A \in Y$, and $A \rightarrow y \in P$. Since $\psi(x) = \psi(x_1\langle r,s,2\rangle x_2) = \psi(x_1)A\psi(x_2)$, $\mathrm{alph}(\psi(x_1)) \subseteq X$, $A \in Y$, and $\mathrm{alph}(\psi(x_2)) \subseteq Z$, we see that $\psi(x) \Rightarrow_G \psi(w)$, which completes the induction step. □

Next, we prove that $L(H) = L(G)$. Consider Claim 1 for $w \in T^*$. Then, $S \Rightarrow_G^* w$ implies that $S \Rightarrow_H^* \tau(w)$. Let $\tau(w) = \hat{a}_1\hat{a}_2 \cdots \hat{a}_h$, where $h = |w|$ (the case when $h = 0$ means $w = \varepsilon$). By (2),

$$\begin{aligned}
&(\hat{a}_i \to \langle a_i, 1\rangle, \emptyset, N' - \hat{T}) \in P_L \\
&(\langle a_i, 1\rangle \to \langle a_i, 2\rangle, \emptyset, N' - T_2) \in P_R \\
&(\langle a_i, 2\rangle \to a_i, \emptyset, N') \in P_L
\end{aligned}$$

for all i, $1 \le i \le h$. Therefore,

$$\begin{aligned}
\hat{a}_1 \cdots \hat{a}_{h-1}\hat{a}_h &\Rightarrow_H \hat{a}_1 \cdots \hat{a}_{h-1}\langle a_h, 1\rangle \\
&\Rightarrow_H \hat{a}_1 \cdots \langle a_{h-1}, 1\rangle\langle a_h, 1\rangle \\
&\ \ \vdots \\
&\Rightarrow_H \langle a_1, 1\rangle \cdots \langle a_{h-1}, 1\rangle\langle a_h, 1\rangle \\
&\Rightarrow_H \langle a_1, 1\rangle \cdots \langle a_{h-1}, 1\rangle\langle a_h, 2\rangle \\
&\Rightarrow_H \langle a_1, 1\rangle \cdots \langle a_{h-1}, 2\rangle\langle a_h, 2\rangle \\
&\ \ \vdots \\
&\Rightarrow_H \langle a_1, 2\rangle\langle a_2, 2\rangle \cdots \langle a_h, 2\rangle \\
&\Rightarrow_H a_1\langle a_2, 2\rangle \cdots \langle a_h, 2\rangle \\
&\Rightarrow_H a_1a_2 \cdots \langle a_h, 2\rangle \\
&\ \ \vdots \\
&\Rightarrow_H a_1a_2 \cdots a_h
\end{aligned}$$

Hence, $L(G) \subseteq L(H)$. Consider Claim 2 for $w \in T^*$. Then, $S \Rightarrow_H^* w$ implies that $S \Rightarrow_G^* \psi(w) = w$, so $L(H) \subseteq L(G)$. Consequently, $L(H) = L(G)$, and the lemma holds. □

Lemma 4.2.2. *For every one-sided forbidding grammar G, there is an s-grammar H such that $L(H) = L(G)$.*

Proof. Let $G = (N, T, P_L, P_R, S)$ be a one-sided forbidding grammar. We next construct an s-grammar H such that $L(H) = L(G)$. Set $V = N \cup T$ and

$$M = \big\{ \langle r, L\rangle \mid r = (A \to y, \emptyset, F) \in P_L \big\} \cup \big\{ \langle r, R\rangle \mid r = (A \to y, \emptyset, F) \in P_R \big\}$$

Without any loss of generality, we assume that $V \cap M = \emptyset$. Construct

$$H = \big(V', T, P', S, K\big)$$

as follows. Initially, set $V' = V \cup M$, $P' = \emptyset$, and

$$K = \big\{ V^* \overline{\{A\}} V^* \mid A \in N \big\}$$

Perform (1) and (2), given next.

(1) For each $r = (A \to y, \emptyset, F) \in P_L$,

(1.1) add $A \to \langle r, L\rangle$ and $\langle r, L\rangle \to y$ to P';
(1.2) add $(V - F)^* \overline{\{\langle r, L\rangle\}} V^*$ to K.

(2) For each $r = (A \to y, \emptyset, F) \in P_R$,

(2.1) add $A \to \langle r, R\rangle$ and $\langle r, R\rangle \to y$ to P';
(2.2) add $V^* \overline{\{\langle r, R\rangle\}} (V - F)^*$ to K.

Before proving that $L(H) = L(G)$, let us informally explain (1) and (2). Since a rule $r = (A \to y, \emptyset, F)$ can be in both P_L and P_R and since there can be several rules with A on their left-hand sides, we simulate the application of a single rule of G in two steps. First, depending on whether $r \in P_L$ or $r \in P_r$, we rewrite an occurrence of A to a special compound nonterminal $\langle r, s\rangle$, which encodes the simulated rule r and the side on which we check the absence of forbidding symbols ($s = L$ or $s = R$). Then, we introduce a selector which checks the absence of all symbols from F to the proper side of $\langle r, s\rangle$, depending on whether $s = L$ or $s = R$.

To establish the identity $L(H) = L(G)$, we prove two claims. First, Claim 1 shows how derivations of G are simulated by H. Then, Claim 2 demonstrates the converse—that is, it shows how derivations of H are simulated by G.

Claim 1. If $S \Rightarrow_G^n w \Rightarrow_G^* z$, where $w \in V^*$ and $z \in T^*$, for some $n \geq 0$, then $S \Rightarrow_H^* w$.

Proof. This claim is established by induction on $n \geq 0$.

Basis. For $n = 0$, this claim obviously holds.

Induction Hypothesis. Suppose that there exists $n \geq 0$ such that the claim holds for all derivations of length ℓ, where $0 \leq \ell \leq n$.

Induction Step. Consider any derivation of the form

$$S \Rightarrow_G^{n+1} w \Rightarrow_G^* z$$

where $w \in V^*$ and $z \in T^*$. Since $n + 1 \geq 1$, this derivation can be expressed as

$$S \Rightarrow_G^n x \Rightarrow_G w \Rightarrow_G^* z$$

for some $x \in V^+$. By the induction hypothesis, $S \Rightarrow_H^* x$.

Next, we consider all possible forms of $x \Rightarrow_G w$, covered by the following two cases—(i) and (ii).

(i) *Application of* $(A \to y, \emptyset, F) \in P_L$. Let $x = x_1 A x_2$, $w = x_1 y x_2$, and $r = (A \to y, \emptyset, F) \in P_L$, so $x \Rightarrow_G w$ by r. This implies that $\mathrm{alph}(x_1) \cap F = \emptyset$. By the initialization part of the construction, $V^* \overline{\{A\}} V^* \in K$, and by (1.1), $A \to \langle r, L\rangle \in P'$. Since $\mathrm{alph}(x_1) \subseteq V$ and $\mathrm{alph}(x_2) \subseteq V$,

$$x_1 A x_2 \Rightarrow_H x_1 \langle r, L\rangle x_2$$

By (1.2), $(V-F)^*\overline{\{\langle r,L\rangle\}}V^* \in K$, and by (1.1), $\langle r,L\rangle \rightarrow y \in P'$. Since $\text{alph}(x_1) \cap F = \emptyset$,

$$x_1 \langle r, L\rangle x_2 \Rightarrow_H x_1 y x_2$$

which completes the induction step for (i).

(ii) *Application of* $(A \rightarrow y,\ \emptyset,\ F) \in P_R$. Proceed by analogy with (i), but use rules from (2) instead of rules from (1).

Observe that cases (i) and (ii) cover all possible forms of $x \Rightarrow_G w$. Thus, the claim holds. ☐

Define the homomorphism φ from V'^* to V^* as $\varphi(A) = A$, for all $A \in N$, $\varphi(\langle r,L\rangle) = A$, for all $r = (A \rightarrow x,\ \emptyset,\ F) \in P_L$, $\varphi(\langle r,R\rangle) = A$, for all $r = (A \rightarrow x,\ \emptyset,\ F) \in P_R$, and $\varphi(a) = a$, for all $a \in T$.

Claim 2. If $S \Rightarrow_H^n w \Rightarrow_H^* z$, *where* $w \in V'^*$ *and* $z \in T^*$, *for some* $n \geq 0$, *then* $S \Rightarrow_G^* \varphi(w)$.

Proof. This claim is established by induction on $n \geq 0$.

Basis. For $n = 0$, this claim obviously holds.

Induction Hypothesis. Suppose that there exists $n \geq 0$ such that the claim holds for all derivations of length ℓ, where $0 \leq \ell \leq n$.

Induction Step. Consider any derivation of the form

$$S \Rightarrow_H^{n+1} w \Rightarrow_H^* z$$

where $w \in V'^*$ and $z \in T^*$. Since $n+1 \geq 1$, this derivation can be expressed as

$$S \Rightarrow_H^n x \Rightarrow_H w \Rightarrow_H^* z$$

for some $x \in V'^+$. By the induction hypothesis, $S \Rightarrow_G^* \varphi(x)$. Next, we consider all possible forms of $x \Rightarrow_H w$, covered by the following four cases—(i) through (iv).

(i) *Application of* $A \rightarrow \langle r,L\rangle \in P'$, *introduced in (1.1).* Let $x = x_1 A x_2$, $w = x_1\langle r,L\rangle x_2$, so $x \Rightarrow_H w$ by $A \rightarrow \langle r,L\rangle \in P'$, introduced in (1.1) from $r = (A \rightarrow y,\ \emptyset,\ F) \in P_L$. Since $\varphi(\langle r,L\rangle) = A$, the induction step for (i) follows directly from the induction hypothesis.

(ii) *Application of* $A \rightarrow \langle r,R\rangle \in P'$, *introduced in (2.1).* Proceed by analogy with (i).

(iii) *Application of* $\langle r,L\rangle \rightarrow y \in P'$, *introduced in (1.1).* Let $x = x_1\langle r,L\rangle x_2$, $w = x_1 y x_2$, $(V-F)^*\overline{\{\langle r,L\rangle\}}V^* \in K$ such that $x_1 \in (V-F)^*$ and $x_2 \in V^*$, so $x \Rightarrow_H w$ by $\langle r,L\rangle \rightarrow y \in P'$, introduced in (1.1) from $r = (A \rightarrow y,\ \emptyset,\ F) \in P_L$. Observe that $x_1 \in (V-F)^*$ implies that $\varphi(x_1) = x_1$ and $\text{alph}(x_1) \cap F = \emptyset$. Since $\varphi(x) = \varphi(x_1\langle r,L\rangle x_2) = x_1 A \varphi(x_2)$,

$$x_1 A\varphi(x_2) \Rightarrow_G x_1 y\varphi(x_2) \text{ by } r$$

As $\varphi(w) = \varphi(x_1yx_2) = x_1y\varphi(x_2)$, the induction step is completed for (iii).

(iv) *Application of $\langle r, R\rangle \to y \in P'$, introduced in (2.1).* Proceed by analogy with (iii).

Observe that cases (i) through (iv) cover all possible forms of $x \Rightarrow_H w$. Thus, the claim holds. □

Next, we prove that $L(H) = L(G)$. Consider Claim 1 for $w \in T^*$. Then, $S \Rightarrow_G^* w$ implies that $S \Rightarrow_H^* w$. Hence, $L(G) \subseteq L(H)$. Consider Claim 2 for $w \in T^*$. Then, $S \Rightarrow_H^* w$ implies that $S \Rightarrow_G^* \varphi(w) = w$, so $L(H) \subseteq L(G)$. Consequently, $L(H) = L(G)$, and the theorem holds. □

Theorem 4.2.3. OFor = S

Proof. This theorem follows from Lemmas 4.2.1 and 4.2.2. □

Theorem 4.2.4. $\mathbf{OFor}^{-\varepsilon} = \mathbf{S}^{-\varepsilon}$

Proof. Reconsider the proof of Lemma 4.2.1. Observe that if G is propagating, then so is H. Hence, $\mathbf{S}^{-\varepsilon} \subseteq \mathbf{OFor}^{-\varepsilon}$. Reconsider the proof of Lemma 4.2.2. Observe that if G is propagating, then so is H. Hence, $\mathbf{OFor}^{-\varepsilon} \subseteq \mathbf{S}^{-\varepsilon}$, and the theorem holds. □

We next turn our attention to one-sided forbidding grammars with the set of left forbidding rules coinciding with the set of right forbidding rules. We prove that they characterize the family of context-free languages.

Lemma 4.2.5. *Let K be a context-free language. Then, there exists a one-sided forbidding grammar, $G = (N, T, P_L, P_R, S)$, satisfying $P_L = P_R$ and $L(G) = K$.*

Proof. Let K be a context-free language. Then, there exists a context-free grammar, $H = (N, T, P, S)$, such that $L(H) = K$. Define the one-sided forbidding grammar

$$G = \left(N, T, P', P', S\right)$$

where

$$P' = \left\{(A \to x, \emptyset, \emptyset) \mid A \to x \in P\right\}$$

Clearly, $L(G) = L(H) = K$, so the lemma holds. □

Lemma 4.2.6. *Let $G = (N, T, P_L, P_R, S)$ be a one-sided forbidding grammar satisfying $P_L = P_R$. Then, $L(G)$ is context-free.*

Proof. Let $G = (N, T, P_L, P_R, S)$ be a one-sided forbidding grammar satisfying $P_L = P_R$. Define the context free grammar $H = (N, T, P', S)$ with

$$P' = \left\{A \to x \mid (A \to x, \emptyset, F) \in P_L\right\}$$

Observe that since $P_L = P_R$, in the construction of P' above, it is sufficient to consider just the rules from P_L. As any successful derivation in G is also a successful derivation in H, the inclusion $L(G) \subseteq L(H)$ holds. On the other hand, let $w \in L(H)$ be a string successfully generated by H. Then, there exists a successful leftmost derivation of w in H (see Theorem 2.3.7). Observe that such a leftmost derivation is also possible in G because the leftmost nonterminal can always be rewritten. Indeed, P' contains only rules originating from the rules in P_L and all rules in P_L are applicable to the leftmost nonterminal. Thus, the other inclusion $L(H) \subseteq L(G)$ holds as well, which completes the proof. □

Theorem 4.2.7. *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 $P_L = P_R$.*

Proof. This theorem follows from Lemmas 4.2.5 and 4.2.6. □

Since erasing rules can be eliminated from any context-free grammar (see Theorem 7.9 in [37]), we obtain the following corollary.

Corollary 4.2.8. *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 $L(H) = L(G) - \{\varepsilon\}$.* □

The family of context-free languages is also characterized by left forbidding grammars.

Theorem 4.2.9 (see Corollary 2 in [31]). $\mathbf{LFor}^{-\varepsilon} = \mathbf{LFor} = \mathbf{CF}$

We conclude this section by several remarks closely related to the previous results. Recall that we have established an equivalence between one-sided forbidding grammars and s-grammars. In [44], it is proved that special versions of s-grammars, referred to as *symmetric s-grammars*, are equivalent to forbidding grammars. Recall that in a symmetric s-grammar, each selector is of the form $X^*\overline{Y}X^*$, where X and Y are alphabets. In a forbidding grammar, the absence of symbols is checked in the entire sentential form. Based on the achieved results, we see that one-sided forbidding grammars form a counterpart to s-grammars just like forbidding grammars form a counterpart to symmetric s-grammars. As symmetric s-grammars are just special versions of s-grammars, we see that one-sided forbidding grammars are at least as powerful as forbidding grammars. This result can be also proved directly, as demonstrated next.

Theorem 4.2.10. $\mathbf{For} \subseteq \mathbf{OFor}$

Proof. Let $G = (N, T, P, S)$ be a forbidding grammar. Without any loss of generality, we assume that $(A \to w, \emptyset, W) \in P$ implies that $A \notin W$ (otherwise, such a rule would not be applicable in G). We next construct a one-sided forbidding grammar H such that $L(H) = L(G)$. Set $R = \{\langle r, 1\rangle, \langle r, 2\rangle \mid r \in P\}$ and define H as

$$H = \big(N \cup R, T, P_L, P_R, S\big)$$

where P_L and P_R are constructed in the following way. Initially, set $P_L = \emptyset$ and $P_R = \emptyset$. To complete the construction, apply the following three steps for each $r = (A \to x, \emptyset, W) \in P$

(1) add $(A \to \langle r, 1\rangle, \emptyset, W \cup R)$ to P_L;
(2) add $(\langle r, 1\rangle \to \langle r, 2\rangle, \emptyset, W \cup R)$ to P_R;
(3) add $(\langle r, 2\rangle \to x, \emptyset, R)$ to P_L.

The simulation of every $r = (A \to x, \emptyset, W) \in P$ is done in three steps. First, we check the absence of all forbidding symbols from W to the left of A by a rule from (1). Then, we check the absence of all forbidding symbols from W to the right of A by a rule from (2). By our assumption, $A \notin W$, so we do not have to check the absence of A in the current sentential form. Finally, we rewrite $\langle r, 2\rangle$ to x by a rule from (3). In all these steps, we also check the absence of all nonterminals from R. In this way, we guarantee that only a single rule is simulated at a time (if this is not the case, then the derivation is blocked). Based on these observations, we see that $L(H) = L(G)$. □

Observe that the construction in the proof of Theorem 4.2.10 does not introduce any erasing rules. Hence, whenever G is propagating, so is H. This implies the following result.

Theorem 4.2.11. $\mathbf{For}^{-\varepsilon} \subseteq \mathbf{OFor}^{-\varepsilon}$ □

In [44], the question whether $\mathbf{S}^{-\varepsilon} = \mathbf{CS}$ is explicitly formulated (see open problem (5) in [44]). As $\mathbf{S}^{-\varepsilon} = \mathbf{OFor}^{-\varepsilon}$ (see Theorem 4.2.4), we obtain a reformulation of this longstanding open question.

Corollary 4.2.12. $\mathbf{S}^{-\varepsilon} = \mathbf{CS}$ *if and only if* $\mathbf{OFor}^{-\varepsilon} = \mathbf{CS}$. □

It is worth pointing out that it is not known whether s-grammars or one-sided forbidding grammars characterize **RE** either. From Theorem 4.2.3, we obtain the following corollary.

Corollary 4.2.13. $\mathbf{S} = \mathbf{RE}$ *if and only if* $\mathbf{OFor} = \mathbf{RE}$. □

Open Problem 4.2.14. What is the generative power of one-sided forbidding grammars and s-grammars? Do they characterize **RE**? □

4.3 One-Sided Permitting Grammars

Finally, we consider one-sided permitting grammars and their generative power. We prove that $\mathbf{CF} \subset \mathbf{OPer}^{-\varepsilon} \subseteq \mathbf{SC}^{-\varepsilon}$.

Lemma 4.3.1. $\mathbf{CF} \subset \mathbf{OPer}^{-\varepsilon} \subseteq \mathbf{OPer}$

Proof. Clearly, $\mathbf{CF} \subseteq \mathbf{OPer}^{-\varepsilon} \subseteq \mathbf{OPer}$. The strictness of the first inclusion follows from Example 3.2.2. □

Lemma 4.3.2. $\mathbf{OPer}^{-\varepsilon} \subseteq \mathbf{SC}^{-\varepsilon}$

Proof. Let $G = (N, T, P_L, P_R, S)$ be a propagating one-sided permitting grammar. We next construct a propagating scattered context grammar H such that $L(H) = L(G)$. Define H as

$$H = \big(N, T, P', S\big)$$

with P' constructed as follows:

(1) for each $(A \to x, \emptyset, \emptyset) \in P_L \cup P_R$, add $(A) \to (x)$ to P';
(2) for each $(A \to x, \{X_1, X_2, \ldots, X_n\}, \emptyset) \in P_L$ and every permutation $(i_1, i_2, \ldots, i_n)$ of $(1, 2, \ldots, n)$, where $n \geq 1$, extend P' by adding

$$(X_{i_1}, X_{i_2}, \ldots, X_{i_n}, A) \to (X_{i_1}, X_{i_2}, \ldots, X_{i_n}, x)$$

(3) for each $(A \to x, \{X_1, X_2, \ldots, X_n\}, \emptyset) \in P_R$ and every permutation $(i_1, i_2, \ldots, i_n)$ of $(1, 2, \ldots, n)$, where $n \geq 1$, extend P' by adding

$$(A, X_{i_1}, X_{i_2}, \ldots, X_{i_n}) \to (x, X_{i_1}, X_{i_2}, \ldots, X_{i_n})$$

Rules with no permitting symbols are simulated by ordinary context-free-like rules, introduced in (1). The presence of permitting symbols is checked by scattered context rules, introduced in (2) and (3), which have every permitting symbol to the left and to the right of the rewritten symbol, respectively. Because the exact order of permitting symbols in a sentential form is irrelevant in one-sided permitting grammars, we introduce every permutation of the all permitting symbols. Based on these observations, we see that $L(H) = L(G)$. □

Theorem 4.3.3. $\mathbf{CF} \subset \mathbf{OPer}^{-\varepsilon} \subseteq \mathbf{SC}^{-\varepsilon} \subseteq \mathbf{CS} = \mathbf{ORC}^{-\varepsilon}$

Proof. By Lemma 4.3.1, $\mathbf{CF} \subset \mathbf{OPer}^{-\varepsilon}$. By Lemma 4.3.2, $\mathbf{OPer}^{-\varepsilon} \subseteq \mathbf{SC}^{-\varepsilon}$. The inclusion $\mathbf{SC}^{-\varepsilon} \subseteq \mathbf{CS}$ follows from Theorem 2.3.18. Finally, $\mathbf{CS} = \mathbf{ORC}^{-\varepsilon}$ follows from Theorem 4.1.3. □

Recall that the one-sided random context grammar from Example 3.2.2 is, in fact, a propagating left permitting grammar. Since every left permitting grammar is a special case of a one-sided permitting grammar, we obtain the following corollary of Theorem 4.3.3.

Corollary 4.3.4. $\mathbf{CF} \subset \mathbf{LPer}^{-\varepsilon} \subseteq \mathbf{SC}^{-\varepsilon} \subseteq \mathbf{CS} = \mathbf{ORC}^{-\varepsilon}$ □

In the conclusion of this chapter, we point out some consequences implied by the results achieved above. Then, we formulate some open problem areas.

Corollary 4.3.5. $\mathbf{RC}^{-\varepsilon} \subset \mathbf{ORC}^{-\varepsilon} \subset \mathbf{RC} = \mathbf{ORC}$

Proof. These inclusions follow from Theorems 4.1.3 and 4.1.4 in this chapter and from Theorem 2.3.14. □

Corollary 4.3.6. $\mathbf{LFor}^{-\varepsilon} = \mathbf{LFor} \subset \mathbf{For}^{-\varepsilon} \subseteq \mathbf{OFor}^{-\varepsilon} \subseteq \mathbf{OFor}$

Proof. These inclusions follow from Theorems 4.2.10 and 4.2.9 in this chapter and from Theorem 2.3.15. □

The previous results give rise to the following four open problem areas.

Open Problem 4.3.7. Establish the relations between families $\mathbf{Per}^{-\varepsilon}$, $\mathbf{LPer}^{-\varepsilon}$, and $\mathbf{OPer}^{-\varepsilon}$. What is the generative power of left random context grammars? □

Open Problem 4.3.8. Recall that $\mathbf{Per}^{-\varepsilon} = \mathbf{Per}$ (see Theorem 2.3.14). Is it true that $\mathbf{OPer}^{-\varepsilon} = \mathbf{OPer}$? □

Open Problem 4.3.9. Theorem 4.2.9 says that $\mathbf{LFor}^{-\varepsilon} = \mathbf{LFor}$. Is it also true that $\mathbf{OFor}^{-\varepsilon} = \mathbf{OFor}$? □

Open Problem 4.3.10. Does $\mathbf{OPer}^{-\varepsilon} = \mathbf{ORC}^{-\varepsilon}$ hold? If so, then Theorem 4.3.3 would imply $\mathbf{SC}^{-\varepsilon} = \mathbf{CS}$ and, thereby, solve a longstanding open question. □

Chapter 5
Normal Forms

Formal language theory has always struggled to turn grammars into *normal forms*, in which grammatical rules satisfy some prescribed properties or format because they are easier to handle from a theoretical as well as practical standpoint. Concerning context-free grammars, there exist two famous normal forms—the Chomsky and Greibach normal forms (see [69]). In the former, every grammatical rule has on its right-hand side either a terminal or two nonterminals. In the latter, every grammatical rule has on its right-hand side a terminal followed by zero or more nonterminals. Similarly, there exist normal forms for general grammars, such as the Kuroda, Penttonen, and Geffert normal forms (see [46] and Section 2.3).

The present chapter establishes four normal forms for one-sided random context grammars. The first of them has the set of left random context rules coinciding with the set of right random context rules. The second normal form, in effect, consists in demonstrating how to turn any one-sided random context grammar to an equivalent one-sided random context grammar with the sets of left and right random context rules being disjoint. The third normal form resembles the Chomsky normal form for context-free grammars, mentioned above. In the fourth normal form, each rule has its permitting or forbidding context empty.

This chapter is divided into Sections 5.1 through 5.4. Each section establishes one of the above-mentioned normal forms of one-sided random context grammars.

5.1 First Normal Form

In the first normal form, the set of left random context rules coincides with the set of right random context rules.

Theorem 5.1.1. *Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. Then, there is a one-sided random context grammar, $H = (N', T, P'_L, P'_R, S)$, such that $L(H) = L(G)$ and $P'_L = P'_R$.*

Proof. Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar, and let $S', \#, \$$ be three new symbols not in $N \cup T$. Define the one-sided random context grammar

$$H = \big(N \cup \{S', \#, \$\}, T, P', P', S'\big)$$

with P' constructed in the following way. Initially, set

$$P' = \big\{(S' \to \#S\$, \emptyset, \emptyset), (\$ \to \varepsilon, \{\#\}, N), (\# \to \varepsilon, \emptyset, \emptyset)\big\}$$

Then, complete the construction by applying the following two steps

(1) for each $(A \to w, U, W) \in P_L$, add $(A \to w, U, W \cup \{\$\})$ to P';
(2) for each $(A \to w, U, W) \in P_R$, add $(A \to w, U, W \cup \{\#\})$ to P'.

In H, the side on which the rules check the presence and absence of symbols is not explicitly prescribed by their membership to a certain set of rules. Instead, the two new symbols, \$ and #, are used to force rules to check for their permitting and forbidding symbols on a proper side. These two new symbols are introduced by $(S' \to \#S\$, \emptyset, \emptyset)$, which is used at the very beginning of every derivation. Therefore, every sentential form of H has its symbols placed between these two end markers. If we want a rule to look to the left, we guarantee the absence of \$; otherwise, we guarantee the absence of #. At the end of a derivation, these two new symbols are erased by $(\$ \to \varepsilon, \{\#\}, N)$ and $(\# \to \varepsilon, \emptyset, \emptyset)$. The former rule checks whether, disregarding #, the only present symbols in the current sentential form are terminals. Observe that if $(\# \to \varepsilon, \emptyset, \emptyset)$ is used prematurely, H cannot derive a sentence because the presence of # is needed to erase \$ by $(\$ \to \varepsilon, \{\#\}, N)$. Based on these observations, we see that $L(H) = L(G)$. Since H has effectively only a single set of rules, the theorem holds. □

Next, we show that Theorem 5.1.1 also holds if we restrict ourselves only to propagating one-sided random context grammars.

Theorem 5.1.2. *Let $G = (N, T, P_L, P_R, S)$ be a propagating one-sided random context grammar. Then, there is a propagating one-sided random context grammar, $H = (N', T, P'_L, P'_R, S)$, such that $L(H) = L(G)$ and $P'_L = P'_R$.*

Proof. We prove this theorem by analogy with the proof of Theorem 5.1.1, but we give the present proof in a greater detail. Since H has to be propagating, instead of # and \$ as end markers, we use boundary symbols appearing in sentential forms. To this end, we keep the leftmost symbol marked by ` and the rightmost symbol marked by ´. If there is only a single symbol in the current sentential form, we mark it by ˇ. At the end of every successful derivation, only terminals and two boundary marked terminals are present. To produce a string of terminals, we unmark these two marked terminals.

Let $G = (N, T, P_L, P_R, S)$ be a propagating one-sided random context grammar. Set $V = N \cup T$, $\grave{V} = \{\grave{X} \mid X \in V\}$, $\acute{V} = \{\acute{X} \mid X \in V\}$, and $\check{N} = \{\check{A} \mid A \in N\}$. Without any loss of generality, assume that V, $\grave{V}$, $\acute{V}$, and $\check{N}$ are pairwise disjoint. Construct the propagating one-sided random context grammar

$$H = \left(N', T, P', P', \check{S}\right)$$

as follows. Initially, set $N' = N \cup \grave{V} \cup \acute{V} \cup \check{N}$ and $P' = \emptyset$. To keep the rest of the construction as readable as possible, we introduce several functions. Define the function $\grave{\pi}$ from 2^N to $2^{2^{N'}}$ as $\grave{\pi}(\emptyset) = \{\emptyset\}$ and

$$\begin{aligned}\grave{\pi}(\{A_1, A_2, \ldots, A_n\}) = \; & \{\{A_1, A_2, \ldots, A_n\}\} \cup \\ & \{\{\grave{A}_1, A_2, \ldots, A_n\}\} \cup \\ & \{\{A_1, \grave{A}_2, \ldots, A_n\}\} \cup \\ & \quad \vdots \\ & \{\{A_1, A_2, \ldots, \grave{A}_n\}\}\end{aligned}$$

Define the function $\acute{\pi}$ from 2^N to $2^{2^{N'}}$ as $\acute{\pi}(\emptyset) = \{\emptyset\}$ and

$$\begin{aligned}\acute{\pi}(\{A_1, A_2, \ldots, A_n\}) = \; & \{\{A_1, A_2, \ldots, A_n\}\} \cup \\ & \{\{\acute{A}_1, A_2, \ldots, A_n\}\} \cup \\ & \{\{A_1, \acute{A}_2, \ldots, A_n\}\} \cup \\ & \quad \vdots \\ & \{\{A_1, A_2, \ldots, \acute{A}_n\}\}\end{aligned}$$

Define the function $\grave{\sigma}$ from 2^N to $2^{N'}$ as $\grave{\sigma}(W) = W \cup \{\grave{Y} \mid Y \in W\} \cup \acute{V}$. Define the function $\acute{\sigma}$ from 2^N to $2^{N'}$ as $\acute{\sigma}(W) = W \cup \{\acute{Y} \mid Y \in W\} \cup \grave{V}$.

To complete the construction, apply the following ten steps.

(1) *Simulation of unit rules when there is only a single symbol present in the current sentential form.*
For each $(A \to B, \emptyset, W) \in P_L \cup P_R$, where $B \in N$, add $(\check{A} \to \check{B}, \emptyset, \emptyset)$ to P'.

(2) *Simulation of rules generating a single terminal when there is only a single symbol present in the current sentential form.*
For each $(A \to a, \emptyset, W) \in P_L \cup P_R$, where $a \in T$, add $(\check{A} \to a, \emptyset, \emptyset)$ to P'.

(3) *Simulation of rules forking the only symbol in the current sentential form into two or more symbols.*
For each $(A \to XwY, \emptyset, W) \in P_L \cup P_R$, where $X, Y \in V$, $w \in V^*$, add $(\check{A} \to \grave{X}w\acute{Y}, \emptyset, \emptyset)$ to P'.

