# Chapter 1

# Introduction

Formal languages fulfill a crucial role in many computer science areas, ranging from compilers through mathematical linguistics to molecular genetics. Whenever dealing with them, we face the problem of choosing their appropriate models in order to grasp them elegantly and precisely. By analogy with the specification of natural languages, we often base these models upon suitable grammars.

A grammar generates its language by performing derivation steps that change strings, called sentential forms, to other strings according to its grammatical productions. During a derivation step, the grammar rewrites a part of its current sentential form with a string according to one of its productions. If in this way it can make a sequence of derivation steps from its start symbol to a sentential form consisting of terminal symbols—that is, the symbols over which the language is defined, the resulting sentential form is called a sentence and belongs to the generated language. The set of all sentences made in this way is the language generated by the grammar.

In the classical formal language theory, we can divide grammatical productions into context-dependent and context-independent productions, and based on this division, we can naturally distinct context-dependent grammars, such as phrase-structure grammars, from context-independent grammars, such as context-free grammars. Making a derivation step according to context-dependent productions depends on rather strict conditions, usually placed on the context surrounding the rewritten symbol while making a step according to context-independent productions does not. From this point of view, we obviously tend to use context-independent grammars. Unfortunately, compared to context-dependent grammars, context-independent grammars are significantly less powerful; in fact, most of them are incapable to grasp some aspects of quite common programming languages. On the other hand, most context-dependent grammars are as powerful as the Turing machines, and this remarkable power represents their indisputable advantage.

From a realistic point of view, the classical context-independent and context-dependent grammars have some other disadvantages. Consider, for instance, English. Context-independent grammars are obviously incapable of capturing all those contextual dependencies in this complex language. However, we may find even the classical context-dependent grammars clumsy for this purpose. To illustrate, in an English sentence, the proper form of verb usually depends on the form of the subject. For instance, we write *I do it*, not *I ~~does~~ it*, and it is the subject, *I*, that implies the proper form of *do*. Of course, there may occur several words, such as adverbs, between the subject and the verb. We could extend

*I do it* to *I often do it*, *I very often do it* and infinitely many other sentences in this way. At this point, however, the classical context-dependent productions, whose conditions are placed on the context surrounding the rewritten symbol, are hardly of any use because the proper form of the verb follows from the subject that does not surround the verb at all; in fact, it occurs many words ahead of this verb.

To overcome the difficulties and, at the same time, maintain the advantages described above, the modern language theory has introduced some new grammars that simultaneously satisfy these three properties:

- they are based on context-independent productions;
- their context conditions are signficantly more simple and flexible than the strict condition placed on the context surrounding the rewritten symbol in the classical context-dependent grammars;
- they are as powerful as classical context-dependent grammars.

In the present book, we overview the most essential types of these grammars, whose alternative context conditions can be classified into these three categories:

- context conditions placed on derivation domains;
- context conditions placed on the use of productions;
- context conditions placed on the neighborhood of the rewritten symbols.

As already pointed out, we want the context conditions as small as possible. Therefore, we pay a lot of attention to the reduction of context conditions in this book. Specifically, we reduce the number of some of their components, such as the number of nonterminals or productions. We study how to achieve this reduction without any decrease of their generative power, which coincides with the power of the Turing machines. By achieving this reduction, we actually make the grammars with context conditions more succinct and economical, and these properties are obviously highly appreciated both from a practical and theoretical standpoint. Regarding each of the dicussed grammars, we introduce and study their parallel and sequential versions, which represent two basic approaches to grammatical generation of languages in today's formal language theory. To be more specific, during a sequential derivation step, a grammar rewrites a single symbol in the current sentential form while during a parallel derivation step, a grammar rewrites all symbols. As context-free and E0L grammars represent perhaps the most fundamental sequential and parallel grammars, respectively, we usually base the discussion of sequential and parallel generation of languages on them.

## Organization

The text consists of the following chapters:

Chapter 2 gives an introduction to formal languages and their grammars.

Chapter 3 restricts grammatical derivation domains in a very simple and natural way. Under these restrictions, both sequential and parallel context-independent grammars characterize the family of recursively enumerable languages, which are defined by the Turing machines.

Chapter 4 studies grammars with conditional use of productions. In these grammars, productions may be applied on condition that some symbols occur in the current sentential form and some others do not. We discuss many sequential and parallel versions of these grammars in detail. Most importantly, new characterizations of some well-known families of L languages, such as the family of ET0L languages, are obtained.

Chapter 5 studies grammars with context conditions placed on the neighborhood of rewritten symbols. We distinguish between scattered and continuous context neighborhood. The latter strictly requires that the neighborhood of the rewritten symbols forms a continuous part of the sentential form while the former drops this requirement of continuity.

Chapter 6 takes a closer look at grammatical transformations, which are frequently studied in the previous chapters. Specifically, it studies how to transform grammars with context-conditions to some other equivalent grammars so that both the input grammars and the transformed grammars generate their languages in a very similar way.

Chapter 7 demostrates the use of grammars with context conditions by several applications related to biology.

Chapter 8 summarizes the main results of this book and states several open problems. It makes historical notes and suggests some general references regarding the theoretical background of grammars with context conditions. In addition, it proposes new directions in the investigation of these grammars.

## Approach

This book represents a theoretically oriented treatment. As a result, we introduce the formalism concerning grammars with enough rigor to make all results quite clear and valid. Every complicated mathematical passage is preceded by its intuitive explanation so even the most complex parts of the book are easy to grasp. As most proofs of the achieved results contain some transformations of grammars, the present book also maintains an emphasis on algorithmical approach to the gramatical models under discussion and, thereby, their use in practice. Several worked-out examples and realistic applications illustrate the theoretical notions.

## Use

This book is useful to every computer scientist interested in formal languages and their gramatically based models discussed in today's theoretical computer science. It can also be used as an accompanying textbook for an advanced course in theoretical computer science at the senior levels; the text allows the flexibility needed to select some of the discussed topics.