

Preface

To live, work, and prosper on the Earth, people need to communicate, and they do so by means of a broad variety of languages developed from the dawn of civilization up to today. First and foremost, human beings use natural languages, such as English. In essence, these natural languages represent systems of communication by written and spoken words, used by the people of a particular country or its parts. Furthermore, researchers often express their ideas, concepts, tests, and results in various artificially made languages introduced for specific purposes in their scientific disciplines. For instance, computer scientists have developed hundreds of programming languages in which they write their algorithms so they can be executed on computers. In addition, today's world is overflowed with modern communication machines, such as mobile phones, which gave rise to developing brand new languages for man-machine and machine-machine communication. It thus comes as no surprise that the scientific development and study of languages and their processors fulfill a more important role than ever before.

Of course, we expect that the study of languages produces concepts and results that are solid and reliable. Therefore, we base this study upon mathematics as a systematized body of unshakable knowledge obtained by exact and infallible reasoning. Indeed, mathematics has developed a highly sophisticated theory that specifies languages quite rigorously and formally, hence the name of this theory—formal language theory or, briefly, language theory. From a mathematical viewpoint, this theory defines languages as sets of sequences consisting of symbols. This definition encompasses almost all languages as they are commonly understood. Indeed, natural languages are included in this definition. Of course, all artificial languages introduced by various scientific disciplines represent formal languages as well.

The strict formalization of languages necessitates an introduction of mathematical models that define them. Traditionally, these *language models* are based upon finitely many rules by which they sequentially rewrite sequences of symbols, called words, and that is why they are referred to as *rewriting systems*. They are classified into two basic categories—generating and accepting rewriting systems. Generating systems, better known as *grammars*, define strings of their language so

their rewriting process generates them from a special start symbol. On the other hand, accepting systems, better known as *automata*, define strings of their language by rewriting process that starts from these strings and ends in a special set of strings, usually called final configurations.

However, apart from these traditional language models, language theory has also developed several modern language models based upon rewriting systems that work with words in a nontraditional way, and many of them have their great advantages over their traditional out-of-date counterparts. To give an insight into these advantages, we first need to understand the fundamental problems and difficulties the classical language models cope with. To start with, the traditional language-defining rewriting systems are defined quite generally. Unfortunately, from a practical viewpoint, this generality actually means that the rewriting systems work in a completely unpredictable way. As such, they are hardly implementable and, therefore, applicable in practice. Being fully aware of this pragmatic difficulty, formal language theory has introduced fully deterministic versions of these rewriting systems; sadly, their application-oriented perspectives are also doubtful. First of all, in an ever-changing environment in which real language processors work, it is naive, if not absurd, that these deterministic versions might adequately reflect and simulate real communication technologies applied in such real-world areas as various engineering techniques for language analysis. Furthermore, in many cases, this determinism decreases the power of their general counterparts, which represents another highly undesirable feature of this strict determinism.

Considering these difficulties and drawbacks, formal language theory has recently introduced new versions of rewriting systems, which avoid the disadvantages mentioned above. From a practical viewpoint, an important advantage of these newly introduced rewriting systems consists in controlling their language-defining process and, therefore, operating in a more deterministic way than classical systems, which perform their rewriting process in a quite traditional way. Perhaps even more significantly, the modern versions are stronger than their traditional counterparts. Considering these advantages, it comes as no surprise that formal language theory has paid an incredibly high attention to these modern versions of grammars and automata. Indeed, over the past quarter century, literally hundreds of studies were written about them, and their investigation represents a vivid trend within formal language theory. This investigation has introduced a number of alternative concepts of grammars and automata, and it has achieved many remarkable results. Nevertheless, all these concepts and results are only scattered in various conference and journal papers.

Modern versions of grammars and automata represent the principal *subject* of this book, whose main *focus* is on their concepts, properties, and applications in computer science. The book selects crucially important models and summarizes key results about them in a compact and uniform way. It always relates each of the selected models to a particular way of modern computation, such as computation in parallel or largely cooperating computation. The text explains how the model in question properly reflects and formalizes the corresponding way of computation, so it allows us to obtain a systematized body of mathematically precise knowledge

concerning the computation under investigation. Apart from this obvious theoretical significance, from a more practical viewpoint, the book demonstrates and illustrates how the developers of new computational technologies can make use of this knowledge to build up and implement their modern methods and techniques in the most efficient way.

The text always starts the discussion concerning the language models under consideration by conceptualizing them and linking them to a corresponding form of computation. Then, it gives their mathematical definition, which is also explained intuitively and illustrated by many examples. After that, the text presents most computation-related topics about the models so it proceeds from their (i) theoretical properties through (ii) transformations up to (iii) applications as described next in a greater detail.