(4) *Simulation of rules from P_L rewriting the leftmost nonterminal.*
For each $(A \to Xw, \emptyset, W) \in P_L$, where $X \in V$, $w \in V^*$, add $(\grave{A} \to \grave{X}w, \emptyset, \emptyset)$ to P'.

(5) *Simulation of rules from P_R rewriting the rightmost nonterminal.*
For each $(A \to wX, \emptyset, W) \in P_R$, where $X \in V$, $w \in V^*$, add $(\acute{A} \to w\acute{X}, \emptyset, \emptyset)$ to P'.

(6) *Simulation of rules from P_L rewriting the rightmost nonterminal.*
For each $(A \to wX, U, W) \in P_L$, where $X \in V$, $w \in V^*$, and every $U' \in \check{\pi}(U)$, add $(\acute{A} \to w\acute{X}, U', \check{\sigma}(W))$ to P'.

(7) *Simulation of rules from P_L rewriting a non-marked nonterminal.*
For each $(A \to w, U, W) \in P_L$, where $w \in V^*$, and every $U' \in \check{\pi}(U)$, add $(A \to w, U', \check{\sigma}(W))$ to P'.

(8) *Simulation of rules from P_R rewriting the leftmost nonterminal.*
For each $(A \to Xw, U, W) \in P_R$, where $X \in V$, $w \in V^*$, and every $U' \in \acute{\pi}(U)$, add $(\grave{A} \to \check{X}w, U', \acute{\sigma}(W))$ to P'.

(9) *Simulation of rules from P_R rewriting a non-marked nonterminal.*
For each $(A \to w, U, W) \in P_R$, where $w \in V^*$, and every $U' \in \acute{\pi}(U)$, add $(A \to w, U', \acute{\sigma}(W)\})$ to P'.

(10) *Unmark both boundary terminals if only terminals and marked terminals are present.*
For each $a, c \in T$, add $(\grave{a} \to a, \emptyset, \emptyset)$ and $(\acute{c} \to c, \{\grave{a}\}, N)$ to P'.

Observe (i) through (vi), given next.

(i) Let $S = X_1 \Rightarrow_G X_2 \Rightarrow_G \cdots \Rightarrow_G X_n \Rightarrow_G a$ be a derivation, where $a \in T$, $X_i \in N$, for all i, $1 \leq i \leq n$, for some $n \geq 1$. Notice that every applied rule in such a derivation in G has to have an empty permitting context; otherwise, it would not be applicable. Then, there is

$$\check{S} = \check{X}_1 \Rightarrow_H \check{X}_2 \Rightarrow_H \cdots \Rightarrow_H \check{X}_n \Rightarrow_H a$$

by rules introduced in (1) and (2). Conversely, for every derivation in H by rules from (1) and (2), there is a corresponding derivation in G.

(ii) Rules from P_L used to rewrite the leftmost nonterminal of a sentential form and rules from P_R used to rewrite the rightmost nonterminal of a sentential form have to have empty permitting contexts; otherwise, they would not be applicable. Therefore, the assumption of empty permitting contexts in rules from P_L and P_R in (1) through (5) is without any loss of generality. Also, for the same reason, the resulting rules, introduced to P', have empty forbidding contexts.

(iii) Excluding case (i), every sentential form of H that has one or more nonterminals is bounded by marked symbols. If the leftmost marked symbol is unmarked prematurely by a rule of the form $(\grave{a} \to a, \emptyset, \emptyset)$, introduced in (10), no sentence can be obtained because the presence of a symbol marked by ` is needed to unmark the rightmost symbol marked by ´ by a rule of the form $(\acute{c} \to c, \{\grave{a}\}, N)$, introduced in (10).

(iv) The simulation of a rewrite of a nonterminal that is not the leftmost nor the rightmost symbol in the current sentential form by a rule from P_L is done by rules from (6) and (7). To force the check to the left, the absence of all symbols marked by ´ is required. Analogously, by (8) and (9), the forced check to the right is done by requiring the absence of all symbols marked by `. Because of the previous observation, this simulation is correct.

(v) Let $S \Rightarrow_G^* w$ be a derivation in G, where $w \in T^+$ such that $|w| \geq 2$. Using the corresponding rules introduced in steps (1) and (3) through (9) and then using two rules from (10), it is possible to derive w in H.

(vi) Every derivation in H leading to a sentence containing more than one terminal is of the form

$$\begin{aligned}
\check{S} &\Rightarrow_H^* \check{A} && \text{(by rules from (1))}\\
&\Rightarrow_H \grave{X}w\acute{Y} && \text{(by a rule from (3))}\\
&\Rightarrow_H^* \grave{a}_1 a_2 \cdots a_{n-1}\acute{a}_n && \text{(by rules from (4) through (9))}\\
&\Rightarrow_H \grave{a}_1 a_2 \cdots a_{n-1}a_n && \text{(by } (\acute{a}_n \rightarrow a_n, \{\grave{a}_1\}, N) \text{ from (10))}\\
&\Rightarrow_H a_1 a_2 \cdots a_{n-1}a_n && \text{(by } (\grave{a}_1 \rightarrow a_1, \emptyset, \emptyset) \text{ from (10))}
\end{aligned}$$

where $A \in N$, $X, Y \in V$, $w \in V^*$, $a_i \in T$, for all i, $1 \leq i \leq n$, for some $n \geq 2$. Such a derivation is also possible in G (of course, without marked symbols and the last two applied rules).

Based on these observations, we see that $L(H) = L(G)$. Since H has effectively only a single set of rules, the theorem holds. □

5.2 Second Normal Form

The second normal form represents a dual normal form to that in Theorems 5.1.1 and 5.1.2. Indeed, we show that every one-sided random context grammar can be turned into an equivalent one-sided random context grammar with the sets of left and right random context rules being disjoint.

Theorem 5.2.1. *Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. Then, there is a one-sided random context grammar, $H = (N', T, P'_L, P'_R, S)$, such that $L(H) = L(G)$ and $P'_L \cap P'_R = \emptyset$. Furthermore, if G is propagating, then so is H.*

Proof. Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. Construct

$$H = \left(N', T, P'_L, P'_R, S\right)$$

where

$$\begin{aligned}
N' &= N \cup \{L, R\}\\
P'_L &= \{(A \rightarrow x, U, W \cup \{L\}) \mid (A \rightarrow x, U, W) \in P_L\}\\
P'_R &= \{(A \rightarrow x, U, W \cup \{R\}) \mid (A \rightarrow x, U, W) \in P_R\}
\end{aligned}$$

Without any loss of generality, we assume that $\{L,R\} \cap (N \cup T) = \emptyset$. Observe that the new nonterminals L and R cannot appear in any sentential form. Therefore, it is easy to see that $L(H) = L(G)$. Furthermore, observe that if G is propagating, then so is H. Since $P'_L \cap P'_R = \emptyset$, the theorem holds. □

5.3 Third Normal Form

The third normal form represents an analogy of the well-known Chomsky normal form for context-free grammars. However, since one-sided random context grammars with erasing rules are more powerful than their propagating versions, we allow the presence of erasing rules in the transformed grammar.

Theorem 5.3.1. *Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. Then, there is a one-sided random context grammar, $H = (N', T, P'_L, P'_R, S)$, such that $L(H) = L(G)$ and $(A \to x, U, W) \in P'_L \cup P'_R$ implies that $x \in N'N' \cup T \cup \{\varepsilon\}$. Furthermore, if G is propagating, then so is H.*

Proof. Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. Set $V = N \cup T$ and $\bar{T} = \{\bar{a} \mid a \in T\}$. Define the homomorphism τ from V^* to $(N \cup \bar{T})^*$ as $\tau(A) = A$ for each $A \in N$, and $\tau(a) = \bar{a}$ for each $a \in T$. Let ℓ be the length of the longest right-hand side of a rule from $P_L \cup P_R$. Set

$$M = \big\{ \langle y \rangle \mid y \in V^+, 2 \le |y| \le \ell - 1 \big\}$$

Without any loss of generality, we assume that V, $\bar{T}$, and M are pairwise disjoint. Construct

$$H = \big(N', T, P'_L, P'_R, S\big)$$

as follows. Initially, set

$$\begin{aligned}
N' &= N \cup \bar{T} \cup M \\
P'_L &= \{(A \to x, U, W \cup M) \mid (A \to x, U, W) \in P_L, x \in T \cup \{\varepsilon\}\} \cup \\
&\quad \{(A \to \tau(x), U, W \cup M) \mid (A \to x, U, W) \in P_L, x \in VV\} \cup \\
&\quad \{(\bar{a} \to a, \emptyset, \emptyset) \mid a \in T\} \\
P'_R &= \{(A \to x, U, W \cup M) \mid (A \to x, U, W) \in P_R, x \in T \cup \{\varepsilon\}\} \cup \\
&\quad \{(A \to \tau(x), U, W \cup M) \mid (A \to x, U, W) \in P_R, x \in VV\}
\end{aligned}$$

Perform (1) and (2), given next.

(1) For each $(A \to X_1 X_2 \cdots X_n, U, W) \in P_L$, where $X_i \in V$ for $i = 1, 2, \ldots, n$, for some $n \ge 3$,

- add $(A \to \langle X_1 X_2 \cdots X_{n-1} \rangle \tau(X_n), U, W \cup M)$ to P'_L;

- add $(\langle X_1X_2\cdots X_{n-1}\rangle \rightarrow \langle X_1X_2\cdots X_{n-2}\rangle\tau(X_{n-1}), \emptyset, M)$ to P'_L;

  $\vdots$
- add $(\langle X_1X_2\rangle \rightarrow \tau(X_1X_2), \emptyset, M)$ to P'_L.

(2) For each $(A \rightarrow X_1X_2\cdots X_n, U, W) \in P_R$, where $X_i \in V$ for $i = 1, 2, \ldots, n$, for some $n \geq 3$,

- add $(A \rightarrow \langle X_1X_2\cdots X_{n-1}\rangle\tau(X_n), U, W \cup M)$ to P'_R;
- add $(\langle X_1X_2\cdots X_{n-1}\rangle \rightarrow \langle X_1X_2\cdots X_{n-2}\rangle\tau(X_{n-1}), \emptyset, M)$ to P'_R;

  $\vdots$
- add $(\langle X_1X_2\rangle \rightarrow \tau(X_1X_2), \emptyset, M)$ to P'_R.

To give an insight into the construction, notice that rules whose right-hand side is either a terminal or the empty string are directly added to P'_L and P'_R in the initialization part of the construction. When the right-hand side of a rule has two symbols, their homomorphic image (with respect to τ) is used, which results in the new right-hand side being formed by two nonterminals, even if the original right-hand side contained terminals. Barred nonterminals are rewritten to their corresponding terminals by rules of the form $(\bar{a} \rightarrow a, \emptyset, \emptyset)$, introduced in the initialization part of the construction. Notice that their permitting and forbidding contexts can be empty.

Rules with more than two symbols on their right-hand side are simulated in a several-step way by rules from (1) and (2). Compound nonterminals of the form $\langle X_1X_2\cdots X_n\rangle$, where each X_i is a symbol, are used to satisfy the required form of every rule in $P'_L \cup P'_R$. Each rule from (1) and (2) forbids the presence of these compound symbols to the left (or right) of the rewritten nonterminal to ensure a proper simulation.

Based on these observations, we see that $L(H) = L(G)$. Moreover, observe that if G is propagating, then so is H. Since H is of the required form, the theorem holds.

□

5.4 Fourth Normal Form

In the fourth normal form, every rule has its permitting or forbidding context empty.

Theorem 5.4.1. *Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. Then, there is a one-sided random context grammar, $H = (N', T, P'_L, P'_R, S)$, such that $L(H) = L(G)$ and $(A \rightarrow x, U, W) \in P'_L \cup P'_R$ implies that $U = \emptyset$ or $W = \emptyset$. Furthermore, if G is propagating, then so is H.*

Proof. Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. Set $V = N \cup T$ and

$$F = \{\langle r, d, i\rangle \mid r = (A \rightarrow x, U, W) \in P_d, d \in \{L, R\}, i \in \{1, 2\}\}$$

Without any loss of generality, we assume that $F \cap V = \emptyset$. Construct

$$H = (N', T, P'_L, P'_R, S)$$

as follows. Initially, set $N' = N \cup F$, $P'_L = \emptyset$, and $P'_R = \emptyset$. Perform (1) and (2), given next.

(1) For each $r = (A \rightarrow x, U, W) \in P_L$,

(1.1) add $(A \rightarrow \langle r, L, 1\rangle, \emptyset, F)$ to P'_R;
(1.2) add $(\langle r, L, 1\rangle \rightarrow \langle r, L, 2\rangle, \emptyset, W \cup F)$ to P'_L;
(1.3) add $(\langle r, L, 2\rangle \rightarrow x, U, \emptyset)$ to P'_L.

(2) For each $r = (A \rightarrow x, U, W) \in P_R$,

(2.1) add $(A \rightarrow \langle r, R, 1\rangle, \emptyset, F)$ to P'_L;
(2.2) add $(\langle r, R, 1\rangle \rightarrow \langle r, R, 2\rangle, \emptyset, W \cup F)$ to P'_R;
(2.3) add $(\langle r, R, 2\rangle \rightarrow x, U, \emptyset)$ to P'_R.

To give an insight into the construction, notice that a single rule from P_L and P_R is simulated in three steps by rules introduced in (1) and (2), respectively. As we cannot check both the presence and absence of symbols in a single step, we split this check into two consecutive steps. Clearly, $L(G) \subseteq L(H)$, so we only prove that $L(H) \subseteq L(G)$.

Observe that if we apply the three rules from (1) in H, then we can apply the original rule in G. A similar application can be reformulated in terms of (2). Therefore, it remains to be shown that H cannot generate improper sentences by invalid intermixed simulations of more than one rule of G at a time. In what follows, we consider only simulations of rules from P_L; rules from P_R are simulated analogously.

Let us consider a simulation of some $r = (A \rightarrow x, U, W) \in P_L$. Observe that the only situation where an improper simulation may occur is that after a rule from (1.2) is applied, another simulation takes places which transforms a nonterminal to the left of $\langle r, L, 2\rangle$ that is not in U into a nonterminal that is in U. To investigate this possibility, set $V' = N' \cup T$ and consider any successful derivation in H, $S \Rightarrow_H^* z$, where $z \in L(H)$. This derivation can be written in the form

$$S \Rightarrow_H^* w \Rightarrow_H y \Rightarrow_H^* z$$

where $w = w_1\langle r, L, 1\rangle w_2$, $y = w_1\langle r, L, 2\rangle w_2$, and $w_1, w_2 \in V'^*$. Since $w \Rightarrow_H y$ by $(\langle r, L, 1\rangle \rightarrow \langle r, L, 2\rangle, \emptyset, W \cup F)$, introduced to P'_L in (1.2) from r,

$$\text{alph}(w_1) \cap (W \cup F) = \emptyset$$

From the presence of $\langle r, L, 2\rangle$, no rule from (1) is now applicable to w_1. Let $w_1 = w_1' B w_1''$ and $(B \to \langle s, R, 1\rangle, \emptyset, F) \in P_L'$, introduced in (2.1) from some $s = (B \to v, X, Y) \in P_R$ such that $B \notin U$ and

$$\mathrm{alph}(v) \cap \big(U - \mathrm{alph}(w_1)\big) \neq \emptyset$$

This last requirement implies that by successfully simulating s prior to r, we necessarily end up with an invalid simulation of r. Then,

$$w_1' B w_1'' \langle r, L, 2\rangle w_2 \Rightarrow_H w_1' \langle s, R, 1\rangle w_1'' \langle r, L, 2\rangle w_2$$

Since $\langle s, R, 1\rangle$ cannot be rewritten to $\langle s, R, 2\rangle$ by a rule from (2.2) because $\langle r, L, 2\rangle$ occurs to the right of $\langle s, R, 1\rangle$, we can either

(a) correctly finish the simulation of r by rewriting $\langle r, L, 2\rangle$ to x (recall that $B \notin U$) or
(b) rewrite some nonterminal in w_1' or w_1''.

However, observe that in (b), we end up in the same situation as we are now.

Based on these observations, we see that no invalid intermixed simulations of more than one rule of G at a time are possible in H. Hence, $L(H) \subseteq L(G)$, so $L(H) = L(G)$. Clearly, $(A \to x, U, W) \in P_L' \cup P_R'$ implies that $U = \emptyset$ or $W = \emptyset$. Furthermore, observe that if G is propagating, then so is H. Thus, the theorem holds. □

We conclude this section by suggesting an open problem.

Open Problem 5.4.2. Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar, and consider the following four normal forms

(I) either $P_L = \emptyset$ or $P_R = \emptyset$;
(II) $(A \to x, U, W) \in P_L \cup P_R$ implies that $\mathrm{card}(U) + \mathrm{card}(W) \leq 1$;
(III) $P_L = \emptyset$ and $(A \to x, U, W) \in P_R$ implies that $W = \emptyset$;
(IV) $P_R = \emptyset$ and $(A \to x, U, W) \in P_L$ implies that $W = \emptyset$.

Can we turn G into an equivalent one-sided random context grammar in any of the above-mentioned forms? □

Chapter 6
Reduction

Recall that one-sided random context grammars characterize the family of recursively enumerable languages (see Theorem 4.1.4). 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 chapter, consisting of three sections, gives an affirmative answer to this question.

More specifically, in Section 6.1, we prove that every recursively enumerable language can be generated by a one-sided random context grammar with no more than ten nonterminals. In addition, we show that an analogous result holds for thirteen nonterminals in terms of these grammars with the set of left random context rules coinciding with the set of right random context rules.

Then, in Section 6.2, we approach the discussion concerning the reduction of these grammars with respect to the number of nonterminals in a finer way. Indeed, we introduce the notion of a *right random context nonterminal*, defined as a nonterminal that appears on the left-hand side of a right random context rule, and demonstrate how to convert any one-sided random context grammar G to an equivalent one-sided random context grammar H with two right random context nonterminals. We also explain how to achieve an analogous conversion in terms of propagating versions of these grammars (recall that they characterize the family of context-sensitive languages, see Theorem 4.1.3). Similarly, we introduce the notion of a *left random context nonterminal* and show how to convert any one-sided random context grammar G to an equivalent one-sided random context grammar H with two left random context nonterminals. We explain how to achieve an analogous conversion in terms of propagating versions of these grammars, too.

Apart from reducing the number of nonterminals, we reduce the number of rules. More specifically, in Section 6.3, we prove that any recursively enumerable language can be generated by a one-sided random context grammar having no more than two right random context rules. As a motivation behind limiting the number of right random context rules in these grammars, consider left random context grammars, which are one-sided random context grammars with no right random context rules (see Section 3). Recall that it is an open question whether these grammars are equally powerful to one-sided random context grammars (see Open Problem 4.3.7). To give an affirmative answer to this question, it is sufficient to show that in one-sided random context grammars, no right random context rules are needed. From this viewpoint, the above-mentioned result may fulfill a useful role during the solution of this problem in the future.

The results sketched above can be also seen as a contribution to the investigation concerning the *descriptional complexity* of formal models, which represents an important trend in today's formal language theory as demonstrated by several recent studies (see [11, 13, 23–26, 36, 53, 55, 56, 59, 61, 62, 76, 90, 104]). As an important part, this trend discusses the *nonterminal complexity* of grammars—an investigation area that is primarily interested in reducing the number of nonterminals in grammars without affecting their power. So, the results mentioned above actually represent new knowledge concerning the descriptional complexity of one-sided random context grammars.

6.1 Total Number of Nonterminals

In this section, we prove that every recursively enumerable language can be generated by a one-sided random context grammar H that satisfies one of conditions (I) and (II), given next.

(I) H has ten nonterminals (Theorem 6.1.1).
(II) The set of left random context rules of H coincides with the set of right random context rules, and H has thirteen nonterminals (Corollary 6.1.2).

Theorem 6.1.1. *Let K be a recursively enumerable language. Then, there is a one-sided random context grammar, $H = (N, T, P_L, P_R, S)$, such that $L(H) = K$ and* $\mathrm{card}(N) = 10$.

Proof. Let K be a recursively enumerable language. Then, by Theorem 2.3.11, there is a phrase-structure grammar in the Geffert normal form

$$G = \big(\{S,A,B,C\},T,P \cup \{ABC \to \varepsilon\},S\big)$$

satisfying $L(G) = K$. We next construct a one-sided random context grammar H such that $L(H) = L(G)$. Set $N = \{S, A, B, C\}$, $V = N \cup T$, and $N' = \{S, A, B, C, \bar{A}, \hat{A}, \bar{B}, \hat{B}, \tilde{C}, \#\}$. Without any loss of generality, we assume that $(\{\bar{A}, \hat{A}, \bar{B}, \hat{B}, \tilde{C}, \#\}) \cap V = \emptyset$. Construct

$$H = \big(N', T, P_L, P_R, S\big)$$

in the following way. Initially, set $P_L = \emptyset$ and $P_R = \emptyset$. Perform the following seven steps

(1) for each $S \rightarrow uSa \in P$, where $u \in \{A, AB\}^*$ and $a \in T$, add $(S \rightarrow uS\#a, \emptyset, \{\bar{A}, \bar{B}, \hat{A}, \hat{B}, \tilde{C}, \#\})$ to P_L;
(2) for each $S \rightarrow uSv \in P$, where $u \in \{A, AB\}^*$ and $v \in \{BC, C\}^*$, add $(S \rightarrow uSv, \emptyset, \{\bar{A}, \bar{B}, \hat{A}, \hat{B}, \tilde{C}, \#\})$ to P_L;
(3) for each $S \rightarrow uv \in P$, where $u \in \{A, AB\}^*$ and $v \in \{BC, C\}^*$, add $(S \rightarrow uv, \emptyset, \{\bar{A}, \bar{B}, \hat{A}, \hat{B}, \tilde{C}, \#\})$ to P_L;
(4) add $(A \rightarrow \bar{A}, \emptyset, N \cup \{\hat{A}, \hat{B}, \tilde{C}, \#\})$ to P_L;
add $(B \rightarrow \bar{B}, \emptyset, N \cup \{\hat{A}, \hat{B}, \tilde{C}, \#\})$ to P_L;
add $(A \rightarrow \hat{A}, \emptyset, N \cup \{\hat{A}, \hat{B}, \tilde{C}, \#\})$ to P_L;
add $(B \rightarrow \hat{B}, \{\hat{A}\}, N \cup \{\hat{B}, \tilde{C}, \#\})$ to P_L;
add $(C \rightarrow \tilde{C}, \{\hat{A}, \hat{B}\}, N \cup \{\tilde{C}, \#\})$ to P_L;
(5) add $(\hat{B} \rightarrow \varepsilon, \{\tilde{C}\}, \{S, \bar{A}, \bar{B}, \hat{A}, \hat{B}\})$ to P_R;
add $(\hat{A} \rightarrow \varepsilon, \{\tilde{C}\}, \{S, \bar{A}, \bar{B}, \hat{A}, \hat{B}\})$ to P_R;
add $(\tilde{C} \rightarrow \varepsilon, \emptyset, N \cup \{\hat{A}, \hat{B}, \tilde{C}, \#\})$ to P_L;
(6) add $(\bar{A} \rightarrow A, \emptyset, \{S, \bar{A}, \bar{B}, \hat{A}, \hat{B}, \tilde{C}\})$ to P_R;
add $(\bar{B} \rightarrow B, \emptyset, \{S, \bar{A}, \bar{B}, \hat{A}, \hat{B}, \tilde{C}\})$ to P_R;
(7) add $(\# \rightarrow \varepsilon, \emptyset, N')$ to P_L.

Before proving that $L(H) = L(G)$, let us informally describe the purpose of rules introduced in (1) through (7). H simulates the derivations of G that satisfy the form described in Theorem 2.3.12. The context-free rules in P are simulated by rules from (1) through (3). The context-sensitive rule $ABC \rightarrow \varepsilon$ is simulated in a several-step way. First, rules introduced in (4) are used to prepare the erasure of ABC. These rules rewrite nonterminals from the left to the right. In this way, it is guaranteed that whenever $\hat{A}$, $\hat{B}$, and $\tilde{C}$ appear in a sentential form, then they form a substring of the form $\hat{A}\hat{B}\tilde{C}$. Then, rules from (5) sequentially erase $\hat{B}$, $\hat{A}$, and $\tilde{C}$. Finally, rules from (6) convert barred nonterminals back to their non-barred versions to prepare another simulation of $ABC \rightarrow \varepsilon$; this conversion is done from the right to the left. For example, $AABCBCab \Rightarrow_G ABCab$ is simulated by H as follows:

$$\begin{aligned} AABCBC\#a\#b &\Rightarrow_H \bar{A}ABCBC\#a\#b \\ &\Rightarrow_H \bar{A}\hat{A}BCBC\#a\#b \\ &\Rightarrow_H \bar{A}\hat{A}\hat{B}CBC\#a\#b \\ &\Rightarrow_H \bar{A}\hat{A}\hat{B}\tilde{C}BC\#a\#b \\ &\Rightarrow_H \bar{A}\hat{A}\tilde{C}BC\#a\#b \\ &\Rightarrow_H \bar{A}\tilde{C}BC\#a\#b \\ &\Rightarrow_H \bar{A}BC\#a\#b \\ &\Rightarrow_H ABC\#a\#b \end{aligned}$$

Symbol # is used to ensure that every sentential form of H is of the form w_1w_2, where $w_1 \in (N' - \{\#\})^*$ and $w_2 \in (T \cup \{\#\})^*$. Since permitting and forbidding contexts cannot contain terminals, a mixture of symbols from T and N in H could produce a terminal string out of $L(G)$. For example, observe that $AaBC \Rightarrow_H^* a$ by rules from (4) and (5), but such a derivation does not exist in G. #s can be eliminated by an application of rules from (7), provided that no nonterminals occur to the left of # in the current sentential form. Consequently, all #s are erased at the end of every successful derivation.

To establish the identity $L(H) = L(G)$, we prove two claims. Claim 1 shows how derivations of G are simulated by H. This claim is then used to prove that $L(G) \subseteq L(H)$. Set $V' = N' \cup T$. Define the homomorphism φ from V^* to V'^* as $\varphi(X) = X$, for all $X \in N$, and $\varphi(a) = \#a$, for all $a \in T$.

Claim 1. If $S \Rightarrow_G^n x \Rightarrow_G^* z$, where $x \in V^*$ and $z \in T^*$, for some $n \geq 0$, then $S \Rightarrow_H^* \varphi(x)$.

Proof. This claim is established by induction on $n \geq 0$.

Basis. For $n = 0$, this claim is clear.

Induction Hypothesis. Suppose that there exists $n \geq 0$ such that the claim holds for all derivations of length ℓ, where $0 \leq \ell \leq n$.

Induction Step. Consider any derivation of the form

$$S \Rightarrow_G^{n+1} w \Rightarrow_G^* z$$

where $w \in V^*$ and $z \in T^*$. Since $n+1 \geq 1$, this derivation can be expressed as

$$S \Rightarrow_G^n x \Rightarrow_G w \Rightarrow_G^* z$$

for some $x \in V^+$. Without any loss of generality, we assume that x is of the form $x = x_1x_2x_3x_4$, where $x_1 \in \{A, AB\}^*$, $x_2 \in \{S, \varepsilon\}$, $x_3 \in \{BC, C\}^*$, and $x_4 \in T^*$ (see Theorem 2.3.12 and [29]).

Next, we consider all possible forms of $x \Rightarrow_G w$, covered by the following four cases—(i) through (iv).

(i) *Application of $S \to uSa \in P$.* Let $x = x_1Sx_3x_4$, $w = x_1uSax_3x_4$, and $S \to uSa \in P$, where x_1, $u \in \{A, AB\}^*$, x_3, $v \in \{BC, C\}^*$, $x_4 \in T^*$, and $a \in T$. Then, by the induction hypothesis,

$$S \Rightarrow_H^* \varphi(x_1Sx_3x_4)$$

By (1), $(S \to uS\#a, \emptyset, \{\bar{A}, \bar{B}, \hat{A}, \hat{B}, \tilde{C}, \#\}) \in P_L$. Since $\varphi(x_1Sx_3x_4) = x_1S\varphi(x_3x_4)$ and $\mathrm{alph}(x_1) \cap \{\bar{A}, \bar{B}, \hat{A}, \hat{B}, \tilde{C}, \#\} = \emptyset$,

$$x_1S\varphi(x_3x_4) \Rightarrow_H x_1uS\#a\varphi(x_3x_4)$$

As $\varphi(x_1uSax_3x_4) = x_1uS\#a\varphi(x_3x_4)$, the induction step is completed for (i).

(ii) *Application of* $S \to uSv \in P$. Let $x = x_1Sx_3x_4$, $w = x_1uSvx_3x_4$, and $S \to uSv \in P$, where $x_1, u \in \{A, AB\}^*$, $x_3, v \in \{BC, C\}^*$, and $x_4 \in T^*$. To complete the induction step for (ii), proceed by analogy with (i), but use a rule from (2) instead of a rule from (1).

(iii) *Application of* $S \to uv \in P$. Let $x = x_1Sx_3x_4$, $w = x_1uvx_3x_4$, and $S \to uv \in P$, where $x_1, u \in \{A, AB\}^*$, $x_3, v \in \{BC, C\}^*$, and $x_4 \in T^*$. To complete the induction step for (iii), proceed by analogy with (i), but use a rule from (3) instead of a rule from (1).

(iv) *Application of* $ABC \to \varepsilon$. Let $x = x_1ABCx_3x_4$, $w = x_1x_3x_4$, where $x_1 \in \{A, AB\}^*$, $x_3 \in \{BC, C\}^*$, and $x_4 \in T^*$, so $x \Rightarrow_G w$ by $ABC \to \varepsilon$. Then, by the induction hypothesis,

$$S \Rightarrow_H^* \varphi(x_1ABCx_3x_4)$$

Let $x_1 = X_1X_2 \cdots X_k$, where $k = |x_1|$ (the case when $k = 0$ means that $x_1 = \varepsilon$). Since $\varphi(x_1ABCx_3x_4) = x_1ABC\varphi(x_3x_4)$ and $\mathrm{alph}(x_1) \subseteq N$, by rules introduced in (4),

$$\begin{aligned} X_1X_2\cdots X_kABC\varphi(x_3x_4) &\Rightarrow_H \bar{X}_1X_2\cdots X_kABC\varphi(x_3x_4) \\ &\Rightarrow_H \bar{X}_1\bar{X}_2\cdots X_kABC\varphi(x_3x_4) \\ &\ \vdots \\ &\Rightarrow_H \bar{X}_1\bar{X}_2\cdots \bar{X}_kABC\varphi(x_3x_4) \\ &\Rightarrow_H \bar{X}_1\bar{X}_2\cdots \bar{X}_k\hat{A}BC\varphi(x_3x_4) \\ &\Rightarrow_H \bar{X}_1\bar{X}_2\cdots \bar{X}_k\hat{A}\hat{B}C\varphi(x_3x_4) \\ &\Rightarrow_H \bar{X}_1\bar{X}_2\cdots \bar{X}_k\hat{A}\hat{B}\tilde{C}\varphi(x_3x_4) \end{aligned}$$

Let $\bar{x}_1 = \bar{X}_1\bar{X}_2\cdots\bar{X}_k$. Since $\mathrm{alph}(\varphi(x_3x_4)) \cap \{S, \bar{A}, \bar{B}, \hat{A}, \hat{B}\} = \emptyset$, by rules introduced in (5),

$$\begin{aligned} \bar{x}_1\hat{A}\hat{B}\tilde{C}\varphi(x_3x_4) &\Rightarrow_H \bar{x}_1\hat{A}\tilde{C}\varphi(x_3x_4) \\ &\Rightarrow_H \bar{x}_1\tilde{C}\varphi(x_3x_4) \\ &\Rightarrow_H \bar{x}_1\varphi(x_3x_4) \end{aligned}$$

Finally, by rules from (6),

$$\begin{aligned} \bar{x}_1\varphi(x_3x_4) &\Rightarrow_H \bar{X}_1\cdots\bar{X}_{k-1}X_k\varphi(x_3x_4) \\ &\Rightarrow_H \bar{X}_1\cdots X_{k-1}X_k\varphi(x_3x_4) \\ &\ \vdots \\ &\Rightarrow_H X_1\cdots X_{k-1}X_k\varphi(x_3x_4) \end{aligned}$$

Recall that $x_1 = X_1\cdots X_{k-1}X_k$. Since $\varphi(x_1x_3x_4) = x_1\varphi(x_3x_4)$, the induction step is completed for (iv).

