THEORY OF FORMAL LANGUAGES WITH APPLICATIONS

Dan A Simovici
Richard L Tenney

World Scientific

THEORY OF FORMAL LANGUAGES WITH APPLICATIONS

Dan A Simovici
Richard L Tenney
Department of Mathematics and Computer Science,
University of Massachusetts at Boston

World Scientific
Singapore • New Jersey • London • Hong Kong

Published by

World Scientific Publishing Co. Pte. Ltd.

P O Box 128, Farrer Road, Singapore 912805

USA office: Suite 1B, 1060 Main Street, River Edge, NJ 07661

UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

THEORY OF FORMAL LANGUAGES WITH APPLICATIONS

ISBN 981-02-3729-4

Printed in Singapore by Uto-Print

THEORY OF FORMAL LANGUAGES WITH APPLICATIONS

Preface

The theory of formal languages has a long and dignified history. A major influence on the nascent theory, around 1960, were the attempts of the linguist Noam Chomsky to formulate a general theory of the syntax of natural languages. Chomsky's intellectual itinerary greatly influenced the field at a time when computers were starting to cope with increasingly complex tasks. A melting pot of ideas then developed, with a surprising convergence of thought between linguists, mathematicians, logicians, and newly born computer scientists.

At present, formal languages are part of the basic training of most computer scientists. They are everywhere to be found in the design and in the very operation of computer systems. A modem, like an interface manager, will have to respond to various external stimuli. Its design and its behavior are then best understood when it is viewed as a device reacting to external events while being governed by a finite set of rules—in short, a finite automaton. Next, the syntax of programming languages is best described by context-free grammars, themselves recognized by pushdown automata. We then have available one of the fundamental building blocks of the design of parsers and compilers. Finally, the last steps of the complexity ladder take us to languages of a higher structural complexity, which swiftly lead to (un)decidability questions. This brings us to the humbling realization that mathematically well-posed problems are far from being all decidable!

The theory of formal languages and their companion automata thus provides a powerful approach to the design of systems and to a variety of problems in computer science. Dan Simovici and Richard Tenney develop the core theory in a lucid manner. Their self-contained presentation combines mathematical rigor and intellectually stimulating applications. For instance, the reader will find in the book a perspective on algorithms for the processing of text files, lexical analysis, and parsing. A notably innovative aspect is the last part that offers two chapters on coding theory, data compression, as well as biological applications. It should be a pleasure for most to discover there formal models that describe the development of simple organisms or the splicing of nucleic acids.

To make a long story short, we have here a new book that offers new perspectives on an old subject. It contains a thorough treatment of a theory that is fundamental not only in computer science but in many other scientific endeavors. The authors have done a great job of exposition. I hope you will enjoy reading the book as much as I did.

Philippe Flajolet
Rocquencourt, February 28, 1999

Introduction

The theory of formal languages is an important part of the fundamental education of computer scientists and linguists. It is also becoming significant for biologists. This discipline blends algebraic techniques with abstract models of computing devices. Its origins can be traced to the work of Chomsky, Rabin, Scott, Nerode, Ginsburg, and Schützenberger, and this beautiful area of theoretical computer science remains active today. Along the way are such milestones as the theory of abstract families of languages and various applications of the theory of complexity in the study of formal languages.

This book combines algebraic and algorithmic methods with decidability results and explores applications both within and outside computer science.

Formal languages provide the theoretical underpinnings for the study of programming languages. They are also the foundation for compiler design, and they are important in such areas as data compression, computer networks, etc. Recently, formal languages have been applied in biology and economics.

The first part of the book presents mathematical preliminaries. It begins with a chapter that elucidates the mathematical background expected of the reader—elementary notions about sets, algebras, and graphs—as well as the notation that we use. It is intended to make this book as self-contained as practical. The second chapter deals with words and languages viewed as collections of words. These are basic ingredients in the discipline of formal languages, so this chapter presents the most important algebraic and combinatorial properties of words and languages in order to make later chapters more readable.

The second part is centered on regular and context-free languages. The class of regular languages is studied in the third chapter, starting with deterministic finite automata. We then consider various extensions of these devices, including nondeterministic automata and transition systems, as alternative ways of defining the same class of languages. We introduce regular expressions as notations for regular languages, and we conclude the chapter by examining several applications of the notions developed in the chapter.

The fourth chapter introduces the notions of semi-Thue system, and especially important, the notion of grammar. We study Chomsky's hierarchy, and we show the closure of each Chomsky class with respect to the regular operations.