- (i) The power of the models represents perhaps the most essential property concerning them. Therefore, the book always determines the language family that the models define. The text also includes many algorithms that modify the models so they satisfy some prescribed properties, which frequently simplify proofs demonstrating results about the models. Apart from this theoretical advantage, the satisfaction of these properties is often strictly required by language processors based on the models.
- (ii) Various transformations of grammars and automata also represent an important investigation area of this book. Specifically, the transformations that reduce the specification of these language models are important to this investigation because the resulting reduced versions of the models define languages in a very succinct and elegant way. As obvious, this reduction simplifies the development of computational technologies, which then work economically and effectively. Of course, the same languages can be defined by different models, and as obvious, every computation-related investigation or application selects the most appropriate models for them under given circumstances. Therefore, whenever discussing different types of equally powerful language models, the book gives transformations that convert them to each other. More specifically, given a language model of one type, the text carefully explains how to transform it to another model so both the original system and the model produced by this transformation define the same language.
- (iii) Finally, the book discusses the use of the models in practice. It describes applications and their perspectives from a general viewpoint. However, the text also covers several real-world applications with a focus on linguistics and biology.

As far as the *writing style* is concerned, we introduce all formalisms with enough rigor to make all results quite clear and valid because we consider this book primarily as a theoretically oriented treatment. Before every complicated mathematical passage, we explain its basic idea intuitively so that even the most complex parts of the book are relatively easy to grasp. We prove most of the results concerning the topics mentioned above effectively—that is, within proofs demonstrating them, we give algorithms that describe how to achieve these results.

For instance, we often present conversions between equally powerful systems as algorithms, whose correctness is then rigorously verified. In this way, apart from their theoretical value, we actually explain how to implement and use them in practice. Several worked-out examples and case studies illustrate this use.

Concerning the *use of the book*, from a general standpoint, this book is helpful to everybody who takes advantage of modern computational technologies based upon grammars or automata. Perhaps most significantly, all scientists who actually make these technologies, ranging from pure mathematicians through computational linguists up to computer engineers, might find this book useful for their work. Furthermore, the entire book can be used as a text for a two-term course in grammars and automata at a graduate level. The text allows the flexibility needed to select some of the discussed topics and, thereby, use it for a one-term course on this subject. Finally, serious undergraduate students may find this book helpful as an accompanying text for a course that deals with formal languages and their models.

Organization and Coverage

The text is divided into six parts, each of which consists of several chapters; altogether, the book contains 19 chapters. Each part starts with an abstract that summarizes its chapters.

Part I, consisting of Chaps. 1 and 2, gives an introduction to this monograph in order that the entire text of the book is completely self-contained. In addition, it places all the coverage of the book into scientific context and reviews important mathematical concepts with a focus on classical language theory.

Part II, which consists of Chaps. 3 through 6, presents an overview of modern grammatical models for languages and corresponding computational modes. Chapter 3 gives the fundamentals of grammars for regulated computation. In essence, these grammars regulate their language generation by additional mechanisms, based upon simple mathematical concepts, such as finite sets of symbols. Chapter 4 studies grammars for computation performed in parallel. These grammars generate their languages in parallel and, thereby, accelerate this generation significantly just like computation in parallel is usually much faster than that made in a sequential way. First, this chapter studies partially parallel generation of languages, after which, it investigates the totally parallel generation of languages. Chapter 5 explores grammars that work on their words in a discontinuous way, thus formalizing a discontinuous way of computation in a very straightforward way. Chapter 6 approaches grammatical models for languages and computation in terms of algebra. In particular, it examines grammatical generation of languages defined over free groups.

Part III consists of Chaps. 7 through 10. To some extent, in terms of automata, this part parallels what Part II covers in terms of grammars. Indeed, Chap. 7 gives the fundamentals of regulated computation formalized by automata. Similarly to grammars discussed in Chap. 5, Chap. 8 formalizes a discontinuous way of

computation. However, Chap. 8 bases this formalization upon automata, which jump across the words they work on discontinuously. Chapter 9 discusses language models for computation based upon new data structures. More specifically, it studies deep pushdown automata, underlined by stacks that can be modified deeper than on their top. Finally, Chap. 10 studies automata that work over free groups, and in this way, it parallels Chap. 6, which studies this topic in grammatical terms.

Part IV, which consists of Chaps. 11 and 12, covers important language-defining devices that combine other rewriting systems, thus formalizing a cooperating way of computation. Chapter 11 untraditionally combines grammars and automata in terms of the way they operate. Specifically, it studies how to generate languages by automata although, traditionally, languages are always generated by grammars. Chapter 12 studies the generation of languages by several grammars that work in a simultaneously cooperative way.

Part V, consisting of Chaps. 13 through 15, discusses computer science applications of rewriting systems studied earlier in the book. First, Chap. 13 covers these computational applications and their perspectives from a rather general viewpoint. Then, more specifically, Chaps. 14 and 15 describe applications in computational linguistics and computational biology, respectively. Both chapters contain several case studies of real-world applications described in detail.

Part VI consists of a single chapter—Chap. 16, which closes the entire book by adding several remarks concerning its coverage. It briefly summarizes all the material covered in the text. Furthermore, it sketches many brand new investigation trends and longtime open problems. Finally, it makes several bibliographical and historical remarks. Further backup materials are available at <http://www.fit.vutbr.cz/~meduna/books/mlmc>.

Brno, Czech Republic
Brno, Czech Republic

Alexander Meduna
Ondřej Soukup

Modern Language Models and Computation

Theory with Applications

Meduna, A.; Soukup, O.

2017, XIX, 548 p. 20 illus., Hardcover

ISBN: 978-3-319-63099-1