Observe that cases (i) through (iv) cover all possible forms of $x \Rightarrow_G w$. Thus, the claim holds. □

Claim 2 demonstrates how G simulates derivations of H. It is then used to prove that $L(H) \subseteq L(G)$. Define the homomorphism π from V'^* to V^* as $\pi(X) = X$, for all $X \in N$, $\pi(\bar{A}) = \pi(\hat{A}) = A$, $\pi(\bar{B}) = \pi(\hat{B}) = B$, $\pi(\tilde{C}) = C$, $\pi(a) = a$, for all $a \in T$, and $\pi(\#) = \varepsilon$. Define the homomorphism τ from V'^* to V^* as $\tau(X) = \pi(X)$, for all $X \in V' - \{\hat{A}, \hat{B}, \tilde{C}\}$, and $\tau(\hat{A}) = \tau(\hat{B}) = \tau(\tilde{C}) = \varepsilon$.

Claim 2. Let $S \Rightarrow_H^n x \Rightarrow_H^* z$, where $x \in V'^*$ and $z \in T^*$, for some $n \geq 0$. Then, $x = x_1x_2x_3x_4x_5$, where $x_1 \in \{\bar{A}, \bar{B}\}^*$, $x_2 \in \{A, B\}^*$, $x_3 \in \{S, \hat{A}BC, \hat{A}\hat{B}C, \hat{A}\hat{B}\tilde{C}, \hat{A}\tilde{C}, \tilde{C}, \varepsilon\}$, $x_4 \in \{B, C\}^*$, and $x_5 \in (T \cup \{\#\})^*$. Furthermore,

(a) if $x_3 \in \{S, \varepsilon\}$, then $S \Rightarrow_G^* \pi(x)$;
(b) if $x_3 \in \{\hat{A}BC, \hat{A}\hat{B}C, \hat{A}\hat{B}\tilde{C}\}$, then $x_2 = \varepsilon$ and $S \Rightarrow_G^* \pi(x)$;
(c) if $x_3 \in \{\hat{A}\tilde{C}, \tilde{C}\}$, then $x_2 = \varepsilon$ and $S \Rightarrow_G^* \tau(x)$.

Proof. This claim is established by induction on $n \geq 0$.

Basis. For $n = 0$, this claim is clear.

Induction Hypothesis. Suppose that there exists $n \geq 0$ such that the claim holds for all derivations of length ℓ, where $0 \leq \ell \leq n$.

Induction Step. Consider any derivation of the form

$$S \Rightarrow_H^{n+1} w \Rightarrow_H^* z$$

where $w \in V'^*$ and $z \in T^*$. Since $n+1 \geq 1$, this derivation can be expressed as

$$S \Rightarrow_H^n x \Rightarrow_H w \Rightarrow_H^* z$$

for some $x \in V'^+$. By the induction hypothesis, $x = x_1x_2x_3x_4x_5$, where $x_1 \in \{\bar{A}, \bar{B}\}^*$, $x_2 \in \{A, B\}^*$, $x_3 \in \{S, \hat{A}BC, \hat{A}\hat{B}C, \hat{A}\hat{B}\tilde{C}, \hat{A}\tilde{C}, \tilde{C}, \varepsilon\}$, $x_4 \in \{B, C\}^*$, and $x_5 \in (T \cup \{\#\})^*$. Furthermore, (a) through (c), stated in the claim, hold.

Next, we consider all possible forms of $x \Rightarrow_H w$, covered by the following five cases—(i) through (v).

(i) *Application of a rule from (1).* Let $x_3 = S$, $x_1 = x_4 = \varepsilon$, and $(S \rightarrow uS\#a, \emptyset, \{\bar{A}, \bar{B}, \hat{A}, \hat{B}, \tilde{C}, \#\}) \in P_L$, introduced in (1), where $u \in \{A, AB\}^*$ and $a \in T$, so

$$x_2Sx_5 \Rightarrow_H x_2uS\#ax_5$$

Observe that if $x_4 \neq \varepsilon$, then $w \Rightarrow_H^* z$ does not hold. Indeed, if $x_4 \neq \varepsilon$, then to erase the nonterminals in x_4, there have to be As in x_2. However, the # symbol, introduced between x_2 and x_4, blocks the applicability of $(C \rightarrow \tilde{C}, \{\hat{A}, \hat{B}\}, N \cup \{\tilde{C}, \#\}) \in P_L$, introduced in (4), which is needed to erase the nonterminals in x_4. Since $(\# \rightarrow \varepsilon, \emptyset, N') \in P_L$, introduced in (7), requires that there are no nonterminals to the left of #, the derivation cannot be successfully finished. Hence, $x_4 = \varepsilon$. Since

$u \in \{A, B\}^*$ and $\#a \in (T \cup \{\#\})^*$, $x_2uS\#ax_5$ is of the required form. As $x_3 = S$, $S \Rightarrow_G^* \pi(x)$. Observe that $\pi(x) = \pi(x_2)S\pi(x_5)$. By (1), $S \rightarrow uSa \in P$, so

$$\pi(x_2)S\pi(x_5) \Rightarrow_G \pi(x_2)uSa\pi(x_5)$$

Since $\pi(x_2)uSa\pi(x_5) = \pi(x_2uS\#ax_5)$ and both $\hat{A}\tilde{C}$ and $\tilde{C}$ are not substrings of $x_2uS\#ax_5$, the induction step is completed for (i).

(ii) *Application of a rule from (2).* Let $x_3 = S$, $x_1 = \varepsilon$, and $(S \rightarrow uSv, \emptyset, \{\bar{A}, \bar{B}, \hat{A}, \hat{B}, \tilde{C}, \#\}) \in P_L$, introduced in (2), where $u \in \{A, AB\}^*$ and $v \in \{BC, C\}^*$, so

$$x_2Sx_4x_5 \Rightarrow_H x_2uSvx_4x_5$$

To complete the induction step for (ii), proceed by analogy with (i), but use $S \rightarrow uSv \in P$ instead of $S \rightarrow uSa \in P$. Observe that x_4 may be nonempty in this case.

(iii) *Application of a rule from (3).* Let $x_3 = S$, $x_1 = \varepsilon$, and $(S \rightarrow uv, \emptyset, \{\bar{A}, \bar{B}, \hat{A}, \hat{B}, \tilde{C}, \#\}) \in P_L$, introduced in (3), where $u \in \{A, AB\}^*$ and $v \in \{BC, C\}^*$, so

$$x_2Sx_4x_5 \Rightarrow_H x_2uvx_4x_5$$

To complete the induction step for (iii), proceed by analogy with (i), but use $S \rightarrow uv \in P$ instead of $S \rightarrow uSa \in P$. Observe that x_4 may be nonempty in this case.

(iv) *Application of a rule from (5).* Let $x_3 \in \{\hat{A}\hat{B}\tilde{C}, \hat{A}\tilde{C}, \tilde{C}\}$. By the induction hypothesis (more specifically, by (b) and (c)), $x_2 = \varepsilon$. Then, there are three subcases, depending on what x_3 actually is.

(iv.i) Let $x_3 = \hat{A}\hat{B}\tilde{C}$. Then, $x_1\hat{A}\hat{B}\tilde{C}x_4x_5 \Rightarrow_H x_1\hat{A}\tilde{C}x_4x_5$ by $(\hat{B} \rightarrow \varepsilon, \{\tilde{C}\}, \{S, \bar{A}, \bar{B}, \hat{A}, \hat{B}\}) \in P_R$, introduced in (5). Observe that this is the only applicable rule from (5). By the induction hypothesis, $S \Rightarrow_G^* \pi(x)$. Since $\pi(x) = \pi(x_1)ABC\pi(x_4x_5)$,

$$\pi(x_1)ABC\pi(x_4x_5) \Rightarrow_G \pi(x_1)\pi(x_4x_5)$$

by $ABC \rightarrow \varepsilon$. As $w = x_1\hat{A}\tilde{C}x_4x_5$ is of the required form and $\pi(x_1)\pi(x_4x_5) = \tau(w)$, the induction step is completed for (iv.i).

(iv.ii) Let $x_3 = \hat{A}\tilde{C}$. Then, $x_1\hat{A}\tilde{C}x_4x_5 \Rightarrow_H x_1\tilde{C}x_4x_5$ by $(\hat{A} \rightarrow \varepsilon, \{\tilde{C}\}, \{S, \bar{A}, \bar{B}, \hat{A}, \hat{B}\}) \in P_R$, introduced in (5). Observe that this is the only applicable rule from (5). By the induction hypothesis, $S \Rightarrow_G^* \tau(x)$. As $w = x_1\tilde{C}x_4x_5$ is of the required form and $\tau(x) = \tau(w)$, the induction step is completed for (iv.ii).

(iv.iii) Let $x_3 = \tilde{C}$. Then, $x_1\tilde{C}x_4x_5 \Rightarrow_H x_1x_4x_5$ by $(\tilde{C} \rightarrow \varepsilon, \emptyset, N \cup \{\hat{A}, \hat{B}, \tilde{C}, \#\}) \in P_L$, introduced in (5). Observe that this is the only applicable rule from (5). By the induction hypothesis, $S \Rightarrow_G^* \tau(x)$. As $w = x_1x_4x_5$ is of the required form and $\tau(x) = \tau(w)$, the induction step is completed for (iv.iii).

(v) *Application of a rule from (4), (6), or (7).* Let $x \Rightarrow_H w$ by a rule from (4), (6), or (7). Observe that $x_3 \notin \{\hat{A}\tilde{C}, \tilde{C}\}$ has to hold; otherwise, none of these rules is applicable. Indeed, if $x_3 \in \{\hat{A}\tilde{C}, \tilde{C}\}$, then $x_2 = \varepsilon$ by the induction hypothesis (more specifically, by (c)), which implies that no rule from (4) is applicable. Also, $x_3 \in \{\hat{A}\tilde{C}, \tilde{C}\}$ would imply that no rule from (6) and (7) is applicable. Therefore, $S \Rightarrow_G^* \pi(w)$ follows directly from the induction hypothesis (obviously, $\pi(w) = \pi(x)$, and since $x_3 \notin \{\hat{A}\tilde{C}, \tilde{C}\}$, $S \Rightarrow_G^* \pi(x)$ by the induction hypothesis). As w is clearly of the required form, the induction step is completed for (v).

Observe that cases (i) through (v) cover all possible forms of $x \Rightarrow_H w$. Thus, the claim holds. □

We next prove that $L(H) = L(G)$. Consider Claim 1 for $x \in T^*$. Then, $S \Rightarrow_H^* \varphi(x)$. Let $x = a_1 a_2 \cdots a_k$, where $k = |x|$ (the case when $k = 0$ means that $x = \varepsilon$), so $\varphi(x) = \#a_1\#a_2 \cdots \#a_k$. By (7), $(\# \rightarrow \varepsilon, \emptyset, N') \in P_L$. Therefore,

$$\begin{aligned}
\#a_1\#a_2 \cdots \#a_k &\Rightarrow_H a_1\#a_2 \cdots \#a_k \\
&\Rightarrow_H a_1 a_2 \cdots \#a_k \\
&\ \ \vdots \\
&\Rightarrow_H a_1 a_2 \cdots a_k
\end{aligned}$$

Hence, $x \in L(G)$ implies that $x \in L(H)$, so $L(G) \subseteq L(H)$.

Consider Claim 2 for $x \in T^*$. Then, $S \Rightarrow_G^* \pi(x)$. Since $x \in T^*$, $\pi(x) = x$. Hence, $x \in L(H)$ implies that $x \in L(G)$, so $L(H) \subseteq L(G)$.

The two inclusions, $L(G) \subseteq L(H)$ and $L(H) \subseteq L(G)$, imply that $L(H) = L(G)$. As $\mathrm{card}(N') = 10$, the theorem holds. □

Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. Recall that in the proof of Theorem 5.1.1, a construction of a one-sided random context grammar, $H = (N', T, P'_L, P'_R, S')$, satisfying $L(H) = L(G)$ and $P'_L = P'_R$, is given. Observe that this construction introduces three new nonterminals—that is, $\mathrm{card}(N') = \mathrm{card}(N) + 3$. Therefore, we obtain the following corollary.

Corollary 6.1.2. *Let K be a recursively enumerable language. Then, there is a one-sided random context grammar, $H = (N, T, P_L, P_R, S)$, such that $L(H) = K$, $P_L = P_R$, and* $\mathrm{card}(N) = 13$. □

6.2 Number of Left and Right Random Context Nonterminals

In this section, we approach the discussion concerning the reduction of one-sided random context grammars with respect to the number of nonterminals in a finer way. Indeed, we introduce the notion of a *right random context nonterminal*, defined as a nonterminal that appears on the left-hand side of a right random context rule, and demonstrate how to convert any one-sided random context grammar G to an equivalent one-sided random context grammar H with two right random context nonterminals. We also explain how to achieve an analogous conversion in terms of propagating versions of these grammars (recall that they characterize the family of context-sensitive languages, see Theorem 4.1.3). Similarly, we introduce the notion of a *left random context nonterminal* and show how to convert any one-sided random context grammar G to an equivalent one-sided random context grammar H with two left random context nonterminals. We explain how to achieve an analogous conversion in terms of propagating versions of these grammars, too.

First, we define these two new measures formally.

Definition 6.2.1. Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. If $(A \rightarrow x, U, W) \in P_R$, then A is a *right random context nonterminal*. The *number of right random context nonterminals* of G is denoted by $\mathrm{nrrcn}(G)$ and defined as

$$\mathrm{nrrcn}(G) = \mathrm{card}\big(\{A \mid (A \rightarrow x, U, W) \in P_R\}\big) \qquad \square$$

Left random context nonterminals and their number in a one-sided random context grammar are defined analogously.

Definition 6.2.2. Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. If $(A \rightarrow x, U, W) \in P_L$, then A is a *left random context nonterminal*. The *number of left random context nonterminals* of G is denoted by $\mathrm{nlrcn}(G)$ and defined as

$$\mathrm{nlrcn}(G) = \mathrm{card}\big(\{A \mid (A \rightarrow x, U, W) \in P_L\}\big) \qquad \square$$

Next, we prove that every recursively enumerable language can be generated by a one-sided random context grammar H that satisfies one of conditions (I) through (III), given next.

(I) H has four right random context nonterminals and six left random context nonterminals (Corollary 6.2.3).
(II) H has two right random context nonterminals (Theorem 6.2.5).
(III) H has two left random context nonterminals (Theorem 6.2.6).

In addition, we demonstrate that every context-sensitive language can be generated by a propagating one-sided random context grammar H with either two right random context nonterminals (Theorem 6.2.8), or with two left random context nonterminals (Theorem 6.2.9).

Observe that the construction in the proof of Theorem 6.1.1 implies the following result concerning the number of left and right random context nonterminals.

Corollary 6.2.3. *Let K be a recursively enumerable language. Then, there is a one-sided random context grammar, $H = (N, T, P_L, P_R, S)$, such that $L(H) = K$,* $\mathrm{nrrcn}(H) = 4$, *and* $\mathrm{nlrcn}(H) = 6$. □

Considering only the number of right random context nonterminals, we can improve the previous corollary as described in the following lemma.

Lemma 6.2.4. *Let G be a one-sided random context grammar. Then, there is a one-sided random context grammar H such that $L(H) = L(G)$ and* $\mathrm{nrrcn}(H) = 2$.

Proof. Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. We next construct a one-sided random context grammar H such that $L(H) = L(G)$ and $\mathrm{nrrcn}(H) = 2$. Set $V = N \cup T$,

$$R = \{\langle r, i\rangle \mid r \in P_R, i = 1, 2\}$$

and

$$_{\$}R = \{\langle \$, r, i\rangle \mid r \in P_R, i = 1, 2\}$$

Without any loss of generality, we assume that R, $_{\$}R$, $\{S', \#_1, \#_2, \$\}$, and V are pairwise disjoint. Construct

$$H = (N', T, P'_L, P'_R, S')$$

as follows. Initially, set $N' = N \cup R \cup {}_{\$}R \cup \{\#_1, \#_2, \$, S'\}$, $P'_L = \emptyset$, and $P'_R = \emptyset$. Furthermore, set $\bar{N} = N' - N$. Perform (1) through (3), given next.

(1) Add $(S' \to S\$, \emptyset, \emptyset)$ and $(\$ \to \varepsilon, \emptyset, N')$ to P'_L.
(2) For each $(A \to y, U, W) \in P_L$, add $(A \to y, U, W \cup \bar{N})$ to P'_L.
(3) For each $r = (A \to y, U, W) \in P_R$,

(3.1) add $(A \to \langle r, 1\rangle, \emptyset, \bar{N})$ to P'_L;
(3.2) add $(\$ \to \langle \$, r, 1\rangle, \{\langle r, 1\rangle\}, \bar{N} - \{\langle r, 1\rangle\})$ to P'_L;
(3.3) add $(\langle r, 1\rangle \to \#_1, \emptyset, \bar{N})$ to P'_L;
(3.4) add $(\#_1 \to \langle r, 2\rangle, \{\langle \$, r, 1\rangle\}, \bar{N} - \{\langle \$, r, 1\rangle\})$ to P'_R;
(3.5) add $(\langle \$, r, 1\rangle \to \langle \$, r, 2\rangle, \{\langle r, 2\rangle\}, \bar{N} - \{\langle r, 2\rangle\})$ to P'_L;
(3.6) add $(\langle r, 2\rangle \to \#_2, \emptyset, \bar{N})$ to P'_L;
(3.7) add $(\#_2 \to y, U \cup \{\langle \$, r, 2\rangle\}, W \cup (\bar{N} - \{\langle \$, r, 2\rangle\}))$ to P'_R;
(3.8) add $(\langle \$, r, 2\rangle \to \$, \emptyset, \bar{N})$ to P'_L.

Before proving that $L(H) = L(G)$, let us informally describe the purpose of rules introduced in (1) through (3). The two rules from (1) are used to start and finish every derivation in H. As we want to reduce the number of right random context nonterminals, rules from P_L are simulated directly by rules from (2). An application of a single rule of G from P_R, $r = (A \to y, U, W) \in P_R$, is simulated by rules introduced in (3) in an eight-step way.

During the simulation of applying r, the very last symbol in sentential forms of H always encodes r for the following two reasons. First, as G can contain more than two right random context nonterminals, whenever a nonterminal is rewritten to $\#_1$ or $\#_2$, we keep track regarding the rule that is being simulated. Second, it rules out intermixed simulations of two different rules from P_R, $r = (A \to y, U, W) \in P_R$ and $r' = (A' \to y', U', W') \in P_R$, where $r \neq r'$.

The only purpose of two versions of every compound nonterminal in angular brackets and the symbols $\#_1$, $\#_2$ is to enforce rewriting $\langle \$, r, i\rangle$ back to \$ before another rule from P_R is simulated. In this way, we guarantee that no terminal string out of $L(G)$ is generated.

To establish the identity $L(H) = L(G)$, we prove four claims. Claim 1 shows how H simulates the application of rules from P_L, and how G simulates the application of rules constructed in (2).

Claim 1. In G, $x_1Ax_2 \Rightarrow_G x_1yx_2$ by $(A \to y, U, W) \in P_L$, where $x_1, x_2 \in V^*$, if and only if in H, $x_1Ax_2\$ \Rightarrow_H x_1yx_2\$$ by $(A \to y, U, W \cup \bar{N}) \in P'_L$, introduced in (2).

Proof. Notice that as $x_1 \in V^*$, $\mathrm{alph}(x_1) \cap \bar{N} = \emptyset$. Thus, this claim holds. □

Claim 2 shows how H simulates the application of rules rules from P_R, and how G simulates the application of rules constructed in (3).

Claim 2. In G, $x_1Ax_2 \Rightarrow_G x_1yx_2$ by $r = (A \to y, U, W) \in P_R$, where $x_1, x_2 \in V^*$, if and only if in H, $x_1Ax_2\$ \Rightarrow^8_H x_1yx_2\$$ by the eight rules introduced in (3) from r.

Proof. The proof is divided into the only-if part and the if part.

Only If. Let $x_1Ax_2 \Rightarrow_G x_1yx_2$ by $r = (A \to y, U, W) \in P_R$, where $x_1, x_2 \in V^*$. By (3.1), $(A \to \langle r, 1\rangle, \emptyset, \bar{N}) \in P'_L$. As $\mathrm{alph}(x_1) \cap \bar{N} = \emptyset$,

$$x_1Ax_2\$ \Rightarrow_H x_1\langle r, 1\rangle x_2\$$$

By (3.2), $(\$ \to \langle \$, r, 1\rangle, \{\langle r, 1\rangle\}, \bar{N} - \{\langle r, 1\rangle\}) \in P'_L$. As $\langle r, 1\rangle \in \mathrm{alph}(x_1\langle r, 1\rangle x_2)$ and $\mathrm{alph}(x_1\langle r, 1\rangle x_2) \cap (\bar{N} - \{\langle r, 1\rangle\}) = \emptyset$,

$$x_1\langle r, 1\rangle x_2\$ \Rightarrow_H x_1\langle r, 1\rangle x_2\langle \$, r, 1\rangle$$

By (3.3), $(\langle r, 1\rangle \to \#_1, \emptyset, \bar{N}) \in P'_L$. As $\mathrm{alph}(x_1) \cap \bar{N} = \emptyset$,

$$x_1\langle r, 1\rangle x_2\langle \$, r, 1\rangle \Rightarrow_H x_1\#_1x_2\langle \$, r, 1\rangle$$

By (3.4), $(\#_1 \to \langle r, 2\rangle, \{\langle \$, r, 1\rangle\}, \bar{N} - \{\langle \$, r, 1\rangle\}) \in P'_R$. As $\langle \$, r, 1\rangle \in \mathrm{alph}(x_2\langle \$, r, 1\rangle)$ and $\mathrm{alph}(x_2\langle \$, r, 1\rangle) \cap (\bar{N} - \{\langle \$, r, 1\rangle\}) = \emptyset$,

$$x_1\#_1x_2\langle \$, r, 1\rangle \Rightarrow_H x_1\langle r, 2\rangle x_2\langle \$, r, 1\rangle$$

By (3.5), $(\langle \$, r, 1\rangle \to \langle \$, r, 2\rangle, \{\langle r, 2\rangle\}, \bar{N} - \{\langle r, 2\rangle\}) \in P'_L$. As $\{\langle r, 2\rangle\} \in \mathrm{alph}(x_1\langle r, 2\rangle x_2)$ and $\mathrm{alph}(x_1\langle r, 2\rangle x_2) \cap (\bar{N} - \{\langle r, 2\rangle\}) = \emptyset$,

$$x_1\langle r, 2\rangle x_2\langle \$, r, 1\rangle \Rightarrow_H x_1\langle r, 2\rangle x_2\langle \$, r, 2\rangle$$

By (3.6), $(\langle r, 2\rangle \to \#_2, \emptyset, \bar{N}) \in P'_L$. As $\mathrm{alph}(x_1) \cap \bar{N} = \emptyset$,

$$x_1\langle r, 2\rangle x_2\langle \$, r, 2\rangle \Rightarrow_H x_1\#_2 x_2\langle \$, r, 2\rangle$$

By (3.7), $(\#_2 \to y, U \cup \{\langle \$, r, 2\rangle\}, W \cup (\bar{N} - \{\langle \$, r, 2\rangle\})) \in P'_R$. As $(U \cup \{\langle \$, r, 2\rangle\}) \subseteq \mathrm{alph}(x_2\langle \$, r, 2\rangle)$ and $\mathrm{alph}(x_2\langle \$, r, 2\rangle) \cap (W \cup (\bar{N} - \{\langle \$, r, 2\rangle\})) = \emptyset$,

$$x_1\#_2 x_2\langle \$, r, 2\rangle \Rightarrow_H x_1 y x_2\langle \$, r, 2\rangle$$

Finally, by (3.8), $(\langle \$, r, 2\rangle \to \$, \emptyset, \bar{N}) \in P'_L$. As $\mathrm{alph}(x_1 y x_2) \cap \bar{N} = \emptyset$,

$$x_1 y x_2\langle \$, r, 2\rangle \Rightarrow_H x_1 y x_2\$$$

Hence, the only-if part of the claim holds.

If. Let $x_1Ax_2\$ \Rightarrow_H^8 x_1yx_2\$$ by the eight rules introduced in (3) from some $r = (A \to y, U, W) \in P_R$. Observe that this eight-step derivation is of the following form

$$\begin{aligned} x_1Ax_2 &\Rightarrow_H x_1\langle r, 1\rangle x_2 \\ &\Rightarrow_H x_1\langle r, 1\rangle x_2\langle r, 1\rangle \\ &\Rightarrow_H x_1\#_1 x_2\langle r, 1\rangle \\ &\Rightarrow_H x_1\langle r, 2\rangle x_2\langle r, 1\rangle \\ &\Rightarrow_H x_1\langle r, 2\rangle x_2\langle r, 2\rangle \\ &\Rightarrow_H x_1\#_2 x_2\langle r, 2\rangle \\ &\Rightarrow_H x_1 y x_2\langle r, 2\rangle \\ &\Rightarrow_H x_1 y x_2 \end{aligned}$$

As $x_1\#_2x_2\langle \$, r, 2\rangle \Rightarrow_H x_1yx_2\langle \$, r, 2\rangle$ by $(\#_2 \to y, U \cup \{\langle \$, r, 2\rangle\}, W \cup (\bar{N} - \{\langle \$, r, 2\rangle\})) \in P'_R$, introduced in (3.7) from r, $U \subseteq \mathrm{alph}(x_2)$ and $W \cap \mathrm{alph}(x_2) = \emptyset$. Therefore, by using r,

$$x_1Ax_2 \Rightarrow_G x_1yx_2$$

Hence, the if part of the claim holds. □

Claim 3 shows that every $x \in L(H)$ can be derived in H by a derivation satisfying properties (i) through (iii), stated next. Set $V' = N' \cup T$.

Claim 3. Let $x \in V'^*$. Then, $x \in L(H)$ if and only if $S' \Rightarrow_H S\$ \Rightarrow_H^* x\$ \Rightarrow_H x$ so that during $S\$ \Rightarrow_H^* x\$$, (i) through (iii) hold:

(i) no rules from (1) are used;

(ii) every application of a rule from (3.1) is followed by applying the remaining seven rules from (3.2) through (3.8) before another rule from (3.1) is applied;
(iii) whenever some $A \in N$ is rewritten to $\langle r,\ 1\rangle$ by a rule from (3.1), constructed from $r = (A \to y,\ U,\ W) \in P_R$, $\langle r,\ 1\rangle$ cannot be rewritten to z with $z \neq y$.

Proof. The if part of the claim is trivial; therefore, we only prove the only-if part. Let $x \in L(H)$. We argue that (i) through (iii) hold.

(i) Observe that $(S' \to S\$,\ \emptyset,\ \emptyset) \in P_L$, introduced in (1), is the only rule with S' on its left-hand side. Therefore, this rule is used at the beginning of every derivation. Furthermore, observe that no rule has S' on its right-hand side. By (1), $(\$ \to \varepsilon,\ \emptyset,\ N') \in P_L'$. Notice that this rule is applicable if and only if the current sentential form is of the form $x\$$, where $x \in T^*$. Therefore, (i) holds.
(ii) Let $x_1\langle r,\ 1\rangle x_2\$$ be the first sentential form in $S\$ \Rightarrow_H^* x\$$ after a rule from (3.1) is applied. Clearly, $x_1,\ x_2 \in V^*$. Observe that after this application, the remaining seven rules from (3.2) through (3.8) are applied before another rule from (3.1) is applied. As rules from (3.1) rule out the presence of symbols from $\bar{N}$, a rule from (3.1) can be only applied to some nonterminal in x_1. Then, however, no terminal string can be obtained anymore. Furthermore, observe that during the application of these seven rules, rules introduced in (2) can be applied to x_1 at any time without affecting the applicability of the seven rules to $\langle r,\ 1\rangle x_2\$$. Based on these observations, we see that (ii) holds.
(iii) By (ii), once a rule from (3.1) is used, the remaining seven rules from (3.2) through (3.8) are applied. Observe that when rules from (3.3) and (3.6) are applied, the last nonterminal of the current sentential form encodes the currently simulated rule. Therefore, we cannot combine simulations of two different rules from P_R. Based on this observation, we see that (iii) holds.

Hence, the only-if part of the claim holds, so the claim holds. □

Claim 4. In G, $S \Rightarrow_G^* x$ if and only if in H, $S' \Rightarrow_H^* x$, where $x \in T^*$.

Proof. This claim follows from Claims 1, 2, and 3. □

Claim 4 implies that $L(G) = L(H)$. Clearly, $\#_1$ and $\#_2$ are the only right random context nonterminals in H, so $\mathrm{nrrcn}(H) = 2$. Hence, the lemma holds. □

Theorem 6.2.5. *For every recursively enumerable language K, there exists a one-sided random context grammar H such that $L(H) = K$ and $\mathrm{nrrcn}(H) = 2$.*

Proof. This theorem follows from Theorem 4.1.4 and Lemma 6.2.4. □

Theorem 6.2.6. *For every recursively enumerable language K, there exists a one-sided random context grammar H such that $L(H) = K$ and $\mathrm{nlrcn}(H) = 2$.*

Proof. This theorem can be proved by analogy with the proofs of Theorem 6.2.5 and Lemma 6.2.4. □

Next, we turn our attention to propagating one-sided random context grammars and their nonterminal complexity.

Lemma 6.2.7. *Let G be a propagating one-sided random context grammar. Then, there is a propagating one-sided random context grammar H such that $L(H) = L(G)$ and* $\text{nrrcn}(H) = 2$.