We place particular emphasis on context-free languages due to their role in compiler design. This class of languages is introduced using the class of context-free grammars; the devices that provide an alternative characterization of this class, pushdown automata are discussed in the next chapter. To allow

Contents

Part I

Introductory Notions

Chapter 1

Preliminaries

1.1 Introduction

In this chapter we present some mathematical preliminaries intended to provide a referrence for the reader. We discuss basic facts of set theory, universal algebra and the notion of cardinality of a set.

1.2 Sets, Relations, and Functions

In this section we review briefly some concepts from set theory.

1.2.1 Sets

Informally, a set is a collection of objects, called *elements*. For any set S and object a, either a is one of the objects in S or it is not. In the former case, we use any of the following phrases: "a is a member of S," "a is an element of S," or "a is in S," and we write $a \in S$.

The elements of a set $\mathcal{C}$ may be themselves sets; in such cases, we refer to $\mathcal{C}$ as a *collection* of sets.

We use the following notations for various subsets of the set $\mathbb{R}$ of real num-

bers:

$\mathbb{N}$	for the set of natural numbers
$\mathbb{P}$	for the set of positive integers
$\mathbb{Q}$	for the set of rational numbers
$\mathbb{Z}$	for the set of integers

Let S and T be two sets. The set S is *included* in T (or S is a *subset* of T) if every element of S is an element of T. In this case, we write $S \subseteq T$. If S is not a subset of T, we write $S \not\subseteq T$.

If S and T are sets such that $S \subseteq T$ and $S \neq T$, then we say that S is *strictly included* in T. We denote this by $S \subset T$. If S is not strictly included in T, we write $S \not\subset T$.

There exists a set with no members. This set is called the *empty set* and is denoted by $\emptyset$.

A *singleton* is a set $\{x\}$ that consists of a unique element.

If S is a set, then the *power set* of S is the set that consists of all the subsets of S. We denote the power set of S by $\mathcal{P}(S)$.

Note that for every set S, $\emptyset \in \mathcal{P}(S)$, so $\mathcal{P}(S)$ is never empty.

1.2.2 Ordered Pairs and Cartesian Products

Given two objects a and b, we can form the set $\{a, b\}$, which we refer to as an *unordered pair.*

Definition 1.2.1 Let a and b be two objects. The *ordered pair*, or simply, the *pair*, (a, b) is the collection of sets $\{\{a\}, \{a, b\}\}$. □

The pair (a, a) is the singleton $\{\{a\}\}$. Conversely, if (a, b) is a singleton then $\{a, b\} = \{a\}$, so $b \in \{a\}$, that is, $b = a$.

It is easy to see that if $(a, b) = (c, d)$, then $a = c$ and $b = d$. For the pair (a, b), we call a the *first component* and b its *second component.*

Definition 1.2.2 Let A and B be two sets. The *Cartesian product* of A and B, written $A \times B$, is the set of all pairs (a, b) such that $a \in A$ and $b \in B$. □

If either of the two sets A or B is empty, then so is their Cartesian product.

Definition 1.2.3 Let $\mathcal{C}$ be a collection of sets. The *union* of $\mathcal{C}$, denoted by $\bigcup \mathcal{C}$, is the set defined by

$$\bigcup \mathcal{C} = \{x \mid x \in A \text{ for some } A \in \mathcal{C}\}.$$

□

If $\mathcal{C} = \{A, B\}$, we have $x \in \bigcup \mathcal{C}$ if and only if $x \in A$ or $x \in B$. In this case, $\bigcup \mathcal{C}$ may be denoted $A \cup B$. We refer to the set $A \cup B$ as the *union of A and B.*

For any sets A, B, C, we have

1. $A \cup (B \cup C) = (A \cup B) \cup C = \bigcup \{A, B, C\}$ (associativity of union),
2. $A \cup B = B \cup A$ (commutativity of union),
3. $A \cup A = A$ (idempotency of union), and
4. $A \cup \emptyset = A$.

Let $\mathcal{C}$ be a nonempty collection of sets. The *intersection* of $\mathcal{C}$, denoted by $\bigcap \mathcal{C}$, is the set defined by

$$\bigcap \mathcal{C} = \{x \mid x \in A \text{ for every } A \in \mathcal{C}\}.$$

If $\mathcal{C} = \{A, B\}$, we may denote $\bigcap \mathcal{C}$ by $A \cap B$, and we refer to $A \cap B$ as the *intersection of A and B*. Then, $x \in A \cap B$ if and only if $x \in A$ and $x \in B$.