Proof. Let $G = (N, T, P_L, P_R, S)$ be a propagating one-sided random context grammar. We prove this lemma by analogy with the proof of Lemma 6.2.4. However, since H has to be propagating, instead of using a special symbol \$, which is erased at the end of a successful derivation, we use the rightmost symbol of a sentential form for this purpose. Therefore, if X is the rightmost symbol of the current sentential form in G, we use $\langle X\rangle$ in H. By analogy with the construction given in Lemma 6.2.4, we introduce new nonterminals, $\langle X, r, 1\rangle$ and $\langle X, r, 2\rangle$, for every $r \in P_R$ and every $X \in N \cup T$, to keep track of the currently simulated rule r. At the end of a derivation, X has to be a terminal, so instead of erasing $\langle X\rangle$, we rewrite it to X, thus finishing the derivation.

We next construct a propagating one-sided random context grammar H such that $L(H) = L(G)$ and $\text{nrrcn}(H) = 2$. Set $V = N \cup T$ and

$$\begin{aligned} \hat{V} &= \{\langle X\rangle \mid X \in V\} \\ R &= \{\langle r, i\rangle \mid r \in P_R, i = 1, 2\} \\ {}_{\$}R &= \{\langle X, r, i\rangle \mid X \in V, r \in P_R, i = 1, 2\} \end{aligned}$$

Without any loss of generality, we assume that $\hat{V}$, R, ${}_{\$}R$, $\{\#_1, \#_2\}$, and V are pairwise disjoint. Construct

$$H = \left(N', T, P'_L, P'_R, \langle S\rangle\right)$$

as follows. Initially, set $N' = N \cup \hat{V} \cup R \cup {}_{\$}R \cup \{\#_1, \#_2\}$, $P'_L = \emptyset$, and $P'_R = \emptyset$. Furthermore, set $\bar{N} = N' - N$. Perform the following five steps

(1) for each $a \in T$, add $(\langle a\rangle \rightarrow a, \emptyset, N')$ to P'_L;
(2) for each $(A \rightarrow y, U, W) \in P_L$, add $(A \rightarrow y, U, W \cup \bar{N})$ to P'_L;
(3) for each $r = (A \rightarrow y, U, W) \in P_R$ and each $X \in V$,

(3.1) add $(A \rightarrow \langle r, 1\rangle, \emptyset, \bar{N})$ to P'_L;
(3.2) add $(\langle X\rangle \rightarrow \langle X, r, 1\rangle, \{\langle r, 1\rangle\}, \bar{N} - \{\langle r, 1\rangle\})$ to P'_L;
(3.3) add $(\langle r, 1\rangle \rightarrow \#_1, \emptyset, \bar{N})$ to P'_L;
(3.4) add $(\#_1 \rightarrow \langle r, 2\rangle, \{\langle X, r, 1\rangle\}, \bar{N} - \{\langle X, r, 1\rangle\})$ to P'_R;
(3.5) add $(\langle X, r, 1\rangle \rightarrow \langle X, r, 2\rangle, \{\langle r, 2\rangle\}, \bar{N} - \{\langle r, 2\rangle\})$ to P'_L;
(3.6) add $(\langle r, 2\rangle \rightarrow \#_2, \emptyset, \bar{N})$ to P'_L;

(3.7) add $(\#_2 \to y, U \cup \{\langle X, r, 2\rangle\}, W \cup (\bar{N} - \{\langle X, r, 2\rangle\}))$ to P'_R;
(3.8) add $(\langle X, r, 2\rangle \to \langle X\rangle, \emptyset, \bar{N})$ to P'_L;

(4) for each $(A \to yY, U, W) \in P_L$, where $y \in V^*$ and $Y \in V$, add $(\langle A\rangle \to y\langle Y\rangle, U, W \cup \bar{N})$ to P'_L;
(5) for each $(A \to yY, \emptyset, W) \in P_R$, where $y \in V^*$ and $Y \in V$, add $(\langle A\rangle \to y\langle Y\rangle, \emptyset, \bar{N})$ to P'_L.

Steps (1) through (3) are similar to the corresponding three steps in the construction given in the proof of Lemma 6.2.4. Rules from (4) and (5) take care of rewriting the rightmost nonterminal. Note that every simulated rule from P_R rewriting this nonterminal has to have its permitting context empty; otherwise, it is not applicable to the rightmost nonterminal. Furthermore, observe that we can simulate such a right random context rule by a left random context rule. As obvious, there are no nonterminals to the right of the rightmost symbol.

The identity $L(H) = L(G)$ can be proved by analogy with proving Lemma 6.2.4, and we leave this proof to the reader. Clearly, $\#_1$ and $\#_2$ are the only right random context nonterminals in H, so $\text{nrrcn}(H) = 2$. Hence, the lemma holds. □

Theorem 6.2.8. *For every context-sensitive language K, there exists a propagating one-sided random context grammar H such that* $L(H) = K$ *and* $\text{nrrcn}(H) = 2$.

Proof. This theorem follows from Theorem 4.1.3 and Lemma 6.2.7. □

Theorem 6.2.9. *For every context-sensitive language K, there exists a propagating one-sided random context grammar H such that* $L(H) = K$ *and* $\text{nlrcn}(H) = 2$.

Proof. This theorem can be proved by analogy with the proofs of Theorem 6.2.8 and Lemma 6.2.7. □

6.3 Number of Right Random Context Rules

In this section, we prove that any recursively enumerable language can be generated by a one-sided random context grammar having no more than two right random context rules.

Theorem 6.3.1. *Let K be a recursively enumerable language. Then, there is a one-sided random context grammar,* $H = (N, T, P_L, P_R, S)$, *such that* $L(H) = K$ *and* $\text{card}(P_R) = 2$.

Proof. Let K be a recursively enumerable language. Then, by Theorem 2.3.12, there is a phrase-structure grammar in the Geffert normal form

$$G = \big(\{S,A,B,C\},T,P \cup \{ABC \to \varepsilon\},S\big)$$

satisfying $L(G) = K$. We next construct a one-sided random context grammar H such that $L(H) = L(G)$. Set $N = \{S, A, B, C\}$, $V = N \cup T$, and $N' = N \cup \{S', \$, \bar{\$}, \#, \bar{A}, \bar{B}, \hat{A}, \hat{B}, \hat{C}\}$. Without any loss of generality, we assume that $V \cap \{S', \$, \bar{\$}, \#, \bar{A}, \bar{B}, \hat{A}, \hat{B}, \hat{C}\} = \emptyset$. Construct

$$H = \left(N', T, P_L, P_R, S'\right)$$

as follows. Initially, set $P_L = \emptyset$ and $P_R = \emptyset$. Perform the following eleven steps

(1) add $(S' \to \$S, \emptyset, \emptyset)$ to P_L;
(2) for each $S \to uSa \in P$, where $u \in \{A, AB\}^*$ and $a \in T$,
add $(S \to uS\#a, \emptyset, \{\bar{A}, \bar{B}, \hat{A}, \hat{B}, \hat{C}, \#\})$ to P_L;
(3) for each $S \to uSv \in P$, where $u \in \{A, AB\}^*$ and $v \in \{BC, C\}^*$,
add $(S \to uSv, \emptyset, \{\bar{A}, \bar{B}, \hat{A}, \hat{B}, \hat{C}, \#\})$ to P_L;
(4) for each $S \to uv \in P$, where $u \in \{A, AB\}^*$ and $v \in \{BC, C\}^*$,
add $(S \to uv, \emptyset, \{\bar{A}, \bar{B}, \hat{A}, \hat{B}, \hat{C}, \#\})$ to P_L;
(5) add $(A \to \bar{A}, \emptyset, N \cup \{\hat{\$}, \hat{A}, \hat{B}, \hat{C}, \#\})$ to P_L;
add $(B \to \bar{B}, \emptyset, N \cup \{\hat{\$}, \hat{A}, \hat{B}, \hat{C}, \#\})$ to P_L;
add $(A \to \hat{A}, \emptyset, N \cup \{\hat{\$}, \hat{A}, \hat{B}, \hat{C}, \#\})$ to P_L;
add $(B \to \hat{B}, \{\hat{A}\}, N \cup \{\hat{\$}, \hat{B}, \hat{C}, \#\})$ to P_L;
add $(C \to \hat{C}, \{\hat{A}, \hat{B}\}, N \cup \{\hat{\$}, \hat{C}, \#\})$ to P_L;
(6) add $(\$ \to \hat{\$}, \{\hat{A}, \hat{B}, \hat{C}\}, \emptyset)$ to P_R;
(7) add $(\hat{A} \to \varepsilon, \{\hat{\$}\}, N \cup \{\hat{A}, \hat{B}, \hat{C}, \#\})$ to P_L;
add $(\hat{B} \to \varepsilon, \{\hat{\$}\}, N \cup \{\hat{A}, \hat{B}, \hat{C}, \#\})$ to P_L;
add $(\hat{C} \to \varepsilon, \{\hat{\$}\}, N \cup \{\hat{A}, \hat{B}, \hat{C}, \#\})$ to P_L;
(8) add $(\bar{A} \to A, \{\hat{\$}\}, N \cup \{\hat{A}, \hat{B}, \hat{C}, \#\})$ to P_L;
add $(\bar{B} \to B, \{\hat{\$}\}, N \cup \{\hat{A}, \hat{B}, \hat{C}, \#\})$ to P_L;
(9) add $(\hat{\$} \to \$, \emptyset, \{\bar{A}, \bar{B}, \hat{A}, \hat{B}, \hat{C}\})$ to P_R;
(10) add $(\$ \to \varepsilon, \emptyset, \emptyset)$ to P_L;
(11) add $(\# \to \varepsilon, \emptyset, N')$ to P_L.

Before proving that $L(H) = L(G)$, let us informally describe the meaning of rules introduced in (1) through (11). The rule from (1) starts every derivation of H. The leftmost symbol of every sentential form having at least one nonterminal is either \$ or $\hat{\$}$. The role of these two symbols is explained later. H simulates the derivations of G that satisfy the form described in Theorem 2.3.12. The context-free rules in P are simulated by rules from (2) through (4). The context-sensitive rule $ABC \to \varepsilon$ is simulated in a several-step way. First, rules introduced in (5) are used to prepare the erasure of ABC. These rules rewrite nonterminals from the left to the right. In this way, it is guaranteed that whenever $\hat{A}$, $\hat{B}$, and $\hat{C}$ appear in a sentential form, then they form a substring of the form $\hat{A}\hat{B}\hat{C}$. Then, \$ is changed to $\hat{\$}$ by using the rule from (6). After that, the rules from (7) erase $\hat{A}$, $\hat{B}$, and $\hat{C}$, one by one. Finally, rules from (8) convert the barred versions of nonterminals back to their non-barred versions to prepare another simulation of $ABC \to \varepsilon$; this conversion is done in a left-to-right way. After this conversion, $\hat{\$}$ is reverted back to \$ by the rule from (9). For example, $AABCBCab \Rightarrow_G ABCab$ is simulated by H as follows:

$$
\begin{aligned}
\$AABCBC\#a\#b &\Rightarrow_H \$\bar{A}ABCBC\#a\#b \\
&\Rightarrow_H \$\bar{A}\hat{A}BCBC\#a\#b \\
&\Rightarrow_H \$\bar{A}\hat{A}\hat{B}CBC\#a\#b \\
&\Rightarrow_H \$\bar{A}\hat{A}\hat{B}\hat{C}BC\#a\#b \\
&\Rightarrow_H \hat{\$}\bar{A}\hat{A}\hat{B}\hat{C}BC\#a\#b \\
&\Rightarrow_H \hat{\$}\bar{A}\hat{B}\hat{C}BC\#a\#b \\
&\Rightarrow_H \hat{\$}\bar{A}\hat{C}BC\#a\#b \\
&\Rightarrow_H \hat{\$}\bar{A}BC\#a\#b \\
&\Rightarrow_H \hat{\$}ABC\#a\#b \\
&\Rightarrow_H \$ABC\#a\#b
\end{aligned}
$$

Symbol # is used to ensure that every sentential form of H is of the form w_1w_2, where $w_1 \in (N' - \{\#\})^*$ and $w_2 \in (T \cup \{\#\})^*$. Since permitting and forbidding contexts cannot contain terminals, a mixture of symbols from T and N in H could produce a terminal string out of $L(G)$. For example, observe that $\$AaBC \Rightarrow_H^* \$a$ by rules from (5) through (9), but such a derivation does not exist in G. #s can be eliminated by an application of rules from (11) provided that no nonterminals occur to the left of # in the current sentential form. Consequently, all #s are erased at the end of every successful derivation.

The leftmost symbol \$ and its hatted version $\hat{\$}$ encode the current phase. When \$ is present, we use rules from (2) through (5). When $\hat{\$}$ is present, we use rules from (7) and (8). When none of these two symbols is present, which happens after the rule from (10) is applied, no substring ABC can be erased anymore so we have to finish the derivation by removing all #*s*. Notice that when \$ is erased prematurely, no terminal string can be derived.

To establish the identity $L(H) = L(G)$, we prove two claims. Claim 1 shows how derivations of G are simulated by H. Then, Claim 2 demonstrates the converse simulation—that is, it shows how derivations of H are simulated by G.

Set $V' = N' \cup T$. Define the homomorphism φ from V^* to V'^* as $\varphi(X) = X$ for $X \in N$, and $\varphi(a) = \#a$ for $a \in T$.

Claim 1. If $S \Rightarrow_G^n x \Rightarrow_G^* z$, where $x \in V^*$ and $z \in T^*$, for some $n \geq 0$, then $S' \Rightarrow_H^*$ $\$\varphi(x)$.

Proof. This claim is established by induction on $n \geq 0$.

Basis. For $n = 0$, this claim is clear (in H, we use $(S' \to \$S, \emptyset, \emptyset)$ from (1)).

Induction Hypothesis. Suppose that there exists $n \geq 0$ such that the claim holds for all derivations of length ℓ, where $0 \leq \ell \leq n$.

Induction Step. Consider any derivation of the form

$$S \Rightarrow_G^{n+1} w \Rightarrow_G^* z$$

where $w \in V^*$ and $z \in T^*$. Since $n+1 \geq 1$, this derivation can be expressed as

$$S \Rightarrow_G^n x \Rightarrow_G w \Rightarrow_G^* z$$

for some $x \in V^+$. Without any loss of generality, we assume that x is of the form $x = x_1x_2x_3x_4$, where $x_1 \in \{A, AB\}^*$, $x_2 \in \{S, \varepsilon\}$, $x_3 \in \{BC, C\}^*$, and $x_4 \in T^*$ (see Theorem 2.3.12 and [29]).

Next, we consider all possible forms of $x \Rightarrow_G w$, covered by the following four cases—(i) through (iv).

(i) *Application of* $S \to uSa \in P$. Let $x = x_1Sx_3x_4$, $w = x_1uSax_3x_4$, and $S \to uSa \in P$, where $x_1, u \in \{A, AB\}^*$, $x_3 \in \{BC, C\}^*$, $x_4 \in T^*$, and $a \in T$. Then, by the induction hypothesis,

$$S' \Rightarrow_H^* \$\varphi(x_1Sx_3x_4)$$

By (2), $(S \to uS\#a, \emptyset, \{\bar{A}, \bar{B}, \hat{A}, \hat{B}, \hat{C}, \#\}) \in P_L$. Since $\$\varphi(x_1Sx_3x_4) = \$x_1S\varphi(x_3x_4)$ and $\mathrm{alph}(\$x_1) \cap \{\bar{A}, \bar{B}, \hat{A}, \hat{B}, \hat{C}, \#\} = \emptyset$,

$$\$x_1S\varphi(x_3x_4) \Rightarrow_H \$x_1uS\#a\varphi(x_3x_4)$$

As $x_1uS\#a\varphi(x_3x_4) = \varphi(x_1uSax_3x_4)$, the induction step is completed for (i).

(ii) *Application of* $S \to uSv \in P$. Let $x = x_1Sx_3x_4$, $w = x_1uSvx_3x_4$, and $S \to uSv \in P$, where $x_1, u \in \{A, AB\}^*$, $x_3, v \in \{BC, C\}^*$, and $x_4 \in T^*$. To complete the induction step for (ii), proceed by analogy with (i), but use a rule from (3) instead of a rule from (1).

(iii) *Application of* $S \to uv \in P$. Let $x = x_1Sx_3x_4$, $w = x_1uvx_3x_4$, and $S \to uv \in P$, where $x_1, u \in \{A, AB\}^*$, $x_3, v \in \{BC, C\}^*$, and $x_4 \in T^*$. To complete the induction step for (iii), proceed by analogy with (i), but use a rule from (4) instead of a rule from (1).

(iv) *Application of* $ABC \to \varepsilon$. Let $x = x_1ABCx_3x_4$ and $w = x_1x_3x_4$, where $x_1 \in \{A, AB\}^*$, $x_3 \in \{BC, C\}^*$, and $x_4 \in T^*$, so $x \Rightarrow_G w$ by $ABC \to \varepsilon$. Then, by the induction hypothesis,

$$S' \Rightarrow_H^* \$\varphi(x_1ABCx_3x_4)$$

Let $x_1 = X_1X_2 \cdots X_k$, where $k = |x_1|$ (the case when $k = 0$ means that $x_1 = \varepsilon$). Since $\$\varphi(x_1ABCx_3x_4) = \$x_1ABC\varphi(x_3x_4)$ and $\mathrm{alph}(x_1) \subseteq N$, by rules introduced in (5),

$$
\begin{aligned}
\$X_1X_2\cdots X_kABC\varphi(x_3x_4) &\Rightarrow_H \$\bar{X}_1X_2\cdots X_kABC\varphi(x_3x_4)\\
&\Rightarrow_H \$\bar{X}_1\bar{X}_2\cdots X_kABC\varphi(x_3x_4)\\
&\ \ \vdots\\
&\Rightarrow_H \$\bar{X}_1\bar{X}_2\cdots \bar{X}_kABC\varphi(x_3x_4)\\
&\Rightarrow_H \$\bar{X}_1\bar{X}_2\cdots \bar{X}_k\hat{A}BC\varphi(x_3x_4)\\
&\Rightarrow_H \$\bar{X}_1\bar{X}_2\cdots \bar{X}_k\hat{A}\hat{B}C\varphi(x_3x_4)\\
&\Rightarrow_H \$\bar{X}_1\bar{X}_2\cdots \bar{X}_k\hat{A}\hat{B}\hat{C}\varphi(x_3x_4)
\end{aligned}
$$

Let $\bar{x}_1 = \bar{X}_1\bar{X}_2\cdots\bar{X}_k$. By the rule introduced in (6),

$$\$\bar{x}_1\hat{A}\hat{B}\hat{C}\varphi(x_3x_4) \Rightarrow_H \hat{\$}\bar{x}_1\hat{A}\hat{B}\hat{C}\varphi(x_3x_4)$$

By the rules introduced in (7),

$$
\begin{aligned}
\hat{\$}\bar{x}_1\hat{A}\hat{B}\hat{C}\varphi(x_3x_4) &\Rightarrow_H \hat{\$}\bar{x}_1\hat{B}\hat{C}\varphi(x_3x_4)\\
&\Rightarrow_H \hat{\$}\bar{x}_1\hat{C}\varphi(x_3x_4)\\
&\Rightarrow_H \hat{\$}\bar{x}_1\varphi(x_3x_4)
\end{aligned}
$$

By rules from (8),

$$
\begin{aligned}
\hat{\$}\bar{x}_1\varphi(x_3x_4) &\Rightarrow_H \hat{\$}\bar{X}_1\cdots\bar{X}_{k-1}X_k\varphi(x_3x_4)\\
&\Rightarrow_H \hat{\$}\bar{X}_1\cdots X_{k-1}X_k\varphi(x_3x_4)\\
&\ \ \vdots\\
&\Rightarrow_H \hat{\$}X_1\cdots X_{k-1}X_k\varphi(x_3x_4)
\end{aligned}
$$

Recall that $X_1\cdots X_{k-1}X_k = x_1$. Finally, by the rule from (9),

$$\hat{\$}x_1\varphi(x_3x_4) \Rightarrow_H \$x_1\varphi(x_3x_4)$$

Since $x_1\varphi(x_3x_4) = \varphi(x_1x_3x_4)$, the induction step is completed for (iv).

Observe that cases (i) through (iv) cover all possible forms of $x \Rightarrow_G w$. Thus, the claim holds. □

Define the homomorphism π from $(V' - \{S'\})^*$ to V^* as $\pi(X) = X$ for $X \in N$, $\pi(\bar{A}) = \pi(\hat{A}) = A$, $\pi(\bar{B}) = \pi(\hat{B}) = B$, $\pi(\hat{C}) = C$, $\pi(a) = a$ for $a \in T$, and $\pi(\#) = \pi(\$) = \pi(\hat{\$}) = \varepsilon$. Define the homomorphism τ from $(V' - \{S'\})^*$ to V^* as $\tau(X) = \pi(X)$ for $X \in V' - \{S', \hat{A}, \hat{B}, \hat{C}\}$, and $\tau(\hat{A}) = \tau(\hat{B}) = \tau(\hat{C}) = \varepsilon$.

Claim 2. Let $S' \Rightarrow_H^n x \Rightarrow_H^* z$, where $x \in V'^*$ and $z \in T^*$, for some $n \geq 1$. Then, $x = x_0x_1x_2x_3x_4x_5$, where $x_0 \in \{\varepsilon, \$, \hat{\$}\}$, $x_1 \in \{\bar{A}, \bar{B}\}^*$, $x_2 \in \{A, B\}^*$, $x_3 \in \{S, \hat{A}BC, \hat{A}\hat{B}C, \hat{A}\hat{B}\hat{C}, \hat{B}\hat{C}, \hat{C}, \varepsilon\}$, $x_4 \in \{B, C\}^*$, and $x_5 \in (T \cup \{\#\})^*$. Furthermore,

(a) if $x_3 \in \{S, \varepsilon\}$, then $S \Rightarrow_G^* \pi(x)$;

(b) if $x_3 \in \{\hat{A}BC, \hat{A}\hat{B}C, \hat{A}\hat{B}\hat{C}\}$*, then* $x_2 = \varepsilon$ *and* $S \Rightarrow_G^* \pi(x)$*;*
(c) if $x_3 \in \{\hat{B}\hat{C}, \hat{C}\}$*, then* $x_0 = \hat{\$}$, $x_2 = \varepsilon$*, and* $S \Rightarrow_G^* \tau(x)$.

Proof. This claim is established by induction on $n \geq 1$.

Basis. For $n = 1$, this claim is clear (the only applicable rule to S' is $(S' \rightarrow \$S, \emptyset, \emptyset) \in P_L$, introduced in (1)).

Induction Hypothesis. Suppose that there exists $n \geq 1$ such that the claim holds for all derivations of length ℓ, where $1 \leq \ell \leq n$.

Induction Step. Consider any derivation of the form

$$S' \Rightarrow_H^{n+1} w \Rightarrow_H^* z$$

where $w \in V'^*$ and $z \in T^*$. Since $n+1 \geq 1$, this derivation can be expressed as

$$S' \Rightarrow_H^n x \Rightarrow_H w \Rightarrow_H^* z$$

for some $x \in V'^+$. By the induction hypothesis, $x = x_0x_1x_2x_3x_4x_5$, where $x_0 \in \{\varepsilon, \$, \hat{\$}\}$, $x_1 \in \{\bar{A}, \bar{B}\}^*$, $x_2 \in \{A, B\}^*$, $x_3 \in \{S, \hat{A}BC, \hat{A}\hat{B}C, \hat{A}\hat{B}\hat{C}, \hat{B}\hat{C}, \hat{C}, \varepsilon\}$, $x_4 \in \{B, C\}^*$, and $x_5 \in (T \cup \{\#\})^*$. Furthermore, (a) through (c), stated in the claim, hold.

Next, we consider all possible forms of $x \Rightarrow_H w$, covered by the following six cases—(i) through (vi).

(i) *Application of a rule from (2).* Let $x_3 = S$, $x_1 = x_4 = \varepsilon$, and

$$(S \rightarrow uS\#a, \emptyset, \{\bar{A}, \bar{B}, \hat{A}, \hat{B}, \hat{C}, \#\}) \in P_L$$

introduced in (2), where $u \in \{A, AB\}^*$ and $a \in T$, so

$$x_0x_2Sx_5 \Rightarrow_H x_0x_2uS\#ax_5$$

Observe that if $x_4 \neq \varepsilon$, then $w \Rightarrow_H^* z$ does not hold. Indeed, if $x_4 \neq \varepsilon$, then to erase the nonterminals in x_4, there have to be As in x_2. However, the # symbol, introduced between x_2 and x_4, blocks the applicability of $(C \rightarrow \hat{C}, \{\hat{A}, \hat{B}\}, N \cup \{\hat{\$}, \hat{C}, \#\}) \in P_L$, introduced in (5), which is needed to erase the nonterminals in x_4. Since $(\# \rightarrow \varepsilon, \emptyset, N') \in P_L$, introduced in (11), requires that there are no nonterminals to the left of #, the derivation cannot be successfully finished. Hence, $x_4 = \varepsilon$. Since $u \in \{A, B\}^*$ and $\#a \in (T \cup \{\#\})^*$, $x_0x_2uS\#ax_5$ is of the required form. As $x_3 = S$, by (a), $S \Rightarrow_G^* \pi(x)$. Observe that $\pi(x) = \pi(x_0x_2)S\pi(x_5)$. By (2), $S \rightarrow uSa \in P$, so

$$\pi(x_0x_2)S\pi(x_5) \Rightarrow_G \pi(x_0x_2)uSa\pi(x_5)$$

Since $\pi(x_0x_2)uSa\pi(x_5) = \pi(x_0x_2uS\#ax_5)$ and both $\hat{B}\hat{C}$ and $\hat{C}$ are not substrings of $x_0x_2uS\#ax_5$, the induction step is completed for (i).

(ii) *Application of a rule from (3).* Make this part of the proof by analogy with (i).

(iii) *Application of a rule from (4).* Make this part of the proof by analogy with (i).

(iv) *Application of a rule from (5), (6), (9), (10), or (11).* Let $x \Rightarrow_H w$ by a rule from (5), (6), (9), (10), or (11). Then, $S \Rightarrow_G^* \pi(w)$ follows directly from the induction hypothesis. Observe that w is of the required form, and the induction step is completed for (iv).

(v) *Application of a rule from (7).* Let $x_3 \in \{\hat{A}\hat{B}\hat{C}, \hat{B}\hat{C}, \hat{C}\}$. By the induction hypothesis (more specifically, by (b) and (c)), $x_2 = \varepsilon$. Then, there are three subcases, depending on x_3, as demonstrated next.

(v.i) Let $x_3 = \hat{A}\hat{B}\hat{C}$. Then, $x_0x_1\hat{A}\hat{B}\hat{C}x_4x_5 \Rightarrow_H x_0x_1\hat{B}\hat{C}x_4x_5$ by $(\hat{A} \rightarrow \varepsilon, \{\hat{\$}\}, N \cup \{\hat{A}, \hat{B}, \hat{C}, \#\}) \in P_L$, introduced in (7). Observe that this is the only applicable rule from (7). By the induction hypothesis, $S \Rightarrow_G^* \pi(x)$. Since $\pi(x) = \pi(x_1)ABC\pi(x_4x_5)$,

$$\pi(x_1)ABC\pi(x_4x_5) \Rightarrow_G \pi(x_1)\pi(x_4x_5)$$

by $ABC \rightarrow \varepsilon$. As $w = x_0x_1\hat{A}\hat{C}x_4x_5$ is of the required form and $\pi(x_0x_1)\pi(x_4x_5) = \tau(w)$, the induction step is completed for (v.i).

(v.ii) Let $x_3 = \hat{B}\hat{C}$. Then, $x_0x_1\hat{B}\hat{C}x_4x_5 \Rightarrow_H x_0x_1\hat{C}x_4x_5$ by $(\hat{B} \rightarrow \varepsilon, \{\hat{\$}\}, N \cup \{\hat{A}, \hat{B}, \hat{C}, \#\}) \in P_L$, introduced in (7). Observe that this is the only applicable rule from (7). By the induction hypothesis, $S \Rightarrow_G^* \tau(x)$. As $w = x_0x_1\hat{C}x_4x_5$ is of the required form and $\tau(x) = \tau(w)$, the induction step is completed for (v.ii).

(v.iii) Let $x_3 = \hat{C}$. Then, $x_0x_1\hat{C}x_4x_5 \Rightarrow_H x_0x_1x_4x_5$ by $(\hat{C} \rightarrow \varepsilon, \{\hat{\$}\}, N \cup \{\hat{A}, \hat{B}, \hat{C}, \#\}) \in P_L$, introduced in (7). Observe that this is the only applicable rule from (7). By the induction hypothesis, $S \Rightarrow_G^* \tau(x)$. As $w = x_0x_1x_4x_5$ is of the required form and $\tau(x) = \tau(w)$, the induction step is completed for (v.iii).

(vi) *Application of a rule from (8).* Let $x \Rightarrow_H w$ by a rule from (8). Then, $x_3 \notin \{\hat{B}\hat{C}, \hat{C}\}$ has to hold; otherwise, no string of terminals can be obtained anymore. Indeed, the deletion of $\hat{B}$ and $\hat{C}$ requires that there are no symbols from N to the left of them, and to rewrite A or B to their barred versions, $\hat{\$}$ cannot be present to the left of them. However, by (c), it is there. Therefore, $S \Rightarrow_G^* \pi(w)$ follows directly from the induction hypothesis. Furthermore, w is of the required form; if not, then observe that no string of terminals can be obtained anymore. Hence, the induction step is completed for (vi).

Observe that cases (i) through (vi) cover all possible forms of $x \Rightarrow_H w$. Thus, the claim holds. □

We next prove that $L(H) = L(G)$. Consider Claim 1 when $x \in T^*$. Then, $S' \Rightarrow_H^* \$\varphi(x)$. By $(\$ \rightarrow \varepsilon, \emptyset, \emptyset) \in P_L$, introduced in (10),

$$\$\varphi(x) \Rightarrow_H \varphi(x)$$

Let $x = a_1 a_2 \cdots a_k$, where $k = |x|$ (the case when $k = 0$ means that $x = \varepsilon$), so $\varphi(x) = \#a_1\#a_2 \cdots \#a_k$. By (11), $(\# \rightarrow \varepsilon, \emptyset, N') \in P_L$, so

$$\begin{aligned}
\#a_1\#a_2 \cdots \#a_k &\Rightarrow_H a_1\#a_2 \cdots \#a_k \\
&\Rightarrow_H a_1 a_2 \cdots \#a_k \\
&\ \ \vdots \\
&\Rightarrow_H a_1 a_2 \cdots a_k
\end{aligned}$$

Hence, $x \in L(G)$ implies that $x \in L(H)$, so $L(G) \subseteq L(H)$.

Consider Claim 2 when $x \in T^*$. Then, $S \Rightarrow_G^* \pi(x)$. Since $x \in T^*$, $\pi(x) = x$. Hence, $x \in L(H)$ implies that $x \in L(G)$, so $L(H) \subseteq L(G)$.

The two inclusions, $L(G) \subseteq L(H)$ and $L(H) \subseteq L(G)$, imply that $L(H) = L(G)$. Since $\mathrm{card}(P_R) = 2$, the theorem holds. □

From Theorem 6.3.1 and its proof, we obtain the following corollary, which strengthens Theorem 6.2.5.

Corollary 6.3.2. *Let K be a recursively enumerable language. Then, there is a one-sided random context grammar, $H = (N, T, P_L, P_R, S)$, such that $L(H) = K$,* $\mathrm{card}(N) = 13$, $\mathrm{nrrcn}(H) = 2$, *and* $\mathrm{card}(P_R) = 2$. □

We close this section by suggesting two important open problem areas.

Open Problem 6.3.3. Can the achieved results be improved? Especially, reconsider Theorem 6.2.5. By proving that every one-sided random context grammar G can be converted into an equivalent one-sided random context H with no right random context nonterminals, we would establish the generative power of left random context grammars (see Definition 3.1.5 and Open Problem 4.3.7). □

Open Problem 6.3.4. Recall that propagating one-sided random context grammars characterize the family of context-sensitive languages (see Theorem 4.1.3). Can we also limit the overall number of nonterminals in terms of this propagating version like in Theorem 6.1.1? □

Chapter 7
Leftmost Derivations

The investigation of grammars that perform leftmost derivations is central to formal language theory as a whole. Indeed, from a practical viewpoint, leftmost derivations fulfill a crucial role in parsing, which represents a key application area of formal grammars (see [1, 2, 12, 70]). From a theoretical viewpoint, an effect of leftmost derivation restrictions to the power of grammars restricted in this way represents an intensively investigated area of this theory as clearly indicated by many studies on the subject. More specifically, [4, 5, 52, 64, 98] contain fundamental results concerning leftmost derivations in classical Chomsky grammars, [9, 30, 65, 93, 101] and Section 5.3 in [16] give an overview of the results concerning leftmost derivations in regulated grammars published until late 1980's, and [14, 22, 23, 68, 72, 75] together with Section 7.3 in [73] present several follow-up results. In addition, [38, 51, 94] cover language-defining devices introduced with some kind of leftmost derivations, and [8] discusses the recognition complexity of derivation languages of various regulated grammars with leftmost derivations. Finally, [51, 71, 88] study grammar systems working under the leftmost derivation restriction, and [27, 28, 89] investigates leftmost derivations in terms of P systems.

Considering the significance of leftmost derivations, it comes as no surprise that the present chapter pays a special attention to them. Indeed, it introduces three types of leftmost derivation restrictions placed upon one-sided random context grammars. In the *type*-1 *derivation restriction*, discussed in Section 7.1, during every derivation step, the leftmost occurrence of a nonterminal has to be rewritten. In the *type*-2 *derivation restriction*, covered in Section 7.2, during every derivation step, the leftmost occurrence of a nonterminal which can be rewritten has to be rewritten. In the *type*-3 *derivation restriction*, studied in Section 7.2, during every derivation step, a rule is chosen, and the leftmost occurrence of its left-hand side is rewritten.

In this chapter, we place the three above-mentioned leftmost derivation restrictions on one-sided random context grammars, and prove results (I) through (III), given next.

(I) One-sided random context grammars with type-1 leftmost derivations characterize **CF** (Theorem 7.1.4). An analogous result holds for propagating one-sided random context grammars (Theorem 7.1.5).
(II) One-sided random context grammars with type-2 leftmost derivations characterize **RE** (Theorem 7.2.4). Propagating one-sided random context grammars with type-2 leftmost derivations characterize **CS** (Theorem 7.2.6).
(III) One-sided random context grammars with type-3 leftmost derivations characterize **RE** (Theorem 7.3.4). Propagating one-sided random context grammars with type-3 leftmost derivations characterize **CS** (Theorem 7.3.6).

7.1 Type-1 Leftmost Derivations

In the first derivation restriction type, during every derivation step, the leftmost occurrence of a nonterminal has to be rewritten. This type of leftmost derivations corresponds to the leftmost derivations in context-free grammars (see Definition 2.3.6).

Definition 7.1.1. Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. The *type*-1 *direct leftmost derivation relation* over V^*, symbolically denoted by ${}_{\text{lm}}^{\ 1}\!\!\Rightarrow_G$, is defined as follows. Let $u \in T^*$, $A \in N$ and $x, v \in V^*$. Then,

$$uAv\ {}_{\text{lm}}^{\ 1}\!\!\Rightarrow_G uxv$$

if and only if

$$uAv \Rightarrow_G uxv$$

Let ${}_{\text{lm}}^{\ 1}\!\!\Rightarrow_G^n$ and ${}_{\text{lm}}^{\ 1}\!\!\Rightarrow_G^*$ denote the nth power of ${}_{\text{lm}}^{\ 1}\!\!\Rightarrow_G$, for some $n \geq 0$, and the reflexive-transitive closure of ${}_{\text{lm}}^{\ 1}\!\!\Rightarrow_G$, respectively. The ${}_{\text{lm}}^{\ 1}$-*language* of G is denoted by $L(G, {}_{\text{lm}}^{\ 1}\!\!\Rightarrow)$ and defined as

$$L\big(G, {}_{\text{lm}}^{\ 1}\!\!\Rightarrow\big) = \big\{w \in T^* \mid S\ {}_{\text{lm}}^{\ 1}\!\!\Rightarrow_G^* w\big\} \qquad \square$$

Notice that if the leftmost occurrence of a nonterminal cannot be rewritten by any rule, then the derivation is blocked.

The language families generated by one-sided random context grammars with type-1 leftmost derivations and propagating one-sided random context grammars with type-1 leftmost derivations are denoted by $\mathbf{ORC}({}_{\text{lm}}^{\ 1}\!\!\Rightarrow)$ and $\mathbf{ORC}^{-\varepsilon}({}_{\text{lm}}^{\ 1}\!\!\Rightarrow)$, respectively.

Next, we prove that $\mathbf{ORC}({}_{\text{lm}}^{\ 1}\!\!\Rightarrow) = \mathbf{ORC}^{-\varepsilon}({}_{\text{lm}}^{\ 1}\!\!\Rightarrow) = \mathbf{CF}$.

Lemma 7.1.2. *For every context-free grammar G, there is a one-sided random context grammar H such that $L(H, {}_{\text{lm}}^{\ 1}\!\!\Rightarrow) = L(G)$. Furthermore, if G is propagating, then so is H.*

Proof. Let $G = (N, T, P, S)$ be a context-free grammar. Construct the one-sided random context grammar

$$H = \big(N, T, P', P', S\big)$$

where

$$P' = \big\{(A \rightarrow x, \emptyset, \emptyset) \mid A \rightarrow x \in P\big\}$$

As the rules in P' have their permitting and forbidding contexts empty, any successful type-1 leftmost derivation in H is also a successful derivation in G, so the inclusion $L(H, {}_{\rm lm}{}^{1}\!\Rightarrow) \subseteq L(G)$ holds. On the other hand, let $w \in L(G)$ be a string successfully generated by G. Then, there exists a successful leftmost derivation of w in G (see Theorem 2.3.7). Observe that such a leftmost derivation is also possible in H. Thus, the other inclusion $L(G) \subseteq L(H, {}_{\rm lm}{}^{1}\!\Rightarrow)$ holds as well. Finally, notice that whenever G is propagating, so is H. Hence, the theorem holds. □

Lemma 7.1.3. *For every one-sided random context grammar G, there is a context-free grammar H such that* $L(H) = L(G, {}_{\rm lm}{}^{1}\!\Rightarrow)$. *Furthermore, if G is propagating, then so is H.*

Proof. Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. In what follows, angle brackets $\langle$ and $\rangle$ are used to incorporate more symbols into a single compound symbol. Construct the context-free grammar

$$H = \big(N', T, P, \langle S, \emptyset\rangle\big)$$

in the following way. Initially, set

$$N' = \big\{\langle A, Q\rangle \mid A \in N, Q \subseteq N\big\}$$

and $P = \emptyset$. Without any loss of generality, assume that $N' \cap V = \emptyset$. Perform (1) and (2), given next.

(1) For each $(A \rightarrow y_0Y_1y_1Y_2y_2 \cdots Y_hy_h, U, W) \in P_R$, where $y_i \in T^*$, $Y_j \in N$, for all i and j, $0 \leq i \leq h$, $1 \leq j \leq h$, for some $h \geq 0$, and for each $\langle A, Q\rangle \in N'$ such that $U \subseteq Q$ and $W \cap Q = \emptyset$, extend P by adding

$$\begin{aligned}\langle A, Q\rangle \rightarrow y_0&\langle Y_1, Q \cup \{Y_2, Y_3, \ldots, Y_h\}\rangle y_1\\ &\langle Y_2, Q \cup \{Y_3, \ldots, Y_h\}\rangle y_2\\ &\vdots\\ &\langle Y_h, Q\rangle y_h\end{aligned}$$

(2) For each $(A \rightarrow y_0Y_1y_1Y_2y_2 \cdots Y_hy_h, \emptyset, W) \in P_L$, where $y_i \in T^*$, $Y_j \in N$, for all i and j, $0 \leq i \leq h$, $1 \leq j \leq h$, for some $h \geq 0$, and for each $\langle A, Q\rangle \in N'$, extend P by adding

$$\begin{aligned}\langle A, Q\rangle \to y_0 &\langle Y_1, Q \cup \{Y_2, Y_3, \ldots, Y_h\}\rangle y_1 \\ &\langle Y_2, Q \cup \{Y_3, \ldots, Y_h\}\rangle y_2 \\ &\vdots \\ &\langle Y_h, Q\rangle y_h\end{aligned}$$

Before proving that $L(H) = L(G, {}_{\rm lm}{\overset{1}{\Rightarrow}})$, let us give an insight into the construction. As G always rewrites the leftmost occurrence of a nonterminal, we use compound nonterminals of the form $\langle A, Q\rangle$ in H, where A is a nonterminal and Q is a set of nonterminals that appear to the right of this occurrence of A. When simulating rules from P_R, the check for the presence and absence of symbols is accomplished by using Q. Also, when rewriting A in $\langle A, Q\rangle$ to some y, the compound nonterminals from N' are generated instead of nonterminals from N.

Rules from P_L are simulated analogously; however, notice that if the permitting set of such a rule is nonempty, it is never applicable in G. Therefore, such rules are not introduced to P'. Furthermore, since there are no nonterminals to the left of the leftmost occurrence of a nonterminal, no check for their absence is done.

Clearly, $L(G, {}_{\rm lm}{\overset{1}{\Rightarrow}}) \subseteq L(H)$. The opposite inclusion, $L(H) \subseteq L(G, {}_{\rm lm}{\overset{1}{\Rightarrow}})$, can be proved by analogy with the proof of Lemma 7.1.2 by simulating the leftmost derivation of every $w \in L(H)$ by G. Observe that since the check for the presence and absence of symbols in H is done in the second components of the compound nonterminals, each rule introduced to P in (1) and (2) can be simulated by a rule from P_R and P_L from which it is constructed.

Since H is propagating whenever G is propagating, the theorem holds. □

Theorem 7.1.4. $\mathbf{ORC}({}_{\rm lm}{\overset{1}{\Rightarrow}}) = \mathbf{CF}$

Proof. By Lemma 7.1.2, $\mathbf{CF} \subseteq \mathbf{ORC}({}_{\rm lm}{\overset{1}{\Rightarrow}})$. By Lemma 7.1.3, $\mathbf{ORC}({}_{\rm lm}{\overset{1}{\Rightarrow}}) \subseteq \mathbf{CF}$. Consequently, $\mathbf{ORC}({}_{\rm lm}{\overset{1}{\Rightarrow}}) = \mathbf{CF}$, so the theorem holds. □

Theorem 7.1.5. $\mathbf{ORC}^{-\varepsilon}({}_{\rm lm}{\overset{1}{\Rightarrow}}) = \mathbf{CF}$

Proof. Since any context-free grammar can be converted to an equivalent context-free grammar without any erasing rules (see Theorem 7.9 in [37]), this theorem follows from Lemmas 7.1.2 and 7.1.3. □

7.2 Type-2 Leftmost Derivations

In the second derivation restriction type, during every derivation step, the leftmost occurrence of a nonterminal that can be rewritten has to be rewritten.

Definition 7.2.1. Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. The *type*-2 *direct leftmost derivation relation* over V^*, symbolically denoted by ${}_{\rm lm}{\overset{2}{\Rightarrow}}_G$, is defined as follows. Let $u, x, v \in V^*$ and $A \in N$. Then,

$$uAv \ {}_{\mathrm{lm}}\!\overset{2}{\Rightarrow}_G \ uxv$$

if and only if $uAv \Rightarrow_G uxv$ and there is no $B \in N$ and $y \in V^*$ such that $u = u_1 B u_2$ and $u_1 B u_2 A v \Rightarrow_G u_1 y u_2 A v$.

Let ${}_{\mathrm{lm}}\!\overset{2}{\Rightarrow}{}^n_G$ and ${}_{\mathrm{lm}}\!\overset{2}{\Rightarrow}{}^*_G$ denote the nth power of ${}_{\mathrm{lm}}\!\overset{2}{\Rightarrow}_G$, for some $n \geq 0$, and the reflexive-transitive closure of ${}_{\mathrm{lm}}\!\overset{2}{\Rightarrow}_G$, respectively. The ${}_{\mathrm{lm}}\!\overset{2}{}$*-language* of G is denoted by $L(G, {}_{\mathrm{lm}}\!\overset{2}{\Rightarrow})$ and defined as

$$L\big(G, {}_{\mathrm{lm}}\!\overset{2}{\Rightarrow}\big) = \big\{ w \in T^* \mid S \ {}_{\mathrm{lm}}\!\overset{2}{\Rightarrow}{}^*_G \ w \big\} \qquad \square$$

The language families generated by one-sided random context grammars with type-2 leftmost derivations and propagating one-sided random context grammars with type-2 leftmost derivations are denoted by $\mathbf{ORC}({}_{\mathrm{lm}}\!\overset{2}{\Rightarrow})$ and $\mathbf{ORC}^{-\varepsilon}({}_{\mathrm{lm}}\!\overset{2}{\Rightarrow})$, respectively.

Next, we prove that $\mathbf{ORC}({}_{\mathrm{lm}}\!\overset{2}{\Rightarrow}) = \mathbf{RE}$ and $\mathbf{ORC}^{-\varepsilon}({}_{\mathrm{lm}}\!\overset{2}{\Rightarrow}) = \mathbf{CS}$.

Lemma 7.2.2. *For every one-sided random context grammar G, there is a one-sided random context grammar H such that $L(H, {}_{\mathrm{lm}}\!\overset{2}{\Rightarrow}) = L(G)$. Furthermore, if G is propagating, then so is H.*

Proof. Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. We construct the one-sided random context grammar H in such a way that always allows it to rewrite an arbitrary occurrence of a nonterminal. Construct

$$H = \big(N', T, P'_L, P'_R, S\big)$$

as follows. Initially, set $\bar{N} = \{\bar{A} \mid A \in N\}$, $\hat{N} = \{\hat{A} \mid A \in N\}$, $N' = N \cup \bar{N} \cup \hat{N}$, and $P'_L = P'_R = \emptyset$. Without any loss of generality, assume that N, $\bar{N}$, and $\hat{N}$ are pairwise disjoint. Define the function ψ from 2^N to $2^{\bar{N}}$ as $\psi(\emptyset) = \emptyset$ and

$$\psi\big(\{A_1, A_2, \dots, A_n\}\big) = \big\{\bar{A}_1, \bar{A}_2, \dots, \bar{A}_n\big\}$$

Perform (1) through (3), given next.

(1) For each $A \in N$,

(1.1) add $(A \to \bar{A}, \emptyset, N \cup \hat{N})$ to P'_L,
(1.2) add $(\bar{A} \to \hat{A}, \emptyset, N \cup \bar{N})$ to P'_R,
(1.3) add $(\hat{A} \to A, \emptyset, \bar{N} \cup \hat{N})$ to P'_R.

(2) For each $(A \to y, U, W) \in P_R$, add $(A \to y, U, W)$ to P'_R.
(3) For each $(A \to y, U, W) \in P_L$, add $(A \to y, \psi(U), \psi(W) \cup N \cup \hat{N})$ to P'_L.

Before proving that $L(H) = L(G)$, let us informally explain (1) through (3). Rules from (2) and (3) simulate the corresponding rules from P_R and P_L, respectively. Rules from (1) allow H to rewrite any occurrence of a nonterminal.

Set $V = N \cup T$. Consider a sentential form x_1Ax_2, where $x_1, x_2 \in V^*$ and $A \in N$. To rewrite A in H using type-2 leftmost derivations, all occurrences of nonterminals in x_1 are first rewritten to their barred versions by rules from (1.1). Then, A can be rewritten by a rule from (2) or (3). By rules from (1.1), every occurrence of a nonterminal in the current sentential form is then rewritten to its barred version. Rules from (1.2) then start rewriting barred nonterminals to hatted nonterminals, which is performed from the right to the left. Finally, hatted nonterminals are rewritten to their original versions by rules from (1.3). This is also performed from the right to the left.

To establish the identity $L(H, {}_{\mathrm{lm}}\!\overset{2}{\Rightarrow}) = L(G)$, we prove two claims. First, Claim 1 shows how derivations of G are simulated by H. Then, Claim 2 demonstrates the converse—that is, it shows how derivations of H are simulated by G.

Claim 1. If $S \Rightarrow_G^n x$, where $x \in V^*$, for some $n \geq 0$, then $S\ {}_{\mathrm{lm}}\!\overset{2}{\Rightarrow}{}_H^* x$.

Proof. This claim is established by induction on $n \geq 0$.

Basis. For $n = 0$, this claim obviously holds.

Induction Hypothesis. Suppose that there exists $n \geq 0$ such that the claim holds for all derivations of length ℓ, where $0 \leq \ell \leq n$.

Induction Step. Consider any derivation of the form

$$S \Rightarrow_G^{n+1} w$$

where $w \in V^*$. Since $n + 1 \geq 1$, this derivation can be expressed as

$$S \Rightarrow_G^n x \Rightarrow_G w$$

for some $x \in V^+$. By the induction hypothesis, $S\ {}_{\mathrm{lm}}\!\overset{2}{\Rightarrow}{}_H^* x$. Next, we consider all possible forms of $x \Rightarrow_G w$, covered by the following two cases—(i) and (ii).

(i) *Application of $(A \to y, U, W) \in P_R$.* Let $x = x_1Ax_2$ and $r = (A \to y, U, W) \in P_R$, where $x_1, x_2 \in V^*$ such that $U \subseteq \mathrm{alph}(x_2)$ and $W \cap \mathrm{alph}(x_2) = \emptyset$, so

$$x_1Ax_2 \Rightarrow_G x_1yx_2$$

If $x_1 \in T^*$, then $x_1Ax_2\ {}_{\mathrm{lm}}\!\overset{2}{\Rightarrow}{}_H\ x_1yx_2$ by the corresponding rule introduced in (2), and the induction step is completed for (i). Therefore, assume that $\mathrm{alph}(x_1) \cap N \neq \emptyset$. Let $x_1 = z_0Z_1z_1Z_2z_2 \cdots Z_hz_h$, where $z_i \in T^*$ and $Z_j \in N$, for all i and j, $0 \leq i \leq h$, $1 \leq j \leq h$, for some $h \geq 1$. By rules introduced in (1.1),

$$z_0Z_1z_1Z_2z_2 \cdots Z_hz_hAx_2\ {}_{\mathrm{lm}}\!\overset{2}{\Rightarrow}{}_H^*\ z_0\bar{Z}_1z_1\bar{Z}_2z_2 \cdots \bar{Z}_hz_hAx_2$$

By the corresponding rule to r introduced in (2),

$$z_0\bar{Z}_1z_1\bar{Z}_2z_2\cdots\bar{Z}_hz_hAx_2 \ {}_{\mathrm{lm}}^{\ 2}\!\!\Rightarrow_H z_0\bar{Z}_1z_1\bar{Z}_2z_2\cdots\bar{Z}_hz_hyx_2$$

By rules introduced in (1.1) through (1.3),

$$z_0\bar{Z}_1z_1\bar{Z}_2z_2\cdots\bar{Z}_hz_hyx_2 \ {}_{\mathrm{lm}}^{\ 2}\!\!\Rightarrow_H^* z_0Z_1z_1Z_2z_2\cdots Z_hz_hyx_2$$

which completes the induction step for (i).

(ii) *Application of* $(A \to y, U, W) \in P_L$. Let $x = x_1Ax_2$ and $r = (A \to y, U, W) \in P_L$, where $x_1, x_2 \in V^*$ such that $U \subseteq \mathrm{alph}(x_1)$ and $W \cap \mathrm{alph}(x_1) = \emptyset$, so

$$x_1Ax_2 \Rightarrow_G x_1yx_2$$

To complete the induction step for (ii), proceed by analogy with (i), but use a rule from (3) instead of a rule from (2).

Observe that cases (i) and (ii) cover all possible forms of $x \Rightarrow_G w$. Thus, the claim holds. □

Set $V = N \cup T$ and $V' = N' \cup T$. Define the homomorphism τ from V'^* to V^* as $\tau(A) = \tau(\bar{A}) = \tau(\hat{A}) = A$, for all $A \in N$, and $\tau(a) = a$, for all $a \in T$.

Claim 2. If $S \ {}_{\mathrm{lm}}^{\ 2}\!\!\Rightarrow_H^n x$, *where* $x \in V'^*$, *for some* $n \geq 0$, *then* $S \Rightarrow_G^* \tau(x)$, *and either* $x \in (\bar{N} \cup T)^*V^*$, $x \in (\bar{N} \cup T)^*(\hat{N} \cup T)^*$, *or* $x \in (\hat{N} \cup T)^*V^*$.

Proof. This claim is established by induction on $n \geq 0$.

Basis. For $n = 0$, this claim obviously holds.

Induction Hypothesis. Suppose that there exists $n \geq 0$ such that the claim holds for all derivations of length ℓ, where $0 \leq \ell \leq n$.

Induction Step. Consider any derivation of the form

$$S \ {}_{\mathrm{lm}}^{\ 2}\!\!\Rightarrow_H^{n+1} w$$

where $w \in V'^*$. Since $n + 1 \geq 1$, this derivation can be expressed as

$$S \ {}_{\mathrm{lm}}^{\ 2}\!\!\Rightarrow_H^n x \ {}_{\mathrm{lm}}^{\ 2}\!\!\Rightarrow_H w$$

for some $x \in V'^+$. By the induction hypothesis, $S \Rightarrow_G^* \tau(x)$, and either $x \in (\bar{N} \cup T)^*V^*$, $x \in (\bar{N} \cup T)^*(\hat{N} \cup T)^*$, or $x \in (\hat{N} \cup T)^*V^*$. Next, we consider all possible forms of $x \ {}_{\mathrm{lm}}^{\ 2}\!\!\Rightarrow_H w$, covered by the following five cases—(i) through (v).

(i) *Application of a rule introduced in* (1.1). Let $(A \to \bar{A}, \emptyset, N \cup \hat{N}) \in P'_L$ be a rule introduced in (1.1). Observe that this rule is applicable only if $x = x_1Ax_2$, where $x_1 \in (\bar{N} \cup T)^*$ and $x_2 \in V^*$. Then,

$$x_1Ax_2 \ {}_{\mathrm{lm}}^{\ 2}\!\!\Rightarrow_H x_1\bar{A}x_2$$

Since $\tau(x_1\bar{A}x_2) = \tau(x_1Ax_2)$ and $x_1\bar{A}x_2 \in (\bar{N} \cup T)^*V^*$, the induction step is completed for (i).

(ii) *Application of a rule introduced in* (1.2). Let $(\bar{A} \to \hat{A}, \emptyset, N \cup \bar{N}) \in P'_R$ be a rule introduced in (1.2). Observe that this rule is applicable only if $x = x_1\bar{A}x_2$, where $x_1 \in (\bar{N} \cup T)^*$ and $x_2 \in (\hat{N} \cup T)^*$. Then,

$$x_1\bar{A}x_2 \;{}_{\rm lm}\!\overset{2}{\Rightarrow}_H\; x_1\hat{A}x_2$$

Since $\tau(x_1\hat{A}x_2) = \tau(x_1\bar{A}x_2)$ and $x_1\hat{A}x_2 \in (\bar{N} \cup T)^*(\hat{N} \cup T)^*$, the induction step is completed for (ii).

(iii) *Application of a rule introduced in* (1.3). Let $(\hat{A} \to A, \emptyset, \bar{N} \cup \hat{N}) \in P'_R$ be a rule introduced in (1.3). Observe that this rule is applicable only if $x = x_1\hat{A}x_2$, where $x_1 \in (\hat{N} \cup T)^*$ and $x_2 \in V^*$. Then,

$$x_1\hat{A}x_2 \;{}_{\rm lm}\!\overset{2}{\Rightarrow}_H\; x_1Ax_2$$

Since $\tau(x_1Ax_2) = \tau(x_1\hat{A}x_2)$ and $x_1Ax_2 \in (\hat{N} \cup T)^*V^*$, the induction step is completed for (iii).

(iv) *Application of a rule introduced in* (2). Let $(A \to y, U, W) \in P'_R$ be a rule introduced in (2) from $(A \to y, U, W) \in P_R$, and let $x = x_1Ax_2$ such that $U \subseteq \mathrm{alph}(x_2)$ and $W \cap \mathrm{alph}(x_2) = \emptyset$. Then,

$$x_1Ax_2 \;{}_{\rm lm}\!\overset{2}{\Rightarrow}_H\; x_1yx_2$$

and

$$\tau(x_1)A\tau(x_2) \Rightarrow_G \tau(x_1)y\tau(x_2)$$

Clearly, x_1yx_2 is of the required form, so the induction step is completed for (iv).

(v) *Application of a rule introduced in* (3). Let $(A \to y, \psi(U), \psi(W) \cup N \cup \hat{N}) \in P'_L$ be a rule introduced in (3) from $(A \to y, U, W) \in P_L$, and let $x = x_1Ax_2$ such that $\psi(U) \subseteq \mathrm{alph}(x_1)$ and $(\psi(W) \cup N \cup \hat{N}) \cap \mathrm{alph}(x_1) = \emptyset$. Then,

$$x_1Ax_2 \;{}_{\rm lm}\!\overset{2}{\Rightarrow}_H\; x_1yx_2$$

and

$$\tau(x_1)A\tau(x_2) \Rightarrow_G \tau(x_1)y\tau(x_2)$$

Clearly, x_1yx_2 is of the required form, so the induction step is completed for (v).

Observe that cases (i) through (v) cover all possible forms of $x \;{}_{\rm lm}\!\overset{2}{\Rightarrow}_H\; w$. Thus, the claim holds. □

We next prove that $L(H, {}_{\rm lm}\!\overset{2}{\Rightarrow}) = L(G)$. Consider Claim 1 for $x \in T^*$. Then, $S \Rightarrow_G^* x$ implies that $S \;{}_{\rm lm}\!\overset{2}{\Rightarrow}{}_H^*\; x$, so $L(G) \subseteq L(H, {}_{\rm lm}\!\overset{2}{\Rightarrow})$. Consider Claim 2 for $x \in T^*$. Then, $S \;{}_{\rm lm}\!\overset{2}{\Rightarrow}{}_H^*\; x$ implies that $S \Rightarrow_G^* x$, so $L(H, {}_{\rm lm}\!\overset{2}{\Rightarrow}) \subseteq L(G)$. Consequently, $L(H, {}_{\rm lm}\!\overset{2}{\Rightarrow}) = L(G)$.

Since H is propagating whenever G is propagating, the theorem holds. □

Lemma 7.2.3. $\mathbf{ORC}({}_{\text{lm}}{}^{2}\!\!\Rightarrow) \subseteq \mathbf{RE}$

Proof. This inclusion follows from Church's thesis. □

Theorem 7.2.4. $\mathbf{ORC}({}_{\text{lm}}{}^{2}\!\!\Rightarrow) = \mathbf{RE}$

Proof. Since $\mathbf{ORC} = \mathbf{RE}$ (see Theorem 4.1.4), Lemma 7.2.2 implies that $\mathbf{RE} \subseteq \mathbf{ORC}({}_{\text{lm}}{}^{2}\!\!\Rightarrow)$. By Lemma 7.2.3, $\mathbf{ORC}({}_{\text{lm}}{}^{2}\!\!\Rightarrow) \subseteq \mathbf{RE}$. Consequently, $\mathbf{ORC}({}_{\text{lm}}{}^{2}\!\!\Rightarrow) = \mathbf{RE}$, so the theorem holds. □

Lemma 7.2.5. $\mathbf{ORC}^{-\varepsilon}({}_{\text{lm}}{}^{2}\!\!\Rightarrow) \subseteq \mathbf{CS}$

Proof. Since the length of sentential forms in derivations of propagating one-sided random context grammars is nondecreasing, propagating one-sided random context grammars can be simulated by context-sensitive grammars. A rigorous proof of this lemma is left to the reader. □

Theorem 7.2.6. $\mathbf{ORC}^{-\varepsilon}({}_{\text{lm}}{}^{2}\!\!\Rightarrow) = \mathbf{CS}$

Proof. Since $\mathbf{ORC}^{-\varepsilon} = \mathbf{CS}$ (see Theorem 4.1.3), Lemma 7.2.2 implies that $\mathbf{CS} \subseteq \mathbf{ORC}^{-\varepsilon}({}_{\text{lm}}{}^{2}\!\!\Rightarrow)$. By Lemma 7.2.5, $\mathbf{ORC}^{-\varepsilon}({}_{\text{lm}}{}^{2}\!\!\Rightarrow) \subseteq \mathbf{CS}$. Consequently, we have $\mathbf{ORC}^{-\varepsilon}({}_{\text{lm}}{}^{2}\!\!\Rightarrow) = \mathbf{CS}$, so the theorem holds. □

7.3 Type-3 Leftmost Derivations

In the third derivation restriction type, during every derivation step, a rule is chosen, and the leftmost occurrence of its left-hand side is rewritten.

Definition 7.3.1. Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. The *type*-3 *direct leftmost derivation relation* over V^*, symbolically denoted by ${}_{\text{lm}}{}^{3}\!\!\Rightarrow_G$, is defined as follows. Let $u, x, v \in V^*$ and $A \in N$. Then,

$$uAv \;{}_{\text{lm}}{}^{3}\!\!\Rightarrow_G uxv$$

if and only if $uAv \Rightarrow_G uxv$ and $\text{alph}(u) \cap \{A\} = \emptyset$.

Let ${}_{\text{lm}}{}^{3}\!\!\Rightarrow^n_G$ and ${}_{\text{lm}}{}^{3}\!\!\Rightarrow^*_G$ denote the nth power of ${}_{\text{lm}}{}^{3}\!\!\Rightarrow_G$, for some $n \geq 0$, and the reflexive-transitive closure of ${}_{\text{lm}}{}^{3}\!\!\Rightarrow_G$, respectively. The ${}_{\text{lm}}{}^{3}$-*language* of G is denoted by $L(G, {}_{\text{lm}}{}^{3}\!\!\Rightarrow)$ and defined as

$$L\big(G, {}_{\text{lm}}{}^{3}\!\!\Rightarrow\big) = \big\{w \in T^* \mid S \;{}_{\text{lm}}{}^{3}\!\!\Rightarrow^*_G w\big\}$$ □

Notice the following difference between the second and the third type. In the former, the leftmost occurrence of a rewritable nonterminal is chosen first, and then, a choice of a rule with this nonterminal on its let-hand side is made. In the latter, a rule is chosen first, and then, the leftmost occurrence of its left-hand side is rewritten.