Definition 1.2.4 Two sets A, B are *disjoint* if $A \cap B = \emptyset$.

A collection of sets $\mathcal{C}$ is *pairwise disjoint* if for every A and B in $\mathcal{C}$, if $A \neq B$, then A and B are disjoint.

A *partition* of a set M is a collection $\mathcal{C} = \{B_i \mid i \in I\}$ of nonempty, pairwise disjoint sets such that $\bigcup \mathcal{C} = M$. We refer to the members of the partition as *blocks* and to I as the *index set.* □

Example 1.2.5 If E, O are the sets of even and odd natural numbers, respectively, then the collection $\{E, O\}$ is a partition of the set $\mathbb{N}$. This partition can be generalized as follows. Let $k \in \mathbb{N}$ be a natural number, $k > 0$. For every number $n \in \mathbb{N}$ there is a unique number $r \in \mathbb{N}$, $0 \leq r \leq k-1$, such that $n = kp + r$, namely the remainder of the division of n by k. Let $B_r = \{n \in \mathbb{N} \mid n = kp + r\}$ for $0 \leq r \leq k-1$. Since the remainder of the division of n by k is uniquely determined for every n, it follows that $\mathcal{C}_k = \{B_0, \ldots, B_{k-1}\}$ is a collection of pairwise disjoint sets. It is easy to see that

$$\bigcup_{0 \leq r \leq k-1} B_r = \mathbb{N},$$

so $\mathcal{C}_k$ is a partition. We refer to $\mathcal{C}_k$ as the *partition of the natural numbers modulo k.*

If $k = 2$, the partition $\{B_0, B_1\}$ is simply the partition $\{E, O\}$ discussed before. □

Definition 1.2.6 Let A, B be two sets. The *difference* of A and B is the set $A - B$ defined by

$$A - B = \{x \in A \mid x \notin B\}.$$

□

Sometimes, when the set A is understood from the context, we write $\overline{B}$ for $A - B$, and we refer to the set $\overline{B}$ as the *complement of B with respect to A* or simply the *complement of B.*

For every set A and nonempty collection $\mathcal{C}$ of sets, we have

$$A - \bigcup \mathcal{C} = \bigcap \{A - C \mid C \in \mathcal{C}\},$$

$$A - \bigcap \mathcal{C} = \bigcup \{A - C \mid C \in \mathcal{C}\}.$$

In the special case when $\mathcal{C} = \{B, C\}$ we have:

$$A - (B \cup C) = (A - B) \cap (A - C),$$

$$A - (B \cap C) = (A - B) \cup (A - C).$$

If the set A is understood, $B, C \subseteq A$, and we denote $A - M$ by $\overline{M}$ for each $M \in \mathcal{P}(M)$, then we get the following equalities

$$\begin{aligned} \overline{B \cup C} &= \overline{B} \cap \overline{C} \\ \overline{B \cap C} &= \overline{B} \cup \overline{C}, \end{aligned}$$

known as the *DeMorgan's laws.*

The linkage between union and intersection is given by the distributivity properties. Namely, for any collection of sets $\mathcal{C}$ and set A, we have

$$\left(\bigcup \mathcal{C}\right) \cap A = \bigcup \{C \cap A \mid C \in \mathcal{C}\}.$$

If $\mathcal{C}$ is nonempty, we also have

$$\left(\bigcap \mathcal{C}\right) \cup A = \bigcap \{C \cup A \mid C \in \mathcal{C}\}.$$

1.2.3 Relations

Definition 1.2.7 A *relation* is a set of ordered pairs. If A and B are sets and ρ is a relation, then we call ρ a *relation from A to B* if $\rho \subseteq A \times B$. A relation from A to A is called a *relation on A.* □

The set $\mathcal{P}(A \times B)$ is the set of all relations from A to B. Among the relations from A to B, we distinguish the *empty relation*, $\emptyset$, and the *full relation*, $A \times B$.

If $(a, b) \in \rho$, we sometimes denote this fact by $a \rho b$, and we write $a \not\rho b$ to denote $(a, b) \notin \rho$.

Example 1.2.8 For any set A, we can consider the *identity relation* $\iota_A \subseteq A \times A$ defined by

$$\iota_A = \{(x, x) \mid x \in A\}.$$

□

Note that $A \subseteq B$ if and only if $\iota_A \subseteq \iota_B$.