The language families generated by one-sided random context grammars with type-3 leftmost derivations and propagating one-sided random context grammars with type-3 leftmost derivations are denoted by $\mathbf{ORC}({}_{\mathrm{lm}}\!\overset{3}{\Rightarrow})$ and $\mathbf{ORC}^{-\varepsilon}({}_{\mathrm{lm}}\!\overset{3}{\Rightarrow})$, respectively.

Next, we prove that $\mathbf{ORC}({}_{\mathrm{lm}}\!\overset{3}{\Rightarrow}) = \mathbf{RE}$ and $\mathbf{ORC}^{-\varepsilon}({}_{\mathrm{lm}}\!\overset{3}{\Rightarrow}) = \mathbf{CS}$.

Lemma 7.3.2. *For every one-sided random context grammar G, there is a one-sided random context grammar H such that $L(H, {}_{\mathrm{lm}}\!\overset{3}{\Rightarrow}) = L(G)$. Furthermore, if G is propagating, then so is H.*

Proof. Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. We prove this lemma by analogy with the proof of Lemma 7.2.2. That is, we construct the one-sided random context grammar H in such a way that always allows it to rewrite an arbitrary occurrence of a nonterminal. Construct

$$H = \big(N', T, P'_L, P'_R, S\big)$$

as follows. Initially, set $\bar{N} = \{\bar{A} \mid A \in N\}$, $N' = N \cup \bar{N}$, and $P'_L = P'_R = \emptyset$. Without any loss of generality, assume that $N \cap \bar{N} = \emptyset$. Define the function ψ from 2^N to $2^{\bar{N}}$ as $\psi(\emptyset) = \emptyset$ and

$$\psi\big(\{A_1, A_2, \dots, A_n\}\big) = \big\{\bar{A}_1, \bar{A}_2, \dots, \bar{A}_n\big\}$$

Perform (1) through (3), given next.

(1) For each $A \in N$,

(1.1) add $(A \to \bar{A}, \emptyset, N)$ to P'_L;
(1.2) add $(\bar{A} \to A, \emptyset, \bar{N})$ to P'_R.

(2) For each $(A \to y, U, W) \in P_R$, add $(A \to y, U, W)$ to P'_R.
(3) For each $(A \to y, U, W) \in P_L$, let $U = \{X_1, X_2, \dots, X_k\}$, and for each

$$U' \in \big\{\{Y_1, Y_2, \dots, Y_k\} \mid Y_i \in \{X_i, \bar{X}_i\}, 1 \le i \le k\big\}$$

add $(A \to y, U', W \cup \Psi(W))$ to P'_L ($U' = \emptyset$ if and only if $U = \emptyset$).

Before proving that $L(G) = L(H, {}_{\mathrm{lm}}\!\overset{3}{\Rightarrow})$, let us give an insight into the construction. Rules introduced in (1) allow H to rewrite an arbitrary occurrence of a nonterminal. Rules from (2) and (3) simulate the corresponding rules from P_R and P_L, respectively.

Consider a sentential form $x_1 A x_2$, where $x_1, x_2 \in (N \cup T)^*$ and $A \in N$, and a rule, $r = (A \to y, U, W) \in P'_L \cup P'_R$, introduced in (2) or (3). If $A \in \mathrm{alph}(x_1)$, all occurrences of nonterminals in x_1 are rewritten to their barred versions by rules from (1). Then, r is applied, and all barred nonterminals are rewritten back to their non-barred versions. Since not all occurrences of nonterminals in x_1 need to be rewritten to their barred versions before r is applied, all combinations of barred and non-barred nonterminals in the left permitting contexts of the resulting rules in (3) are considered.

The identity $L(H, {}_{\mathrm{lm}}\!\overset{3}{\Rightarrow}) = L(G)$ can be established by analogy with the proof given in Lemma 7.2.2, and we leave its proof to the reader. Finally, notice that whenever G is propagating, then so is H. Hence, the theorem holds. □

Lemma 7.3.3. $\mathbf{ORC}({}_{\mathrm{lm}}\!\overset{3}{\Rightarrow}) \subseteq \mathbf{RE}$

Proof. This inclusion follows from Church's thesis. □

Theorem 7.3.4. $\mathbf{ORC}({}_{\mathrm{lm}}\!\overset{3}{\Rightarrow}) = \mathbf{RE}$

Proof. Since $\mathbf{ORC} = \mathbf{RE}$ (see Theorem 4.1.4), Lemma 7.3.2 implies that $\mathbf{RE} \subseteq \mathbf{ORC}({}_{\mathrm{lm}}\!\overset{3}{\Rightarrow})$. By Lemma 7.3.3, $\mathbf{ORC}({}_{\mathrm{lm}}\!\overset{3}{\Rightarrow}) \subseteq \mathbf{RE}$. Consequently, $\mathbf{ORC}({}_{\mathrm{lm}}\!\overset{3}{\Rightarrow}) = \mathbf{RE}$, so the theorem holds. □

Lemma 7.3.5. $\mathbf{ORC}^{-\varepsilon}({}_{\mathrm{lm}}\!\overset{3}{\Rightarrow}) \subseteq \mathbf{CS}$

Proof. This lemma can be proved by analogy with proving Lemma 7.2.5. □

Theorem 7.3.6. $\mathbf{ORC}^{-\varepsilon}({}_{\mathrm{lm}}\!\overset{3}{\Rightarrow}) = \mathbf{CS}$

Proof. Since $\mathbf{ORC}^{-\varepsilon} = \mathbf{CS}$ (see Theorem 4.1.3), Lemma 7.3.2 implies that $\mathbf{CS} \subseteq \mathbf{ORC}^{-\varepsilon}({}_{\mathrm{lm}}\!\overset{3}{\Rightarrow})$. By Lemma 7.3.5, $\mathbf{ORC}^{-\varepsilon}({}_{\mathrm{lm}}\!\overset{3}{\Rightarrow}) \subseteq \mathbf{CS}$. Consequently, we have $\mathbf{ORC}^{-\varepsilon}({}_{\mathrm{lm}}\!\overset{3}{\Rightarrow}) = \mathbf{CS}$, so the theorem holds. □

In the conclusion of this chapter, we compare the achieved results with some well-known results of formal language theory. More specifically, we relate the language families generated by one-sided random context grammars with leftmost derivations to the language families generated by random context grammars with leftmost derivations.

The families of languages generated by random context grammars with type-1 leftmost derivations, random context grammars with type-2 leftmost derivations, and random context grammars with type-3 leftmost derivations are denoted by $\mathbf{RC}({}_{\mathrm{lm}}\!\overset{1}{\Rightarrow})$, $\mathbf{RC}({}_{\mathrm{lm}}\!\overset{2}{\Rightarrow})$, and $\mathbf{RC}({}_{\mathrm{lm}}\!\overset{3}{\Rightarrow})$, respectively (see [16] for the definitions of all these families). The notation without ε stands for the corresponding propagating family. For example, $\mathbf{RC}^{-\varepsilon}({}_{\mathrm{lm}}\!\overset{1}{\Rightarrow})$ denotes the language family generated by propagating random context grammars with type-1 leftmost derivations.

The fundamental relations between random context grammars and one-sided random context grammars without leftmost derivations are summarized next.

Corollary 7.3.7. $\mathbf{CF} \subset \mathbf{RC}^{-\varepsilon} \subset \mathbf{ORC}^{-\varepsilon} = \mathbf{CS} \subset \mathbf{ORC} = \mathbf{RC} = \mathbf{RE}$

Proof. This corollary follows from Theorems 2.3.14, 4.1.3, and 4.1.4. □

Considering type-1 leftmost derivations, we significantly decrease the power of both one-sided random context grammars and random context grammars.

Corollary 7.3.8. $\mathbf{ORC}({}_{\mathrm{lm}}{\overset{1}{\Rightarrow}}) = \mathbf{RC}({}_{\mathrm{lm}}{\overset{1}{\Rightarrow}}) = \mathbf{CF}$

Proof. This corollary follows from Theorem 7.1.4 in this chapter and from Theorem 1.4.1 in [16]. □

Type-2 leftmost derivations increase the generative power of propagating random context grammars, but the generative power of random context grammars and one-sided random context grammars remains unchanged.

Corollary 7.3.9.

(i) $\mathbf{ORC}^{-\varepsilon}({}_{\mathrm{lm}}{\overset{2}{\Rightarrow}}) = \mathbf{RC}^{-\varepsilon}({}_{\mathrm{lm}}{\overset{2}{\Rightarrow}}) = \mathbf{CS}$
(ii) $\mathbf{ORC}({}_{\mathrm{lm}}{\overset{2}{\Rightarrow}}) = \mathbf{RC}({}_{\mathrm{lm}}{\overset{2}{\Rightarrow}}) = \mathbf{RE}$

Proof. This corollary follows from Theorems 7.2.4 and 7.2.6 in this chapter and from Theorem 1.4.4 in [16]. □

Finally, type-3 leftmost derivations are not enough for propagating random context grammars to generate the family of context-sensitive languages, so one-sided random context grammars with type-3 leftmost derivations are more powerful.

Corollary 7.3.10.

(i) $\mathbf{RC}^{-\varepsilon}({}_{\mathrm{lm}}{\overset{3}{\Rightarrow}}) \subset \mathbf{ORC}^{-\varepsilon}({}_{\mathrm{lm}}{\overset{3}{\Rightarrow}}) = \mathbf{CS}$
(ii) $\mathbf{ORC}({}_{\mathrm{lm}}{\overset{3}{\Rightarrow}}) = \mathbf{RC}({}_{\mathrm{lm}}{\overset{3}{\Rightarrow}}) = \mathbf{RE}$

Proof. This corollary follows from Theorems 7.3.4 and 7.3.6 in the this chapter, from Theorem 1.4.5 in [16], and from Remarks 5.11 in [22]. □

We close this chapter by making a remark about rightmost derivations. Of course, we can define and study rightmost derivations in one-sided random context grammars by analogy with their leftmost counterparts, discussed above. We can also reformulate and establish the same results as above in terms of the rightmost derivations. All this discussion of rightmost derivations is so analogous with the above discussion of leftmost derivations that we leave it to the reader.

Chapter 8
Generalized One-Sided Forbidding Grammars

In [66], so-called *generalized forbidding grammars* that are based upon context-free rules, each of which may be associated with finitely many *forbidding strings*, were introduced and investigated. A rule like this can rewrite a nonterminal provided that none of its forbidding strings occur in the current sentential form; apart from this, these grammars work just like context-free grammars. As opposed to context-free grammars, however, they are computationally complete—that is, they generate the family of recursively enumerable languages (see Theorem 1 in [66]), and this property obviously represents their crucially important advantage over ordinary context-free and forbidding grammars (see Theorem 2.3.15).

Taking a closer look at the rewriting process in generalized forbidding grammars, we see that they always verify the absence of forbidding strings within their entire sentential forms. To simplify and accelerate their rewriting process, it is obviously more than desirable to modify these grammars so they make this verification only within some prescribed portions of the rewritten sentential forms while remaining computationally complete. *Generalized one-sided forbidding grammars*, which are defined and studied in the present chapter, represent a modification satisfying these properties.

More precisely, in a generalized one-sided forbidding grammar, the set of rules is divided into the set of *left forbidding rules* and the set of *right forbidding rules*. When applying a left forbidding rule, the grammar checks the absence of its forbidding strings only in the prefix to the left of the rewritten nonterminal in the current sentential form. Similarly, when applying a right forbidding rule, it performs an analogous check to the right. Apart from this, it works like any generalized forbidding grammar.

Most importantly, the present chapter demonstrates that generalized one-sided forbidding grammars characterize the family of recursively enumerable languages. In fact, these grammars remain computationally complete even under the restriction that any of their forbidding strings is of length two or less. On the other hand, if a generalized one-sided forbidding grammar has all left forbidding rules without any forbidding strings, then it necessarily generates a context-free language; an analogous result holds in terms of right forbidding rules, too. Even more surprisingly, any generalized one-sided forbidding grammar that has the set of left forbidding rules coinciding with the set of right forbidding rules generates a context-free language.

This chapter is divided into two sections. First, Section 8.1 defines generalized one-sided forbidding grammars and illustrate them by an example. Then, Section 8.2 establishes their generative power.

8.1 Definitions and Examples

Without further ado, let us define generalized one-sided forbidding grammars and illustrate them by an example. Recall that for an alphabet N and a string $x \in N^*$, $\mathrm{sub}(x)$ denotes the set of all substrings of x, and $\mathrm{fin}(N)$ denotes the set of all finite languages over N (see Section 2.2).

Definition 8.1.1. A *generalized one-sided forbidding 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

$$P_L, P_R \subseteq N \times \big(N \cup T\big)^* \times \mathrm{fin}(N)$$

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 forbidding rules*, the set of *right forbidding rules*, and the *start symbol*, respectively. Each $(A, x, F) \in P_L \cup P_R$ is written as $(A \rightarrow x, F)$ throughout this chapter. For $(A \rightarrow x,\ F) \in P_L$, F is called the *left forbidding context*. Analogously, for $(A \rightarrow x,\ F) \in P_R$, F is called the *right forbidding context*. The *direct derivation relation* over V^*, symbolically denoted by $\Rightarrow_G$, is defined as follows. Let $u, v \in V^*$ and $(A \rightarrow x,\ F) \in P_L \cup P_R$. Then,

$$uAv \Rightarrow_G uxv$$

if and only if

$$(A \rightarrow x, F) \in P_L \text{ and } F \cap \mathrm{sub}(u) = \emptyset$$

or

$$(A \rightarrow x, F) \in P_R \text{ and } F \cap \mathrm{sub}(v) = \emptyset$$

Let $\Rightarrow_G^n$ and $\Rightarrow_G^*$ denote the nth power of $\Rightarrow_G$, for some $n \geq 0$, and the reflexive-transitive closure of $\Rightarrow_G$, respectively. The *language* of G is denoted by $L(G)$ and defined as

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

□

Next, we introduce the notion of a degree of G. Informally, it is the length of the longest string in the forbidding contexts of the rules of G. Let N be an alphabet. For $L \in \text{fin}(N)$, max-len(L) denotes the length of the longest string in L. We set max-len$(\emptyset) = 0$.

Definition 8.1.2. Let $G = (N, T, P_L, P_R, S)$ be a generalized one-sided forbidding grammar. G is of *degree* (m,n), where $m,n \geq 0$, if $(A \to x, F) \in P_L$ implies that max-len$(F) \leq m$ and $(A \to x, F) \in P_R$ implies that max-len$(F) \leq n$. □

Next, we illustrate the previous definitions by an example.

Example 8.1.3. Consider the generalized 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

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

and P_R contains the following four rules

$$\begin{array}{ll} (A \to aA', \{B'\}) & (A' \to A, \{B\}) \\ (A \to \bar{A}, \{B'\}) & (\bar{A} \to \varepsilon, \{B\}) \end{array}$$

Since the length of the longest string in the forbidding contexts of rules from P_L and P_R is 1, G is of degree $(1,1)$. It is rather easy to see that every derivation that generates a nonempty string of $L(G)$ is of the form

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

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

$$\big\{a^nb^nc^n \mid n \geq 0\big\}$$ □

The language family generated by generalized one-sided forbidding grammars of degree (m,n) is denoted by $\mathbf{GOF}(m,n)$. Furthermore, set

$$\mathbf{GOF} = \bigcup_{m,n\geq 0} \mathbf{GOF}(m,n)$$

8.2 Generative Power

In this section, we establish the generative power of generalized one-sided forbidding grammars. More specifically, we prove results (I) through (IV), given next.

(I) Generalized one-sided forbidding grammars of degrees $(n,0)$ or $(0,n)$, for any non-negative integer n, characterize only the family of context-free languages (Theorem 8.2.3).
(II) Generalized one-sided forbidding grammars of degree $(1,1)$ generate a proper superfamily of the family of context-free languages (Theorem 8.2.4).
(III) Generalized one-sided forbidding grammars of degrees $(1,2)$ or $(2,1)$ characterize the family of recursively enumerable languages (Theorem 8.2.7).
(IV) Generalized one-sided forbidding grammars with the set of left forbidding rules coinciding with the set of right forbidding rules characterize only the family of context-free languages (Theorem 8.2.13).

First, we consider generalized one-sided forbidding grammars of degrees $(n,0)$ and $(0,n)$, where $n \geq 0$.

Lemma 8.2.1. $\mathbf{GOF}(n,0) = \mathbf{CF}$ *for every* $n \geq 0$.

Proof. Let n be a non-negative integer. As any context-free grammar is also a generalized one-sided forbidding grammar in which the empty sets are attached to each of its rules, the inclusion $\mathbf{CF} \subseteq \mathbf{GOF}(n,0)$ holds. To establish the other inclusion, $\mathbf{GOF}(n,0) \subseteq \mathbf{CF}$, let $G = (N, T, P_L, P_R, S)$ be a generalized one-sided forbidding grammar of degree $(n,0)$, and let

$$H = \big(N, T, P', S\big)$$

be a context-free grammar with

$$P' = \big\{A \rightarrow x \mid (A \rightarrow x, F) \in P_L \cup P_R\big\}$$

As any successful derivation in G is also a successful derivation in H, the inclusion $L(G) \subseteq L(H)$ holds. On the other hand, let $w \in L(H)$ be a string successfully generated by H. Then, there exists a successful leftmost derivation of w in H (see Theorem 2.3.7). Such a leftmost derivation is, however, also possible in G because the leftmost nonterminal can always be rewritten. Thus, the other inclusion $L(H) \subseteq L(G)$ holds as well, which completes the proof. □

Lemma 8.2.2. $\mathbf{GOF}(0,n) = \mathbf{CF}$ *for every* $n \geq 0$.

Proof. This lemma can be proved by analogy with the proof of Lemma 8.2.1. The only difference is that instead of leftmost derivations, we use rightmost derivations. □

Theorem 8.2.3. $\mathbf{GOF}(n,0) = \mathbf{GOF}(0,n) = \mathbf{CF}$ *for every* $n \geq 0$.

Proof. This theorem follows from Lemmas 8.2.1 and 8.2.2. □

Next, we consider generalized one-sided forbidding grammars of degree $(1,1)$.

Theorem 8.2.4. $\mathbf{CF} \subset \mathbf{GOF}(1,1)$

Proof. This theorem follows from Example 8.1.3. □

In what follows, we prove that generalized one-sided forbidding grammars of degrees $(1,2)$ and $(2,1)$ are computationally complete—that is, they characterize the family of recursively enumerable languages.

Lemma 8.2.5. $\mathbf{GOF}(2,1) = \mathbf{RE}$

Proof. The inclusion $\mathbf{GOF}(2,1) \subseteq \mathbf{RE}$ follows from Church's thesis, so we only prove that $\mathbf{RE} \subseteq \mathbf{GOF}(2,1)$.

Let $K \in \mathbf{RE}$. By Theorem 2.3.9, there is a phrase-structure grammar $G = (N, T, P, S)$ in the Penttonen normal form such that $L(G) = K$. We next construct a generalized one-sided forbidding grammar H of degree $(2,1)$ such that $L(H) = L(G)$. Set

$$W = \big\{ \langle r,i \rangle \mid r = (AB \to AC) \in P, A, B, C \in N, i = 1,2 \big\}$$

Let S' and # be two new symbols. Without any loss of generality, assume that N, W, and $\{S', \#\}$ are pairwise disjoint. Construct

$$H = \big(N', T, P_L, P_R, S' \big)$$

as follows. Initially, set $N' = N \cup W \cup \{S', \#\}$, $P_L = \emptyset$, and $P_R = \emptyset$. Perform (1) through (5), given next.

(1) Add $(S' \to \#S, \emptyset)$ to P_L.
(2) For each $A \to a \in P$, where $A \in N$ and $a \in T$, add $(A \to a, N')$ to P_R.
(3) For each $A \to y \in P$, where $A \in N$ and $y \in \{\varepsilon\} \cup NN$, add $(A \to y, \emptyset)$ to P_L.
(4) For each $r = (AB \to AC) \in P$, where $A, B, C \in N$,

(4.1) add $(B \to \langle r,1 \rangle \langle r,2 \rangle, W)$ to P_L;
(4.2) add $(\langle r,2 \rangle \to C, N'W - \{A\langle r,1 \rangle\})$ to P_L;
(4.3) add $(\langle r,1 \rangle \to \varepsilon, W)$ to P_R.

(5) Add $(\# \rightarrow \varepsilon, N')$ to P_R.

Before proving that $L(H) = L(G)$, let us informally describe (1) through (5). G generates each string of $L(G)$ by simulating the corresponding derivations of H as follows. Every derivation is started by $(S' \rightarrow \#S, \emptyset) \in P_L$, introduced in (1). Context-free rules of the form $A \rightarrow y$, where $A \in N$ and $y \in T \cup \{\varepsilon\} \cup NN$, are simulated by rules from (2) and (3). Since rules introduced in (2) forbid the presence of nonterminals to the right of the rewritten symbol, every sentential form of H is of the form xy, where $x \in N'^*$ and $y \in T^*$—that is, it begins with a string of nonterminals and ends with a string of terminals. In this way, no terminal is followed by a nonterminal. This is needed to properly simulate context-sensitive rules, described next. Rules of the form $AB \rightarrow AC$, where $A, B, C \in N$, are simulated in a three-step way by rules from (4). Observe that the forbidding context of rules from (4.2) ensures that the rewritten symbol B is directly preceded by A. Indeed, if B is not directly preceded by A, then a string different from $A\langle r, 1\rangle$, where $r = (AB \rightarrow AC)$, occurs to the left of $\langle r, 2\rangle$ (recall that # is at the beginning of every sentential form having at least one nonterminal). The end-marker # is erased at the end of every successful derivation by $(\# \rightarrow \varepsilon, N') \in P_R$, introduced in (5).

To establish the identity $L(H) = L(G)$, we prove three claims. Claim 1 demonstrates that every $y \in L(G)$ can be generated by G in two stages; first, only nonterminals are generated, and then, all nonterminals are rewritten to terminals. Claim 2 shows how such derivations of G are simulated by H. Finally, Claim 3 shows how derivations of H are simulated by G.

Claim 1. For every $y \in L(G)$, there exists a derivation of the form $S \Rightarrow_G^* x \Rightarrow_G^* y$, where $x \in N^+$, and during $x \Rightarrow_G^* y$, only rules of the form $A \rightarrow a$, where $A \in N$ and $a \in T$, are applied.

Proof. Let $y \in L(G)$. Since there are no rules in P with symbols from T on their left-hand sides, we can always rearrange all the applications of the rules occurring in $S \Rightarrow_G^* y$ so the claim holds. □

Claim 2. If $S \Rightarrow_G^n x$, where $x \in N^*$, for some $n \geq 0$, then $S' \Rightarrow_H^* \#x$.

Proof. This claim is established by induction on $n \geq 0$.

Basis. Let $n = 0$. Then, for $S \Rightarrow_G^0 S$, there is $S' \Rightarrow_H \#S$ by the rule from (1), so the basis holds.

Induction Hypothesis. Suppose that there exists $n \geq 0$ such that the claim holds for all derivations of length ℓ, where $0 \leq \ell \leq n$.

Induction Step. Consider any derivation of the form

$$S \Rightarrow_G^{n+1} w$$

where $w \in N^*$. Since $n+1 \geq 1$, this derivation can be expressed as

$$S \Rightarrow_G^n x \Rightarrow_G w$$

for some $x \in N^+$. By the induction hypothesis, $S' \Rightarrow_H^* \#x$.

Next, we consider all possible forms of $x \Rightarrow_G w$, covered by the following two cases—(i) and (ii).

(i) Let $A \rightarrow y \in P$ and $x = x_1Ax_2$, where $A \in N$, $x_1, x_2 \in N^*$, and $y \in \{\varepsilon\} \cup NN$. Then,

$$x_1Ax_2 \Rightarrow_G x_1yx_2$$

By (2), $(A \rightarrow y, \emptyset) \in P_L$, so

$$\#x_1Ax_2 \Rightarrow_H \#x_1yx_2$$

which completes the induction step for (i).

(ii) Let $AB \rightarrow AC \in P$ and $x = x_1ABx_2$, where $A, B, C \in N$ and $x_1, x_2 \in N^*$. Then,

$$x_1ABx_2 \Rightarrow_G x_1BCx_2$$

Let $r = (AB \rightarrow AC)$. By (4.1), $(B \rightarrow \langle r,1\rangle\langle r,2\rangle, W) \in P_L$. Since $\mathrm{sub}(\#x_1A) \cap W = \emptyset$,

$$\#x_1ABx_2 \Rightarrow_H \#x_1A\langle r,1\rangle\langle r,2\rangle x_2$$

By (4.2), $(\langle r,2\rangle \rightarrow C, N'W - \{A\langle r,1\rangle\}) \in P_L$. Since $\mathrm{sub}(\#x_1A\langle r,1\rangle) \cap (N'W - \{A\langle r,1\rangle\}) = \emptyset$,

$$\#x_1A\langle r,1\rangle\langle r,2\rangle x_2 \Rightarrow_H \#x_1A\langle r,1\rangle Cx_2$$

By (4.3), $(\langle r,1\rangle \rightarrow \varepsilon, W) \in P_R$. Since $\mathrm{sub}(Cx_2) \cap W = \emptyset$,

$$\#x_1A\langle r,1\rangle Cx_2 \Rightarrow_H \#x_1ACx_2$$

which completes the induction step for (ii).

Observe that cases (i) and (ii) cover all possible forms of $x \Rightarrow_G w$. Thus, the claim holds. □

Set $V = N \cup T$ and $V' = N' \cup T$. Define the homomorpishm τ from V'^* to V^* as $\tau(X) = X$ for all $X \in V$, $\tau(S') = S$, $\tau(\#) = \varepsilon$, $\tau(\langle r,1\rangle) = \varepsilon$ and $\tau(\langle r,2\rangle) = B$ for all $r = (AB \rightarrow AC) \in P$.

Claim 3. If $S' \Rightarrow_H^n x$, where $x \in V'^*$, for some $n \geq 1$, then $S \Rightarrow_G^* \tau(x)$ and x is of the form uv, where $u \in \{\varepsilon\} \cup \{\#\}(N \cup W)^*$, $v \in T^*$, and if $\langle r,2\rangle \in \mathrm{sub}(u)$, then this occurrence of $\langle r,2\rangle$ is directly preceded by $\langle r,1\rangle$.

Proof. This claim is established by induction on $n \geq 0$.

Basis. Let $n = 1$. Then, for $S' \Rightarrow_H \#S$ by the rule from (1), there is $S \Rightarrow_G^0 S$. Since $\#S$ is of the required form, the basis holds.

Induction Hypothesis. Suppose that there exists $n \geq 1$ such that the claim holds for all derivations of length ℓ, where $1 \leq \ell \leq n$.

Induction Step. Consider any derivation of the form

$$S' \Rightarrow_H^{n+1} w$$

where $w \in V'^*$. Since $n + 1 \geq 1$, this derivation can be expressed as

$$S' \Rightarrow_H^n x \Rightarrow_H w$$

for some $x \in V'^+$. By the induction hypothesis, $S \Rightarrow_G^* \tau(x)$ and x is of the form uv, where $u \in \{\varepsilon\} \cup \{\#\}(N \cup W)^*$, $v \in T^*$, and if $\langle r, 2\rangle \in \mathrm{sub}(u)$, then this occurrence of $\langle r, 2\rangle$ is directly preceded by $\langle r, 1\rangle$.

Next, we consider all possible forms of $x \Rightarrow_H w$, covered by the following six cases—(i) through (vi).

(i) *Application of a rule from (2).* Let $x = x_1 A x_2$ and $(A \rightarrow a, N') \in P_R$ so that $N' \cap \mathrm{sub}(x_2) = \emptyset$, where $x_1, x_2 \in V'^*$, $A \in N$, and $a \in T$. Then,

$$x_1 A x_2 \Rightarrow_H x_1 a x_2$$

Clearly, $x_1 a x_2$ is of the required form. By the induction hypothesis, $\tau(x) = \tau(x_1) A \tau(x_2)$. By (2), $A \rightarrow a \in P$, so

$$\tau(x_1) A \tau(x_2) \Rightarrow_G \tau(x_1) a \tau(x_2)$$

which completes the induction step for (i).

(ii) *Application of a rule from (3).* Let $x = x_1 A x_2$ and $(A \rightarrow y, \emptyset) \in P_L$, where $x_1, x_2 \in V'^*$, $A \in N$, and $y \in \{\varepsilon\} \cup NN$. Then,

$$x_1 A x_2 \Rightarrow_H x_1 y x_2$$

Clearly, $x_1 y x_2$ is of the required form. By the induction hypothesis, $\tau(x) = \tau(x_1) A \tau(x_2)$. By (3), $A \rightarrow y \in P$, so

$$\tau(x_1) A \tau(x_2) \Rightarrow_G \tau(x_1) y \tau(x_2)$$

which completes the induction step for (ii).

(iii) *Application of a rule from (4.1).* Let $x = x_1Bx_2$ and $(B \rightarrow \langle r,1\rangle\langle r,2\rangle, W) \in P_L$ so that $W \cap \mathrm{sub}(x_1) = \emptyset$, where $x_1, x_2 \in V'^*$, $B \in N$, and $r = (AB \rightarrow AC) \in P$. Then,

$$x_1Bx_2 \Rightarrow_H x_1\langle r,1\rangle\langle r,2\rangle x_2$$

Clearly, $x_1\langle r,1\rangle\langle r,2\rangle x_2$ is of the required form, and since $\tau(x_1\langle r,1\rangle\langle r,2\rangle x_2) = \tau(x_1)B\tau(x_2)$, the induction step for (iii) follows directly from the induction hypothesis.

(iv) *Application of a rule from (4.2).* Let $x = x_1\langle r,2\rangle x_2$ and $(\langle r,2\rangle \rightarrow C, N'W - \{A\langle r,1\rangle\}) \in P_L$ so that $(N'W - \{A\langle r,1\rangle\}) \cap \mathrm{sub}(x_1) = \emptyset$, where $x_1, x_2 \in V'^*$, $C \in N$, and $r = (AB \rightarrow AC) \in P$. By the induction hypothesis, since $\langle r,2\rangle \in \mathrm{sub}(x)$, x is of the form $\#x'_1\langle r,1\rangle\langle r,2\rangle x_2$, where $x'_1 \in V'^*$. Furthermore, since $(N'W - \{A\langle r,1\rangle\}) \cap \mathrm{sub}(x_1) = \emptyset$, x'_1 is of the form x''_1A. So,

$$\#x''_1A\langle r,1\rangle\langle r,2\rangle x_2 \Rightarrow_H \#x''_1A\langle r,1\rangle Cx_2$$

Clearly, $\#x''_1A\langle r,1\rangle Cx_2$ is of the required form. By the induction hypothesis, $\tau(x) = \tau(x''_1)AB\tau(x_2)$. By (4.2), $AB \rightarrow AC \in P$, so

$$\tau(x''_1)AB\tau(x_2) \Rightarrow_G \tau(x''_1)AC\tau(x_2)$$

which completes the induction step for (iv).

(v) *Application of a rule from (4.3).* Let $x = x_1\langle r,1\rangle x_2$ and $(\langle r,1\rangle \rightarrow \varepsilon, W) \in P_R$ so that $W \cap \mathrm{sub}(x_2) = \emptyset$, where $x_1, x_2 \in V'^*$ and $r = (AB \rightarrow AC) \in P$. Then,

$$x_1\langle r,1\rangle x_2 \Rightarrow_H x_1x_2$$

Clearly, x_1x_2 is of the required form. Since $\tau(x_1\langle r,1\rangle x_2) = \tau(x_1x_2)$, the induction step for (v) follows directly from the induction hypothesis.

(vi) *Application of a rule from (5).* Let $x = \#x'$ and $(\# \rightarrow \varepsilon, N') \in P_R$ so that $N' \cap \mathrm{sub}(x') = \emptyset$ (this implies that $x' \in T^*$). Then, $\#x' \Rightarrow_H x'$. Clearly, x' is of the required form. Since $x' \in T^*$, $\tau(x) = x$, so the induction step for (vi) follows directly from the induction hypothesis.

Observe that cases (i) through (vi) cover all possible forms of $x \Rightarrow_H w$. Thus, the claim holds. □

We next establish the identity $L(H) = L(G)$. Let $y \in L(G)$. Then, by Claim 1, there exists a derivation $S \Rightarrow_G^* x \Rightarrow_G^* y$ such that $x \in N^+$ and during $x \Rightarrow_G^* y$, G uses only rules of the form $A \rightarrow a$, where $A \in N$ and $a \in T$. By Claim 2, $S' \Rightarrow_H^* \#x$. Let $x = X_1X_2 \cdots X_k$ and $y = a_1a_2 \cdots a_k$, where $k = |x|$. Since $x \Rightarrow_G^* y$, $X_i \rightarrow a_i \in P$ for $i = 1, 2, \ldots, k$. By (2), $(X_i \rightarrow a_i, N') \in P_R$ for $i = 1, 2, \ldots, k$. Then,

$$\begin{aligned}\#X_1\cdots X_{k-1}X_k &\Rightarrow_H \#X_1\cdots X_{k-1}a_k\\ &\Rightarrow_H \#X_1\cdots a_{k-1}a_k\\ &\;\vdots\\ &\Rightarrow_H \#a_1a_2\cdots a_k\end{aligned}$$

By (5), $(\# \to \varepsilon, N') \in P_R$. Since $y \in T^*$, $\#y \Rightarrow_H y$. Consequently, $y \in L(G)$ implies that $y \in L(H)$, so $L(G) \subseteq L(H)$.

Consider Claim 3 for $x \in T^*$. Then, $x \in L(H)$ implies that $\tau(x) = x \in L(G)$, so $L(H) \subseteq L(G)$. As $L(G) \subseteq L(H)$ and $L(H) \subseteq L(G)$, $L(H) = L(G)$. Since H is of degree $(2,1)$, the theorem holds. □

Lemma 8.2.6. GOF$(1,2) =$ **RE**

Proof. This lemma can be proved by analogy with the proof of Lemma 8.2.5. First, by modifying the proofs given in [91], we can convert any phrase-structure grammar into an equivalent phrase-structure grammar $G = (N, T, P, S)$, where every rule in P is in one of the following four forms:

(i) $BA \to CA$,
(ii) $A \to BC$,
(iii) $A \to a$,
(iv) $A \to \varepsilon$, where $A, B, C \in N$, and $a \in T$.

Notice that this normal form differs from the Penttonen normal only by the form of context-sensitive rules. Then, in the proof Lemma 8.2.5, we accordingly modify the rules introduced to P_L and P_R so that the resulting grammar is of degree $(1,2)$ instead of $(2,1)$. A rigorous proof of this lemma is left to the reader. □

Theorem 8.2.7. GOF$(1,2) =$ **GOF**$(2,1) =$ **RE**

Proof. This theorem follows from Lemmas 8.2.5 and 8.2.6. □

From Theorem 8.2.7, we obtain the following three corollaries.

Corollary 8.2.8. GOF$(m,n) =$ **RE** *for every $m \geq 2$ and $n \geq 1$.* □

Corollary 8.2.9. GOF$(m,n) =$ **RE** *for every $m \geq 1$ and $n \geq 2$.* □

Corollary 8.2.10. GOF $=$ **RE** □

We next turn our attention to generalized one-sided forbidding grammars with the set of left forbidding rules coinciding with the set of right forbidding rules.

Lemma 8.2.11. *Let K be a context-free language. Then, there exists a generalized one-sided forbidding grammar, $G = (N, T, P_L, P_R, S)$, satisfying $P_L = P_R$ and $L(G) = K$.*

Proof. Let K be a context-free language. Then, there exists a context-free grammar, $H = (N, T, P, S)$, such that $L(H) = K$. Define the generalized one-sided forbidding grammar

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

with

$$P' = \big\{(A \to x, \emptyset, \emptyset) \mid A \to x \in P\big\}$$

Clearly, $L(G) = L(H) = K$, so the lemma holds. □

Lemma 8.2.12. *Let $G = (N,\ T,\ P_L,\ P_R,\ S)$ be a generalized one-sided forbidding grammar satisfying $P_L = P_R$. Then, $L(G)$ is context-free.*

Proof. Let $G = (N, T, P_L, P_R, S)$ be a generalized one-sided forbidding grammar satisfying $P_L = P_R$. Define the context free grammar $H = (N, T, P', S)$ with

$$P' = \big\{A \to x \mid (A \to x, \emptyset, F) \in P_L\big\}$$

Observe that since $P_L = P_R$, it is sufficient to consider just the rules from P_L. As any successful derivation in G is also a successful derivation in H, the inclusion $L(G) \subseteq L(H)$ holds. On the other hand, let $w \in L(H)$ be a string successfully generated by H. Then, there exists a successful leftmost derivation of w in H (see Theorem 2.3.7). Observe that such a leftmost derivation is also possible in G because the leftmost nonterminal can always be rewritten. Indeed, P' contains only rules originating from the rules in P_L and all rules in P_L are applicable to the leftmost nonterminal. Thus, the other inclusion $L(H) \subseteq L(G)$ holds as well, which completes the proof. □

Theorem 8.2.13. *A language K is context-free if and only if there is a generalized one-sided forbidding grammar, $G = (N,\ T,\ P_L,\ P_R,\ S)$, satisfying $K = L(G)$ and $P_L = P_R$.*

Proof. This theorem follows from Lemmas 8.2.11 and 8.2.12. □

In the conclusion of this chapter, we first describe relations of generalized one-sided forbidding grammars to other variants of forbidding grammars. Then, we state several open problems related to the achieved results.

We begin by considering generalized forbidding grammars (see [66]). Let **GF** denote the family of languages generated by these generalized forbidding grammars.

Corollary 8.2.14. GF = GOF

Proof. This corollary follows from Corollary 8.2.10 in this chapter and from Theorem 1 in [66], which says that **GF** = **RE**. □

Next, we move to forbidding grammars (see Definition 2.3.13).

Corollary 8.2.15. For $\subset$ GOF

Proof. This corollary follows from Corollary 8.2.10 in this chapter and from Theorems 2.3.5 and 2.3.15, which imply that **For** $\subset$ **RE**. □

From the definition of a one-sided forbidding grammar, we immediately obtain the following corollary.

Corollary 8.2.16. GOF$(1,1) =$ **OFor** □

The next three open problem areas are related to the achieved results.

Open Problem 8.2.17. By Theorem 8.2.16, **GOF**$(1,1) =$ **OFor**. However, recall that it is not known whether **OFor** $=$ **RE** or **OFor** $\subset$ **RE** (see Chapter 4). Are generalized one-sided forbidding grammars of degree $(1,1)$ capable of generating all recursively enumerable languages? □

Open Problem 8.2.18. Let $G = (N, T, P_L, P_R, S)$ be a generalized one-sided forbidding grammar. If $(A \to x, F) \in P_L \cup P_R$ implies that $|x| \geq 1$, then G is said to be *propagating*. What is the generative power of propagating generalized one-sided forbidding grammars? Do they characterize the family of context-sensitive languages?

□

Open Problem 8.2.19. By Theorem 8.2.7, the degrees $(2,1)$ or $(1,2)$ suffice to characterize the family of recursively enumerable languages. Can we also place a limitation on the number of nonterminals or on the number of rules with nonempty forbidding contexts? Recall that in terms of generalized forbidding grammars, a limitation like this has been achieved (see [59, 66, 76]). □

Chapter 9
LL One-Sided Random Context Grammars

In the previous chapters, have introduced and studied one-sided random context grammars from a purely theoretical viewpoint. From a more practical viewpoint, however, it is also desirable to make use of them in such grammar-based application-oriented fields as syntax analysis (see [1, 2]). An effort like this obviously gives rise to introducing and investigating their parsing-related variants, such as LL versions—the subject of the present chapter.

LL one-sided random context grammars, introduced in this chapter, represent ordinary one-sided random context grammars restricted by analogy with LL requirements placed upon LL context-free grammars. That is, for every positive integer k, (1) LL(k) one-sided random context grammars always rewrite the leftmost nonterminal in the current sentential form during every derivation step, and (2) if there are two or more applicable rules with the same nonterminal on their left-hand sides, then the sets of all terminal strings of length k that can begin a string obtained by a derivation started by using these rules are disjoint. The class of LL grammars is the union of all LL(k) grammars, for every $k \geq 1$.

Recall that one-sided random context grammars characterize the family of recursively enumerable languages (see Theorem 4.1.4). Of course, it is natural to ask whether LL one-sided random context grammars generate the family of LL context-free languages or whether they are more powerful. As its main result, this chapter proves that the families of LL one-sided random context languages and LL context-free languages coincide. Indeed, it describes transformations that convert any LL(k) one-sided random context grammar to an equivalent LL(k) context-free grammar and conversely.

In fact, we take a closer look at the generation of languages by both versions of LL grammars. That is, we demonstrate an advantage of LL one-sided random context grammars over LL context-free grammars. More precisely, for every $k \geq 1$, we present a specific LL(k) one-sided random context grammar G and prove that every equivalent LL(k) context-free grammar has necessarily more nonterminals or rules than G. Thus, to rephrase this result more broadly and pragmatically, we actually show that LL(k) one-sided random context grammars can possibly allow us to specify LL(k) languages more succinctly and economically than LL(k) context-free grammars do.

This chapter is divided into three sections. First, Section 9.1 defines LL one-sided random context grammars. Then, Section 9.2 gives a motivational example. After that, Section 9.3 proves the main result sketched above, and formulates three open problems.

9.1 Definitions

In this section, we define LL context-free grammars and LL one-sided random context grammars. Since we pay a principal attention to context-free and one-sided random context grammars working in the leftmost way, in what follows, by a context-free and one-sided random context grammar, respectively, we always mean a context-free and one-sided random context grammar working in the leftmost way, respectively (see Section 2.3 and Chapter 7). In terms of one-sided random context grammars, by this leftmost way, we mean the type-1 leftmost derivations (see Section 7.1).

We begin by defining the LL(k) property of context-free grammars, for every $k \geq 1$. To simplify the definition, we end all sentential forms by k end-markers, denoted by \$, and we extend the derivation relation to $V^*\{\$\}^k$ in the standard way—that is, $u\$^k \Rightarrow v\$^k$ if and only if $u \Rightarrow v$.

Definition 9.1.1 (see [2]). Let $G = (N, T, P, S)$ be a context-free grammar and $\$ \notin N \cup T$ be a symbol. For every $r = (A \to x) \in P$ and $k \geq 1$, define

$$\text{Predict}_k(r) \subseteq T^*\{\$\}^*$$

as follows: $\gamma \in \text{Predict}_k(r)$ if and only if $|\gamma| = k$ and

$$S\$^k \;{}_{\text{lm}}\!\Rightarrow_G^* uAv\$^k \;{}_{\text{lm}}\!\Rightarrow_G uxv\$^k \;{}_{\text{lm}}\!\Rightarrow_G^* u\gamma w$$

where $u \in T^*$, $v, x \in V^*$, and $w \in V^*\{\$\}^*$. □

Using the above definition, we next define LL context-free grammars.

Definition 9.1.2 (see [2]). Let $G = (N, T, P, S)$ be a context-free grammar. G is an *LL(k) context-free grammar*, where $k \geq 1$, if it satisfies the following condition: if $p = (A \to x) \in P$ and $r = (A \to y) \in P$ such that $x \neq y$, then

$$\text{Predict}_k(p) \cap \text{Predict}_k(r) = \emptyset$$

If there exists $k \geq 1$ such that G is an LL(k) context-free grammar, then G is an *LL context-free grammar*. □

Next, we move to the definition of LL one-sided random context grammars. To simplify this definition, we first introduce the notion of leftmost applicability. Informally, a random context rule r is *leftmost-applicable* to a sentential form y if the leftmost nonterminal in y can be rewritten by applying r.

Definition 9.1.3. Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. A rule $(A \to x, U, W) \in P_L \cup P_R$ is *leftmost-applicable* to $y \in V^*$ if and only if $y = uAv$, where $u \in T^*$ and $v \in V^*$, and

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

or

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

Let us note that the leftmost property of the direct derivation relation has significant consequences to the applicability of rules from P_L. We point out these consequences later in Lemma 9.3.1.

By analogy with the Predict set in context-free grammars, we introduce such a set to one-sided random context grammars. It is then used to define LL one-sided random context grammars. Notice that as opposed to context-free grammars, in the current sentential form, the applicability of a random context rule $(A \to x, U, W)$ depends not only on the presence of A but also on the presence and absence of symbols from U and W, respectively. This has to be properly reflected in the definition.

Definition 9.1.4. Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar and $\$ \notin N \cup T$ be a symbol. For every $r = (A \to x, U, W) \in P_L \cup P_R$ and $k \geq 1$, define

$$\text{Predict}_k(r) \subseteq T^*\{\$\}^*$$

as follows: $\gamma \in \text{Predict}_k(r)$ if and only if $|\gamma| = k$ and

$$S\$^k \; {}_{\text{lm}}^{\;1}\!\!\Rightarrow_G^* \; uAv\$^k \; {}_{\text{lm}}^{\;1}\!\!\Rightarrow_G \; uxv\$^k \; {}_{\text{lm}}^{\;1}\!\!\Rightarrow_G^* \; u\gamma w$$

where $u \in T^*$, $v, x \in V^*$, $w \in V^*\{\$\}^*$, and r is leftmost-applicable to uAv. $\square$

Making use of the above definition, we next define LL one-sided random context grammars.

Definition 9.1.5. Let $G = (N, T, P_L, P_R, S)$ be a one-sided random context grammar. G is an *LL(k) one-sided random context grammar*, where $k \geq 1$, if it satisfies the following condition: for any $p = (A \to x, U, W), r = (A \to x', U', W') \in P_L \cup P_R$ such that $p \neq r$, if $\text{Predict}_k(p) \cap \text{Predict}_k(r) \neq \emptyset$, then there is no $w \in V^*$ such that $S \; {}_{\text{lm}}^{\;1}\!\!\Rightarrow_G^* \; w$ with both p and r being leftmost-applicable to w.

If there exists $k \geq 1$ such that G is an LL(k) one-sided random context grammar, then G is an *LL one-sided random context grammar*. $\square$

9.2 A Motivational Example

In this short section, we give an example of an LL(k) one-sided random context grammar, for every $k \geq 1$. In this example, we argue that LL(k) one-sided random context grammars can describe some languages more succinctly than LL(k) context-free grammars.

Example 9.2.1. Let k be a positive integer and $G = (N, T, \emptyset, P_R, S)$ be a one-sided random context grammar, where $N = \{S\}$, $T = \{a,b,c,d\}$, and

$$P_R = \big\{(S \rightarrow d^{k-1}c, \emptyset, \emptyset), (S \rightarrow d^{k-1}aSS, \emptyset, \{S\}), (S \rightarrow d^{k-1}bS, \{S\}, \emptyset)\big\}$$

Notice that G is an LL(k) one-sided random context grammar. Observe that the second rule can be applied only to a sentential form containing exactly one occurrence of S, while the third rule can be applied only to a sentential form containing at least two occurrences of S. The generated language $L(G)$ can be described by the following expression

$$\big(d^{k-1}a(d^{k-1}b)^*d^{k-1}c\big)^*d^{k-1}c$$

Next, we argue that $L(G)$ cannot be generated by any LL(k) context-free grammar having a single nonterminal and at most three rules. This shows us that for some languages, LL(k) one-sided random context grammars need fewer rules or nonterminals than LL(k) context-free grammars do to describe them.

We proceed by contradiction. Suppose that there exists an LL(k) context-free grammar

$$H = \big(\{S\}, T, P', S\big)$$

such that $L(H) = L(G)$ and $\text{card}(P') \leq 3$. Observe that since there is only a single nonterminal, to satisfy the LL(k) property, the right-hand side of each rule in P' has to start with a string of terminals. Furthermore, since there is only a single nonterminal and all the strings in $L(G)$ begin with either $d^{k-1}a$ or $d^{k-1}c$, each rule has to begin with $d^{k-1}a$ or $d^{k-1}c$. Therefore, to satisfy the LL(k) property, there can be at most two rules. However, then at least one of these rules has to have b somewhere on its right-hand side, so the number of occurrences of b depends on the number of occurrences of a or c. Thus, $L(H) \neq L(G)$, which contradicts $L(H) = L(G)$. Hence, there is no LL(k) context-free grammar that generates $L(G)$ with only a single nonterminal and at most three rules. □

9.3 Generative Power

In this section, we prove that LL one-sided random context grammars characterize the family of LL context-free languages.

First, we establish a normal form for LL one-sided random context grammars, which greatly simplifies the proof of the subsequent Lemma 9.3.2. In this normal form, an LL one-sided random context grammar does not have any left random context rules.

Lemma 9.3.1. *For every LL(k) one-sided random context grammar G, where $k \geq 1$, there is an LL(k) one-sided random context grammar $H = (N, T, \emptyset, P_R, S)$ such that $L(H) = L(G)$.*

Proof. Let $H = (N, T, P_L, P_R, S)$ be an LL(k) one-sided random context grammar, where $k \geq 1$. Construct the one-sided random context grammar

$$H = \big(N, T, \emptyset, P'_R, S\big)$$

where

$$P'_R = P_R \cup \big\{(A \to x, \emptyset, \emptyset) \mid (A \to x, \emptyset, W) \in P_L\big\}$$

Notice that rules from P_L are simulated by right random context rules from P'_R. In a greater detail, let $r = (A \to x, U, W) \in P_L$. Observe that if $U \neq \emptyset$, then r is never applicable in G, so if this is the case, we do not add a rule corresponding to r to P'_R. Furthermore, observe that we do not have to check the absence of nonterminals from W because there are no nonterminals to the left of the leftmost nonterminal in any sentential form.

Clearly, $L(H) = L(G)$ and H is an LL(k) one-sided random context grammar. Hence, the lemma holds. □

To establish the equivalence between LL(k) one-sided random context grammars and LL(k) context-free grammars, we first show how to transform any LL(k) one-sided random context grammar into an equivalent LL(k) context-free grammar. Our transformation is based on the construction used in the proof of Lemma 7.1.3.

Let G be a one-sided random context grammar. By analogy with rule labels in phrase-structure grammars (see page 10), in the remainder of this chapter, we write $x \Rightarrow_G y\ [r]$ to denote that in this derivation step, rule r was used.

Lemma 9.3.2. *For every LL(k) one-sided random context grammar G, where $k \geq 1$, there is an LL(k) context-free grammar H such that $L(H) = L(G)$.*

Proof. Let $G = (N, T, P_L, P_R, S)$ be an LL(k) one-sided random context grammar, where $k \geq 1$. Without any loss of generality, making use of Lemma 9.3.1, we assume that $P_L = \emptyset$. In what follows, angle brackets $\langle$ and $\rangle$ are used to incorporate more symbols into a single compound symbol. Construct the context-free grammar

$$H = \big(N', T, P', \langle S, \emptyset\rangle\big)$$

in the following way. Initially, set

$$N' = \{\langle A, Q\rangle \mid A \in N, Q \subseteq N\}$$

and $P' = \emptyset$. Without any loss of generality, we assume that $N' \cap (N \cup T) = \emptyset$. Next, for each

$$(A \to y_0Y_1y_1Y_2y_2 \cdots Y_hy_h, U, W) \in P_R$$

where $y_i \in T^*$, $Y_j \in N$, for all i and j, $0 \le i \le h$, $1 \le j \le h$, for some $h \ge 0$, and for each $\langle A, Q\rangle \in N'$ such that $U \subseteq Q$ and $W \cap Q = \emptyset$, extend P' by adding

$$\begin{aligned}\langle A, Q\rangle \to y_0 & \langle Y_1, Q \cup \{Y_2, Y_3, \ldots, Y_h\}\rangle y_1 \\ & \langle Y_2, Q \cup \{Y_3, \ldots, Y_h\}\rangle y_2 \\ & \vdots \\ & \langle Y_h, Q\rangle y_h\end{aligned}$$

Before proving that $L(H) = L(G)$, let us give an insight into the construction. As G always rewrites the leftmost occurrence of a nonterminal, we use compound nonterminals of the form $\langle A, Q\rangle$ in H, where A is a nonterminal, and Q is a set of nonterminals that appear to the right of this occurrence of A. When simulating rules from P_R, the check for the presence and absence of symbols is accomplished by using Q. Also, when rewriting A in $\langle A, Q\rangle$ to some y, the compound nonterminals from N' are generated instead of nonterminals from N.

To establish the identity $L(H) = L(G)$, we prove two claims. First, Claim 1 shows how derivations of G are simulated by H. Then, Claim 2 demonstrates the converse—that is, it shows how G simulates derivations of H.

Set $V = N \cup T$ and $V' = N' \cup T$. Define the homomorphism τ from V'^* to V^* as $\tau(\langle A, Q\rangle) = A$ for all $A \in N$ and $Q \subseteq N$, and $\tau(a) = a$ for all $a \in T$.

Claim 1. If $S\ {}_{\rm lm}^{\ 1}\!\!\Rightarrow_G^m x$, *where* $x \in V^*$ *and* $m \ge 0$, *then* $\langle S, \emptyset\rangle\ {}_{\rm lm}\!\!\Rightarrow_H^* x'$, *where* $\tau(x') = x$ *and* x' *is of the form*

$$x' = x_0\langle X_1, \{X_2, X_3, \ldots, X_n\}\rangle x_1\langle X_2, \{X_3, \ldots, X_n\}\rangle x_2 \cdots \langle X_n, \emptyset\rangle x_n$$

where $X_i \in N$ *for* $i = 1, 2, \ldots, n$ *and* $x_j \in T^*$ *for* $j = 0, 1, \ldots, n$, *for some* $n \ge 0$.

Proof. This claim is established by induction on $m \ge 0$.

Basis. Let $m = 0$. Then, for $S\ {}_{\rm lm}^{\ 1}\!\!\Rightarrow_G^0 S$, $\langle S, \emptyset\rangle\ {}_{\rm lm}\!\!\Rightarrow_H^0 \langle S, \emptyset\rangle$, so the basis holds.

Induction Hypothesis. Suppose that there exists $m \ge 0$ such that the claim holds for all derivations of length ℓ, where $0 \le \ell \le m$.

Induction Step. Consider any derivation of the form

$$S\ {}_{\rm lm}^{\ 1}\!\!\Rightarrow_G^{m+1} w$$

where $w \in V^*$. Since $m + 1 \ge 1$, this derivation can be expressed as

$$S\ {}_{\rm lm}^{\ 1}\!\Rightarrow_G^m x\ {}_{\rm lm}^{\ 1}\!\Rightarrow_G w\ [r]$$

for some $x \in V^+$ and $r \in P_R$. By the induction hypothesis, $\langle S, \emptyset \rangle\ {}_{\rm lm}\!\Rightarrow_H^* x'$, where $\tau(x') = x$ and x' is of the form

$$x' = x_0 \langle X_1, \{X_2, X_3, \dots, X_n\} \rangle x_1 \langle X_2, \{X_3, \dots, X_n\} \rangle x_2 \cdots \langle X_n, \emptyset \rangle x_n$$

where $X_i \in N$ for $i = 1, 2, \dots, n$ and $x_j \in T^*$ for $j = 0, 1, \dots, n$, for some $n \geq 1$. As $x\ {}_{\rm lm}^{\ 1}\!\Rightarrow_G w\ [r]$, $x = x_0 X_1 x_1 X_2 x_2 \cdots X_n x_n$, $r = (X_1 \to y, U, W)$, $U \subseteq \{X_2, X_3, \dots, X_n\}$, $W \cap \{X_2, X_3, \dots, X_n\} = \emptyset$, and $w = x_0 y x_1 X_2 x_2 \cdots X_n x_n$. By the construction of H, there is

$$r' = \big(\langle X_1, \{X_2, X_3, \dots, X_n\} \rangle \to y'\big) \in P'$$

where $\tau(y') = y$. Then,

$$x'\ {}_{\rm lm}\!\Rightarrow_H x_0 y' x_1 \langle X_2, \{X_3, \dots, X_n\} \rangle x_2 \cdots \langle X_n, \emptyset \rangle x_n\ [r']$$

Since $w' = x_0 y' x_1 \langle X_2, \{X_3, \dots, X_n\} \rangle x_2 \cdots \langle X_n, \emptyset \rangle x_n$ is of the required form and, moreover, $\tau(w') = w$, the induction step is completed. □

Claim 2. If $\langle S, \emptyset \rangle\ {}_{\rm lm}\!\Rightarrow_H^m x$*, where* $x \in V'^*$ *and* $m \geq 0$*, then* $S\ {}_{\rm lm}^{\ 1}\!\Rightarrow_G^* \tau(x)$ *and* x *is of the form*

$$x_0 \langle X_1, \{X_2, X_3, \dots, X_n\} \rangle x_1 \langle X_2, \{X_3, \dots, X_n\} \rangle x_2 \cdots \langle X_n, \emptyset \rangle x_n$$

where $X_i \in N$ *for* $i = 1, 2, \dots, n$ *and* $x_j \in T^*$ *for* $j = 0, 1, \dots, n$*, for some* $n \geq 0$*.*

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

Basis. Let $m = 0$. Then, for $\langle S, \emptyset \rangle\ {}_{\rm lm}\!\Rightarrow_H^0 \langle S, \emptyset \rangle$, $S\ {}_{\rm lm}^{\ 1}\!\Rightarrow_G^0 S$, so the basis holds.

Induction Hypothesis. Suppose that there exists $m \geq 0$ such that the claim holds for all derivations of length ℓ, where $0 \leq \ell \leq m$.

Induction Step. Consider any derivation of the form

$$\langle S, \emptyset \rangle\ {}_{\rm lm}\!\Rightarrow_H^{m+1} w$$

where $w \in V'^*$. Since $m + 1 \geq 1$, this derivation can be expressed as

$$\langle S, \emptyset \rangle\ {}_{\rm lm}\!\Rightarrow_H^m x\ {}_{\rm lm}\!\Rightarrow_H w\ [r']$$

for some $x \in V^+$ and $r' \in P'$. By the induction hypothesis, $S\ {}_{\rm lm}^{\ 1}\!\Rightarrow_G^* \tau(x)$ and x is of the form

$$x_0 \langle X_1, \{X_2, X_3, \dots, X_n\} \rangle x_1 \langle X_2, \{X_3, \dots, X_n\} \rangle x_2 \cdots \langle X_n, \emptyset \rangle x_n$$

where $X_i \in N$ for $i = 1, 2, \ldots, n$ and $x_j \in T^*$ for $j = 0, 1, \ldots, n$, for some $n \geq 0$. As $x\ {}_{\rm lm}\!\Rightarrow_H w\ [r']$,

$$r' = (\langle X_1, \{X_2, X_3, \ldots, X_n\}\rangle \to y') \in P'$$

where $y' \in V'^*$, and there is $r = (X_1 \to y, U, W) \in P_R$, where $U \subseteq \{X_2, X_3, \ldots, X_n\}$, $W \cap \{X_2, X_3, \ldots, X_n\} = \emptyset$, and $\tau(y') = y$. Then,

$$x_0 X_1 x_1 X_2 x_2 \cdots X_n x_n\ {}_{\rm lm}\!\overset{1}{\Rightarrow}_G x_0 y x_1 X_2 x_2 \cdots X_n x_n\ [r]$$

Since $x_0 y x_1 X_2 x_2 \cdots X_n x_n$ is of the required form and it equals $\tau(w)$, the induction step is completed. $\square$

Consider Claim 1 for $x \in T^*$. Then, $S\ {}_{\rm lm}\!\overset{1}{\Rightarrow}{}^*_G x$ implies that $S\ {}_{\rm lm}\!\Rightarrow^*_H x$, so $L(G) \subseteq L(H)$. Consider Claim 2 for $x \in T^*$. Then, $\langle S, \emptyset\rangle\ {}_{\rm lm}\!\Rightarrow^*_H x$ implies that $S\ {}_{\rm lm}\!\overset{1}{\Rightarrow}{}^*_G x$, so $L(H) \subseteq L(G)$. Hence, $L(H) = L(G)$.

Finally, we argue that H is an LL(k) context-free grammar. To simplify the argumentation, we establish another claim. It represents a slight modification of Claim 2. Let $\$ \notin V' \cup V$ be an end-marker.

Claim 3. If $\langle S, \emptyset\rangle \$^k\ {}_{\rm lm}\!\Rightarrow^m_H x\$^k$, where $x \in V'^*$ and $m \geq 0$, then $S\$^k\ {}_{\rm lm}\!\overset{1}{\Rightarrow}{}^*_G \tau(x)\$^k$ and x is of the form specified in Claim 2.

Proof. This claim can be established by analogy with the proof of Claim 2, so we leave its proof to the reader. $\square$

For the sake of contradiction, suppose that H is not an LL(k) context-free grammar—that is, assume that there are $p' = (X \to y_1) \in P'$ and $r' = (X \to y_2) \in P'$ such that $y_1 \neq y_2$ and $\mathrm{Predict}_k(p) \cap \mathrm{Predict}_k(r) \neq \emptyset$. Let γ be a string from $\mathrm{Predict}_k(p') \cap \mathrm{Predict}_k(r')$. By the construction of P', $X = \langle A, Q\rangle$, for some $A \in N$ and $Q \subseteq N$, and there are $p = (A \to \tau(y_1), U_1, W_1) \in P_R$ and $r = (A \to \tau(y_2), U_2, W_2) \in P_R$ such that $U_1 \subseteq Q$, $U_2 \subseteq Q$, $W_1 \cap Q = \emptyset$, and $W_2 \cap Q = \emptyset$. Since $\gamma \in \mathrm{Predict}_k(p') \cap \mathrm{Predict}_k(r')$,

$$\langle S, \emptyset\rangle \$^k\ {}_{\rm lm}\!\Rightarrow^*_H u\langle A, Q\rangle v\$^k\ {}_{\rm lm}\!\Rightarrow_H u y_1 v \$^k\ [p']\ {}_{\rm lm}\!\Rightarrow^*_H u\gamma w_1$$

and

$$\langle S, \emptyset\rangle \$^k\ {}_{\rm lm}\!\Rightarrow^*_H u\langle A, Q\rangle v\$^k\ {}_{\rm lm}\!\Rightarrow_H u y_2 v \$^k\ [r']\ {}_{\rm lm}\!\Rightarrow^*_H u\gamma w_2$$

for some $u \in T^*$, $v \in V'^*$ such that $\mathrm{alph}(\tau(v)) = Q$ (see Claim 3), and $\gamma, w_1, w_2 \in V'^*\{\$\}^*$. Then, by Claim 3,

$$S\$^k\ {}_{\rm lm}\!\overset{1}{\Rightarrow}{}^*_G uA\tau(v)\$^k\ {}_{\rm lm}\!\overset{1}{\Rightarrow}_G u\tau(y_1 v)\$^k\ [p]\ {}_{\rm lm}\!\overset{1}{\Rightarrow}{}^*_G u\gamma\tau(w_1)$$

and

$$S\$^k\ {}_{\rm lm}\!\overset{1}{\Rightarrow}{}^*_G uA\tau(v)\$^k\ {}_{\rm lm}\!\overset{1}{\Rightarrow}_G u\tau(y_1 v)\$^k\ [r]\ {}_{\rm lm}\!\overset{1}{\Rightarrow}{}^*_G u\gamma\tau(w_2)$$

However, by Definition 9.1.4, $\gamma \in \text{Predict}_k(p)$ and $\gamma \in \text{Predict}_k(r)$, so

$$\text{Predict}_k(p) \cap \text{Predict}_k(r) \neq \emptyset$$

Since both p and r have the same left-hand side and since both are leftmost-applicable to $uA\tau(v)$, we have a contradiction with the fact that G is an LL(k) one-sided random context grammar. Hence, H is an LL(k) context-free grammar, and the lemma holds. □

Next, we show how to transform any LL(k) context-free grammar into an equivalent LL(k) one-sided random context grammar, for every $k \geq 1$.