Definition 1.2.9 The *domain* of a relation ρ is the set

$$\mathrm{Dom}(\rho) = \{a \mid (a, b) \in \rho \text{ for some } b\}.$$

The *range* of ρ is the set

$$\mathrm{Ran}(\rho) = \{b \mid (a, b) \in \rho \text{ for some } a\}.$$

□

It follows easily that if ρ is a relation and A and B are sets, then ρ is a relation from A to B if and only if $\mathrm{Dom}(\rho) \subseteq A$ and $\mathrm{Ran}(\rho) \subseteq B$. Naturally, ρ is always a relation from $\mathrm{Dom}(\rho)$ to $\mathrm{Ran}(\rho)$.

If ρ and σ are relations and $\rho \subseteq \sigma$, then we have $\mathrm{Dom}(\rho) \subseteq \mathrm{Dom}(\sigma)$ and $\mathrm{Ran}(\rho) \subseteq \mathrm{Ran}(\sigma)$. We also remark that $\mathrm{Dom}(A \times B) = A$ unless $B = \emptyset$. In the latter case, we have $\mathrm{Dom}(A \times \emptyset) = \emptyset$. Similarly, $\mathrm{Ran}(A \times B) = B$ if $A \neq \emptyset$. For $A = \emptyset$, we have $\mathrm{Ran}(\emptyset \times B) = \emptyset$.

Example 1.2.10 Let A be a subset of $\mathbb{N}$. The relation "less than" on A is given by

$$\{(x, y) \mid x, y \in A \text{ and } y = x + z \text{ for some positive integer } z\}.$$

□

Example 1.2.11 Consider the relation $\nu \subseteq \mathbb{Z} \times \mathbb{Q}$ given by

$$\nu = \{(n, q) \mid n \in \mathbb{Z},\ q \in \mathbb{Q},\ \text{and } n \leq q < n + 1\}.$$

We have $(-3, -2.3) \in \nu$ and $(2, 2.3) \in \nu$. Clearly, $(n, q) \in \nu$ if and only if n is the integral part of the rational number q. □

Example 1.2.12 We can define a relation $\delta_{\mathbb{Z}} \subseteq \mathbb{Z} \times \mathbb{Z}$, where $(m, n) \in \delta_{\mathbb{Z}}$ if there is $k \in \mathbb{Z}$ such that $n = mk$. In other words, $(m, n) \in \delta_{\mathbb{Z}}$ if m divides n evenly. □

Example 1.2.13 Let S be a set. Then, the relation from S to $\mathcal{P}(S)$ "is a member of" is given by

$$\{(x, X) \mid X \in \mathcal{P}(S) \text{ and } x \in X\}.$$

□

If ρ and σ are relations from A to B, then so are $\rho \cup \sigma$, $\rho \cap \sigma$, and $\rho - \sigma$.

Definition 1.2.14 Let ρ be a relation. The *inverse* of ρ is the relation ρ^{-1} given by

$$\rho^{-1} = \{(y, x) \mid (x, y) \in \rho\}.$$

□

Theorem 1.2.15 *Let ρ and σ be relations.*

1. $\mathrm{Dom}(\rho^{-1}) = \mathrm{Ran}(\rho)$.
2. $\mathrm{Ran}(\rho^{-1}) = \mathrm{Dom}(\rho)$.
3. *If ρ is a relation from A to B, then ρ^{-1} is a relation from B to A.*
4. $(\rho^{-1})^{-1} = \rho$.
5. *If $\rho \subseteq \sigma$, then $\rho^{-1} \subseteq \sigma^{-1}$ (montonicity of the inverse).*

Proof. We leave these arguments to the reader. ∎

Definition 1.2.16 Let ρ and σ be relations. The *product* of ρ and σ is the relation $\rho \circ \sigma$, where

$$\rho \circ \sigma = \{(x, z) \mid (x, y) \in \rho \text{ and } (y, z) \in \sigma \text{ for some } y\}.$$

□

The product of two relations ρ and σ is also called the *composition* of ρ and σ, and we also use the alternative notation $\sigma\rho$ for the relation product $\rho \circ \sigma$.

Definition 1.2.17 The powers of a relation $\rho \subseteq A \times A$ are relations denoted by $\rho^n \subseteq A \times A$, defined recursively as follows:

$$\begin{aligned} \rho^0 &= \iota_A \\ \rho^{n+1} &= \rho^n \circ \rho, \end{aligned}$$

for $n \in \mathbb{N}$. □