Lemma 9.3.3. *For every LL(k) context-free grammar G, where $k \geq 1$, there is an LL(k) one-sided random context grammar H such that $L(H) = L(G)$.*

Proof. Let $G = (N, T, P, S)$ be an LL(k) context-free grammar, where $k \geq 1$. Then, the one-sided random context grammar $H = (N, T, P', \emptyset, S)$, where

$$P' = \left\{(A \to x, \emptyset, \emptyset) \mid A \to x \in P\right\}$$

is clearly an LL(k) one-sided random context grammar that satisfies $L(H) = L(G)$. Hence, the lemma holds. □

For every $k \geq 1$, let $\mathbf{LL\text{-}CF}(k)$ and $\mathbf{LL\text{-}ORC}(k)$ denote the families of languages generated by LL(k) context-free grammars and LL(k) one-sided random context grammars, respectively.

The following theorem represents the main result of this chapter.

Theorem 9.3.4. $\mathbf{LL\text{-}ORC}(k) = \mathbf{LL\text{-}CF}(k)$ *for* $k \geq 1$.

Proof. This theorem follows from Lemmas 9.3.2 and 9.3.3. □

Define the language families $\mathbf{LL\text{-}CF}$ and $\mathbf{LL\text{-}ORC}$ as

$$\mathbf{LL\text{-}CF} = \bigcup_{k \geq 1} \mathbf{LL\text{-}CF}(k)$$

$$\mathbf{LL\text{-}ORC} = \bigcup_{k \geq 1} \mathbf{LL\text{-}ORC}(k)$$

From Theorem 9.3.4, we obtain the following corollary.

Corollary 9.3.5. $\mathbf{LL\text{-}ORC} = \mathbf{LL\text{-}CF}$ □

We conclude this chapter by proposing three open problem areas as suggested topics of future investigations related to the topic of the present chapter.

Open Problem 9.3.6. Is the LL property of LL one-sided random context grammars decidable? Reconsider the construction in the proof of Lemma 9.3.2. Since it is decidable whether a given context-free grammar is an LL context-free grammar (see [2]), we might try to convert a one-sided random context grammar G into a context-free grammar H, and show that G is an LL one-sided random context grammar if and only if H is an LL context-free grammar. The only-if part of this equivalence follows from Lemma 9.3.2. However, the if part does not hold. Indeed, we give an example of a one-sided random context grammar that is not LL, but which the construction in the proof of Lemma 9.3.2 turns into a context-free grammar that is LL. Consider the one-sided random context grammar

$$G = \big(\{S, F_1, F_2\}, \{a\}, \emptyset, P_R, S\big)$$

where

$$P_R = \big\{(S \to a, \emptyset, \{F_1\}), (S \to a, \emptyset, \{F_2\})\big\}$$

Clearly, G is not an LL one-sided random context grammar. However, observe that the construction converts G into the LL context-free grammar

$$H = \big(N, \{a\}, P', \langle S, \emptyset\rangle\big)$$

where P' contains $\langle S, \emptyset\rangle \to a$ (other rules are never applicable so we do not list them here). Hence, the construction used in the proof of Lemma 9.3.2 cannot be straightforwardly used for deciding the LL property of one-sided random context grammars. □

Open Problem 9.3.7. Reconsider the proof of Lemma 9.3.2. Observe that for a single right random context rule from P_R, the construction introduces several rules to P', depending on the number of nonterminals of G. Hence, H contains many more rules than G. Obviously, we may eliminate all useless nonterminals and rules from H by using standard methods. However, does an elimination like this always result into the most economical context-free grammar? In other words, is there an algorithm which, given G, finds an equivalent LL(k) context-free grammar H such that there is no other equivalent LL(k) context-free grammar with fewer nonterminals or rules than H has? □

Open Problem 9.3.8. Given an LL(k) context-free grammar G, where $k \geq 1$, is there an algorithm which converts G into an equivalent LL(k) one-sided random context grammar that contains fewer rules than G? □

Part III
Conclusion

This final part of the present thesis closes its discussion by adding remarks regarding its coverage. Most of these remarks concern application perspectives and open problem areas. The historical development of the discussed types of regulated grammars is described as well, and this description includes many relevant bibliographic comments and references.

This part consists of a single Chapter 10, which discusses all the above-mentioned topics.

Chapter 10
Concluding Remarks

This concluding chapter makes several final remarks concerning the material covered in this thesis with a special focus on its future developments. First, it suggests application perspectives of one-sided random context grammars (Section 10.1). Then, it chronologically summarizes the concepts and results achieved in most significant studies on the subject of the present thesis (Section 10.2). Finally, this chapter lists the most important open problems resulting from the study of this thesis (Section 10.3).

10.1 Application Perspectives

As already stated in Chapter 1, this thesis is primarily and principally meant as a theoretical treatment of one-sided random context grammars. Nevertheless, to demonstrate their possible practical importance, we make some general remarks regarding their applications in the present section.

Taking the definition and properties of one-sided random context grammars into account, we see that they are suitable to underly information processing based on the existence or absence of some information parts. Therefore, in what follows, we pay major attention to this application area.

Molecular Genetics

We believe that one-sided random context grammars can formally and elegantly simulate processing information in molecular genetics, including information concerning macromolecules, such as DNA, RNA, and polypeptides. For instance, consider an organism consisting of DNA molecules made by enzymes. It is a common phenomenon that a molecule *m* made by a specific enzyme can be modified unless molecules made by some other enzymes occur either to the left or to the right of *m* in the organism. Consider a string *w* that formalizes this organism so every molecule is represented by a symbol. As obvious, to simulate a change of the symbol *a* that represents *m* requires random context occurrences of some symbols that either precede or follow *a* in *w*. As obvious, one-sided random context grammars can provide a string-changing formalism that can capture this random context requirement in a very succinct and elegant way. To put it more generally, one-sided random context grammars can simulate the behavior of molecular organisms in a rigorous and uniform way.

Computer Science

Considering that one-sided random context grammars have a greater power than context-free grammars, we may immediately think of applying them in terms of syntax analysis of complicated non-context-free structures during language translation. However, as one-sided random context grammars are computationally complete (see Theorem 4.1.4), Rice's theorem (see Section 9.3.3 in [37]) implies that we cannot use them to parse all recursively enumerable languages. Therefore, we should focus on variants of one-sided random context grammars that are not computationally complete, such as propagating one-sided random context grammars.

In Chapter 9, we have studied LL versions of one-sided random context grammars, which may be suitable for syntax analysis. Even though they are equally powerful as context-free grammars (see Corollary 9.3.5), they still may be useful since for some languages, they can describe languages in a more economical way (see Section 9.2).

Linguistics

In terms of linguistics, one-sided random context grammars may be used for generating or verifying that the given texts contain no forbidding passages, such as vulgarisms or classified information. More specifically, generalized one-sided forbidding grammars (see Chapter 8), which are one-sided forbidding grammars that can forbid the occurrences of strings, are suitable to formally capture such applications.

Another application area of one-sided random context grammars may be syntax-oriented linguistics. Observe that many common English sentences contain expressions and words that mutually depend on each other although they are not adjacent to each other in the sentences. For example, consider the following sentence: *He sometimes goes to bed very late.* The subject (*he*) and the predicator (*goes*) are related. Therefore, we cannot rewrite *goes* to *go* because of the subject. One-sided random context grammars form a suitable formalism to capture and verify such dependencies.

Application-oriented topics like the ones outlined in this section obviously represent a future investigation area concerning one-sided random context grammars.

10.2 Bibliographical and Historical Remarks

This section gives an overview of the crucially important studies published on the subject of this thesis from a historical perspective. As this thesis represents primarily a theoretically oriented treatment, we concentrate our attention primarily on theoretical studies.

For a summary of the fundamental knowledge about regulated grammars published by 1989, consult [16]. Furthermore, Chapter 13 of [54] and Chapter 3 of [99] give a brief overview of recent results concerning regulated grammars. The book [79] summarizes recent results concerning various transformations of regulated grammars. More specifically, it concentrates its attention on algorithms that transform these grammars and some related regulated language-defining models so the resulting transformed models are equivalent and, in addition, satisfy some prescribed properties. Finally and most importantly, [87] gives an up-to-date overview of both classical and very recent results concerning regulated grammars and automata.

Random context grammars were introduced in [106]. Strictly speaking, in [106], their definition coincides with the definition of permitting grammars in this thesis. Forbidding grammars, also known as *N-grammars* (see [92]), together with other variants of random context grammars were originally studied by Lomkovskaya in [48–50]. After these studies, many more papers discussed these grammars, including [3, 18–21, 56, 77, 111]. Generalized forbidding grammars were introduced in [66] and further investigated in [60, 76, 77]. In [15, 58, 63], simplified versions of random context grammars, called *restricted context-free grammars*, were studied. Finally, [10, 31, 47, 57] studied grammar systems with their components represented by random context grammars.

Selective substitution grammars were introduced in [95] and further studied in many papers, including [17, 32–34, 39–45, 96, 102, 103].

Originally, scattered context grammars were defined in [35]. Their original version disallowed erasing rules, however. Four years later, [105] generalized them to scattered context grammars with erasing rules (see also [67]). For an in-depth overview of scattered context grammars and their applications, consult [73] and the references given therein.

One-sided random context grammars were introduced in [80]. Their special variants, left permitting and left forbidding grammars, were originally introduced in [10] and [31], respectively. The generative power of one-sided forbidding grammars and their relation to selective substitution grammars were studied in [82]. The nonterminal complexity of one-sided random context grammars was investigated in [81]. A reduction of the number of right random context rules was the topic of [86]. Several normal forms of these grammars were established in [109]. Leftmost derivations were studied in [83]. The generalized version of one-sided forbidding grammars was introduced and investigated in [84]. A list of open problems concerning these grammars appears in [110]. Finally, the LL versions of one-sided random context grammars are based on [74] and appear in this thesis for the first time.

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 [85], where ET0L grammars (see [97]) and their variants enhanced with left random context were studied. Their nonterminal complexity was investigated in [108]. An improvement of the result achieved in [108] appears in Section 10.4 of [87].

10.3 Open Problem Areas

Throughout this thesis, we have already formulated many open problems. Out of them, we next select and repeat the most important questions, which deserve our special attention. To see their significance completely, however, we suggest that the reader returns to the referenced parts of the thesis in order to view these questions in the full context of their formulation and discussion in detail.

(I) What is the generative power of left random context grammars? What is the role of erasing rules in this left variant? That is, are left random context grammars more powerful than propagating left random context grammars?

(II) What is the generative power of one-sided forbidding grammars? We only know that they are equally powerful as selective substitution grammars (see Theorems 4.2.3 and 4.2.4). Thus, by establishing the generative power of one-sided forbidding grammars, we would establish the power of selective substitution grammars, too.

(III) By Theorem 6.1.1, ten nonterminals suffice to generate any recursively enumerable language by a one-sided random context grammar. Is this limit optimal? In other words, can Theorem 6.1.1 be improved?

(IV) Recall that propagating one-sided random context grammars characterize the family of context-sensitive languages (see Theorem 4.1.3). Can we also limit the overall number of nonterminals in terms of this propagating version like in Theorem 6.1.1?

(V) What is the generative power of one-sided forbidding grammars and one-sided permitting grammars? Moreover, what is the power of left permitting grammars? Recall that every propagating scattered context grammar can be turned to an equivalent context-sensitive grammar (see Theorem 3.21 in [73]), but it is a longstanding open problem whether these two kinds of grammars are actually equivalent—the *PSC = CS problem*. If in the future one proves that propagating one-sided permitting grammars and propagating one-sided random context grammars are equivalent, then so are propagating scattered context grammars and context-sensitive grammars (see Theorem 4.3.3), so the PSC = CS problem would be solved.

(VI) By Theorem 6.2.5, any recursively enumerable language is generated by a one-sided random context grammar having no more than two right random context nonterminals. Does this result hold with one or even zero right random context nonterminals? Notice that by proving that no right random context nonterminals are needed, we would establish the generative power of left random context grammars.

(VII) By Theorem 6.3.1, any recursively enumerable language is generated by a one-sided random context grammar having no more than two right random context rules. Does this result hold with one or even zero right random context rules? Again, notice that by proving that no right random context rules are needed, we would establish the generative power of left random context grammars.

References

[1] Aho, A.V., Lam, M.S., Sethi, R., Ullman, J.D.: Compilers: Principles, Techniques, and Tools, 2nd edn. Addison-Wesley, Boston (2006)

[2] Aho, A.V., Ullman, J.D.: The Theory of Parsing, Translation and Compiling, Volume I: Parsing. Prentice-Hall, New Jersey (1972)

[3] Atcheson, B., Ewert, S., Shell, D.: A note on the generative capacity of random context. South African Computer Journal **36**, 95–98 (2006)

[4] Baker, B.S.: Non-context-free grammars generating context-free languages. Information and Control **24**(3), 231–246 (1974)

[5] Cannon, R.L.: Phrase structure grammars generating context-free languages. Information and Control **29**(3), 252–267 (1975)

[6] Chomsky, N.: Three models for the description of language. IRE Transactions on Information Theory **2**(3), 113–124 (1956)

[7] Chomsky, N.: On certain formal properties of grammars. Information and Control **2**, 137–167 (1959)

[8] Cojocaru, L., Mäkinen, E.: On the complexity of Szilard languages of regulated grammars. Tech. rep., Department of Computer Sciences, University of Tampere, Tampere, Finland (2010)

[9] Cremers, A.B., Maurer, H.A., Mayer, O.: A note on leftmost restricted random context grammars. Information Processing Letters **2**(2), 31–33 (1973)

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

[11] Csuhaj-Varjú, E., Vaszil, G.: Scattered context grammars generate any recursively enumerable language with two nonterminals. Information Processing Letters **110**(20), 902–907 (2010)

[12] Cytron, R., Fischer, C., LeBlanc, R.: Crafting a Compiler. Addison-Wesley, Boston (2009)

[13] Czeizler, E., Czeizler, E., Kari, L., Salomaa, K.: On the descriptional complexity of Watson-Crick automata. Theoretical Computer Science **410**(35), 3250–3260 (2009)

[14] Dassow, J., Fernau, H., Păun, G.: On the leftmost derivation in matrix grammars. International Journal of Foundations of Computer Science **10**(1), 61–80 (1999)

[15] Dassow, J., Masopust, T.: On restricted context-free grammars. Journal of Computer and System Sciences **78**(1), 293–304 (2012)

[16] Dassow, J., Păun, G.: Regulated Rewriting in Formal Language Theory. Springer, New York (1989)

[17] Ehrenfeucht, A., Kleijn, H.C.M., Rozenberg, G.: Adding global forbidding context to context-free grammars. Theoretical Computer Science **37**, 337–360 (1985)

[18] Ewert, S., Walt, A.: A shrinking lemma for random forbidding context languages. Theoretical Computer Science **237**(1–2), 149–158 (2000)

[19] Ewert, S., Walt, A.: A pumping lemma for random permitting context languages. Theoretical Computer Science **270**(1–2), 959–967 (2002)

[20] Ewert, S., Walt, A.: The power and limitations of random context. In: Grammars and Automata for String Processing: from Mathematics and Computer Science to Biology, pp. 33–43. Taylor and Francis (2003)

[21] Ewert, S., Walt, A.: Necessary conditions for subclasses of random context languages. Theoretical Computer Science **475**, 66–72 (2013)

[22] Fernau, H.: Regulated grammars under leftmost derivation. Grammars **3**(1), 37–62 (2000)

[23] Fernau, H.: Nonterminal complexity of programmed grammars. Theoretical Computer Science **296**(2), 225–251 (2003)

[24] Fernau, H., Freund, R., Oswald, M., Reinhardt, K.: Refining the nonterminal complexity of graph-controlled, programmed, and matrix grammars. Journal of Automata, Languages and Combinatorics **12**(1–2), 117–138 (2007)

[25] Fernau, H., Meduna, A.: On the degree of scattered context-sensitivity. Theoretical Computer Science **290**(3), 2121–2124 (2003)

[26] Fernau, H., Meduna, A.: A simultaneous reduction of several measures of descriptional complexity in scattered context grammars. Information Processing Letters **86**(5), 235–240 (2003)

[27] Ferretti, C., Mauri, G., Păun, G., Zandron, C.: On three variants of rewriting P systems. Theoretical Computer Science **301**(1–3), 201–215 (2003)

[28] Freund, R., Oswald, M.: P systems with activated/prohibited membrane channels. In: Membrane Computing, *Lecture Notes in Computer Science*, vol. 2597, pp. 261–269. Springer Berlin / Heidelberg (2003)

[29] Geffert, V.: Normal forms for phrase-structure grammars. Theoretical Informatics and Applications **25**(5), 473–496 (1991)

[30] Ginsburg, S., Spanier, E.H.: Control sets on grammars. Theory of Computing Systems **2**(2), 159–177 (1968)
[31] Goldefus, F., Masopust, T., Meduna, A.: Left-forbidding cooperating distributed grammar systems. Theoretical Computer Science **20**(3), 1–11 (2010)
[32] Gonczarowski, J., Kleijn, H.C.M., Rozenberg, G.: Closure properties of selective substitution grammars (part I). International Journal of Computer Mathematics **14**, 19–42 (1983)
[33] Gonczarowski, J., Kleijn, H.C.M., Rozenberg, G.: Closure properties of selective substitution grammars (part II). International Journal of Computer Mathematics **14**, 109–135 (1983)
[34] Gonczarowski, J., Kleijn, H.C.M., Rozenberg, G.: Grammatical constructions in selective substitution grammars. Acta Cybernetica **6**, 239–269 (1984)
[35] Greibach, S.A., Hopcroft, J.E.: Scattered context grammars. Journal of Computer and System Sciences **3**(3), 233–247 (1969)
[36] Holzer, M., Kutrib, M.: Nondeterministic finite automata—recent results on the descriptional and computational complexity. In: Implementation and Applications of Automata, *Lecture Notes in Computer Science*, vol. 5148, pp. 1–16. Springer (2008)
[37] Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation, 3rd edn. Addison-Wesley, Boston (2006)
[38] Kasai, T.: An hierarchy between context-free and context-sensitive languages. Journal of Computer and System Sciences **4**, 492–508 (1970)
[39] Kleijn, H.C.M.: Selective substitution grammars based on context-free productions. Ph.D. thesis, Leiden University, Netherlands (1983)
[40] Kleijn, H.C.M.: Basic ideas of selective substitution grammars. In: Trends, Techniques, and Problems in Theoretical Computer Science, *Lecture Notes in Computer Science*, vol. 281, pp. 75–95. Springer (1987)
[41] Kleijn, H.C.M., Rozenberg, G.: Context-free like restrictions on selective rewriting. Theoretical Computer Science **16**, 237–269 (1981)
[42] Kleijn, H.C.M., Rozenberg, G.: A general framework for comparing sequential and parallel rewriting. In: Mathematical Foundations of Computer Science, pp. 360–368 (1981)
[43] Kleijn, H.C.M., Rozenberg, G.: On the role of selectors in selective substitution grammars. In: Fundamentals of Computation Theory, vol. 117, pp. 190–198 (1981)
[44] Kleijn, H.C.M., Rozenberg, G.: Sequential, continuous and parallel grammars. Information and Control **48**(3), 221–260 (1981)
[45] Kleijn, H.C.M., Rozenberg, G.: On the generative power of regular pattern grammars. Acta Informatica **20**, 391–411 (1983)
[46] Kuroda, S.Y.: Classes of languages and linear-bounded automata. Information and Control **7**(2), 207–223 (1964)

[47] Křivka, Z., Masopust, T.: Cooperating distributed grammar systems with random context grammars as components. Acta Cybernetica **20**(2), 269–283 (2011)

[48] Lomkovskaya, M.V.: Conditional grammars and intermediate classes of languages (in Russian). Soviet Mathematics – Doklady **207**, 781–784 (1972)

[49] Lomkovskaya, M.V.: On *c*-conditional and other commutative grammars (in Russian). Nauchno-Tekhnicheskaya Informatsiya **2**(2), 28–31 (1972)

[50] Lomkovskaya, M.V.: On some properties of *c*-conditional grammars (in Russian). Nauchno-Tekhnicheskaya Informatsiya **2**(1), 16–21 (1972)

[51] Lukáš, R., Meduna, A.: Multigenerative grammar systems. Schedae Informaticae **2006**(15), 175–188 (2006)

[52] Luker, M.: A generalization of leftmost derivations. Theory of Computing Systems **11**(1), 317–325 (1977)

[53] Madhu, M.: Descriptional complexity of rewriting P systems. Journal of Automata, Languages and Combinatorics **9**(2–3), 311–316 (2004)

[54] Martín-Vide, C., Mitrana, V., Păun, G. (eds.): Formal Languages and Applications. Springer, Berlin (2004)

[55] Masopust, T.: Descriptional complexity of multi-parallel grammars. Information Processing Letters **108**(2), 68–70 (2008)

[56] Masopust, T.: On the descriptional complexity of scattered context grammars. Theoretical Computer Science **410**(1), 108–112 (2009)

[57] Masopust, T.: On the terminating derivation mode in cooperating distributed grammar systems with forbidding components. International Journal of Foundations of Computer Science **20**(2), 331–340 (2009)

[58] Masopust, T.: Simple restriction in context-free rewriting. Journal of Computer and System Sciences **76**(8), 837–846 (2010)

[59] Masopust, T., Meduna, A.: Descriptional complexity of generalized forbidding grammars. In: Proceedings of 9th International Workshop on Descriptional Complexity of Formal Systems, pp. 170–177. University of Pavol Jozef Šafárik, SK (2007)

[60] Masopust, T., Meduna, A.: Descriptional complexity of grammars regulated by context conditions. In: LATA '07 Pre-proceedings. Reports of the Research Group on Mathematical Linguistics 35/07, Universitat Rovira i Virgili, pp. 403–411 (2007)

[61] Masopust, T., Meduna, A.: On descriptional complexity of partially parallel grammars. Fundamenta Informaticae **87**(3), 407–415 (2008)

[62] Masopust, T., Meduna, A.: Descriptional complexity of three-nonterminal scattered context grammars: An improvement. In: Proceedings of 11th International Workshop on Descriptional Complexity of Formal Systems, pp. 235–245. Otto-von-Guericke-Universität Magdeburg (2009)

[63] Masopust, T., Meduna, A.: On context-free rewriting with a simple restriction and its computational completeness. RAIRO – Theoretical Informatics and Applications – Informatique Théorique et Applications **43**(2), 365–378 (2009)

[64] Matthews, G.H.: A note on asymmetry in phrase structure grammars. Information and Control **7**, 360–365 (1964)

[65] Maurer, H.A.: Simple matrix languages with a leftmost restriction. Information and Control **23**(2), 128–139 (1973)

[66] Meduna, A.: Generalized forbidding grammars. International Journal of Computer Mathematics **36**(1-2), 31–38 (1990)

[67] Meduna, A.: Syntactic complexity of scattered context grammars. Acta Informatica **1995**(32), 285–298 (1995)

[68] Meduna, A.: On the number of nonterminals in matrix grammars with leftmost derivations. In: New Trends in Formal Languages — Control, Cooperation, and Combinatorics (to Jürgen Dassow on the occasion of his 50th birthday), pp. 27–38. Springer, New York (1997)

[69] Meduna, A.: Automata and Languages: Theory and Applications. Springer, London (2000)

[70] Meduna, A.: Elements of Compiler Design. Auerbach Publications, Boston (2007)

[71] Meduna, A., Goldefus, F.: Weak leftmost derivations in cooperative distributed grammar systems. In: 5th Doctoral Workshop on Mathematical and Engineering Methods in Computer Science, pp. 144–151. Brno University of Technology, Brno, CZ (2009)

[72] Meduna, A., Techet, J.: Canonical scattered context generators of sentences with their parses. Theoretical Computer Science **2007**(389), 73–81 (2007)

[73] Meduna, A., Techet, J.: Scattered Context Grammars and their Applications. WIT Press, Southampton (2010)

[74] Meduna, A., Vrábel, L., Zemek, P.: LL one-sided random context grammars. Unpublished manuscript

[75] Meduna, A., Škrkal, O.: Combined leftmost derivations in matrix grammars. In: Proceedings of 7th International Conference on Information Systems Implementation and Modelling (ISIM'04), pp. 127–132. Ostrava, CZ (2004)

[76] Meduna, A., Švec, M.: Descriptional complexity of generalized forbidding grammars. International Journal of Computer Mathematics **80**(1), 11–17 (2003)

[77] Meduna, A., Švec, M.: Grammars with Context Conditions and Their Applications. Wiley, New Jersey (2005)

[78] Meduna, A., Zemek, P.: One-sided random context grammars: A survey. Unpublished manuscript

[79] Meduna, A., Zemek, P.: Regulated Grammars and Their Transformations. Faculty of Information Technology, Brno University of Technology, Brno, CZ (2010)
[80] Meduna, A., Zemek, P.: One-sided random context grammars. Acta Informatica **48**(3), 149–163 (2011)
[81] Meduna, A., Zemek, P.: Nonterminal complexity of one-sided random context grammars. Acta Informatica **49**(2), 55–68 (2012)
[82] Meduna, A., Zemek, P.: One-sided forbidding grammars and selective substitution grammars. International Journal of Computer Mathematics **89**(5), 586–596 (2012)
[83] Meduna, A., Zemek, P.: One-sided random context grammars with leftmost derivations. In: LNCS Festschrift Series: Languages Alive, vol. 7300, pp. 160–173. Springer Verlag (2012)
[84] Meduna, A., Zemek, P.: Generalized one-sided forbidding grammars. International Journal of Computer Mathematics **90**(2), 127–182 (2013)
[85] Meduna, A., Zemek, P.: Left random context ET0L grammars. Fundamenta Informaticae **123**(3), 289–304 (2013)
[86] Meduna, A., Zemek, P.: One-sided random context grammars with a limited number of right random context rules. Theoretical Computer Science **516**(1), 127–132 (2014)
[87] Meduna, A., Zemek, P.: Regulated Grammars and Automata. Springer, New York (2014)
[88] Mihalache, V.: Matrix grammars versus parallel communicating grammar systems. In: Mathematical Aspects of Natural and Formal Languages, pp. 293–318. World Scientific Publishing, River Edge (1994)
[89] Mutyam, M., Krithivasan, K.: Tissue P systems with leftmost derivation. Ramanujan Mathematical Society Lecture Notes Series **3**, 187–196 (2007)
[90] Okubo, F.: A note on the descriptional complexity of semi-conditional grammars. Information Processing Letters **110**(1), 36–40 (2009)
[91] Penttonen, M.: One-sided and two-sided context in formal grammars. Information and Control **25**(4), 371–392 (1974)
[92] Penttonen, M.: ET0L-grammars and N-grammars. Information Processing Letters **4**(1), 11–13 (1975)
[93] Păun, G.: A variant of random context grammars: semi-conditional grammars. Theoretical Computer Science **41**(1), 1–17 (1985)
[94] Rosenkrantz, D.J.: Programmed grammars and classes of formal languages. Journal of the ACM **16**(1), 107–131 (1969)
[95] Rozenberg, G.: Selective substitution grammars (towards a framework for rewriting systems). Part 1: Definitions and examples. Elektronische Informationsverarbeitung und Kybernetik **13**(9), 455–463 (1977)

[96] Rozenberg, G.: On coordinated selective substitutions: Towards a unified theory of grammars and machines. Theoretical Computer Science **37**, 31–50 (1985)

[97] Rozenberg, G., Salomaa, A.: Mathematical Theory of L Systems. Academic Press, Orlando (1980)

[98] Rozenberg, G., Salomaa, A. (eds.): Handbook of Formal Languages, Vol. 1: Word, Language, Grammar. Springer, New York (1997)

[99] Rozenberg, G., Salomaa, A. (eds.): Handbook of Formal Languages, Vol. 2: Linear Modeling: Background and Application. Springer, New York (1997)

[100] Rozenberg, G., Salomaa, A. (eds.): Handbook of Formal Languages, Vol. 3: Beyond Words. Springer (1997)

[101] Salomaa, A.: Matrix grammars with a leftmost restriction. Information and Control **20**(2), 143–149 (1972)

[102] Siromoney, R., Dare, V.R.: On infinite words obtained by selective substitution grammars. Theoretical Computer Science **39**, 281–295 (1985)

[103] Siromoney, R., Subramanian, K.G.: Selective substitution array grammars. Information Sciences **25**(1), 73–83 (1981)

[104] Vaszil, G.: On the descriptional complexity of some rewriting mechanisms regulated by context conditions. Theoretical Computer Science **330**(2), 361–373 (2005)

[105] Virkkunen, V.: On scattered context grammars. Acta Universitatis Ouluensis **20**(6), 75–82 (1973)

[106] Walt, A.: Random context grammars. In: Proceedings of Symposium on Formal Languages, pp. 163–165 (1970)

[107] Wood, D.: Theory of Computation: A Primer. Addison-Wesley, Boston (1987)

[108] Zemek, P.: On the nonterminal complexity of left random context E0L grammars. In: Proceedings of the 17th Conference STUDENT EEICT 2011, vol. 3, pp. 510–514. Brno University of Technology, Brno, CZ (2011)

[109] Zemek, P.: Normal forms of one-sided random context grammars. In: Proceedings of the 18th Conference STUDENT EEICT 2012, vol. 3, pp. 430–434. Brno University of Technology, Brno, CZ (2012)

[110] Zemek, P.: One-sided random context grammars: Established results and open problems. In: Proceedings of the 19th Conference STUDENT EEICT 2013, vol. 3, pp. 222–226. Brno University of Technology, Brno, CZ (2013)

[111] Zetzsche, G.: On erasing productions in random context grammars. In: ICALP'10: Proceedings of the 37th International Colloquium on Automata, Languages and Programming, pp. 175–186. Springer (2010)

Language Family Index

Subject Index