IISc Research Monographs Series

Deepak D'Souza
Priti Shankar
editors

Modern Applications of Automata Theory

IISc Press
World Scientific

IISc Research Monographs Series

Modern Applications of Automata Theory

editors

Deepak D'Souza

Priti Shankar

Indian Institute of Science, India

World Scientific

NEW JERSEY • LONDON • SINGAPORE • BEIJING • SHANGHAI • HONG KONG • TAIPEI • CHENNAI

Published by

World Scientific Publishing Co. Pte. Ltd.
5 Toh Tuck Link, Singapore 596224
USA office: 27 Warren Street, Suite 401-402, Hackensack, NJ 07601
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.

IISc Research Monographs Series — Vol. 2
MODERN APPLICATIONS OF AUTOMATA THEORY

ISBN-13 978-981-4271-04-2
ISBN-10 981-4271-04-7

Printed in Singapore by B & Jo Enterprise Pte Ltd

Modern Applications of Automata Theory

Priti Shankar, 1947–2011.

This book is dedicated to the memory of Professor Priti Shankar who passed away on the 17th of October 2011, while this book was in the final stages of production.

Foreword

Ask yourself what are indispensable parts of a core curriculum in computer science. A few highly hyped subjects will probably not make it into the core. Automata theory however has always been part of the core and, I believe, will always remain in the core. After all, our beloved machines on which we do our daily work are finite state machines albeit with quite a huge number of states.

Finite-state machines, originally invented by McCulloch and Pitts to describe the behavior of nervous systems, found their first formal treatment in Kleene's 1956 paper containing his famous theorem about the equivalence of finite-state machines and regular expressions. Rabin and Scott, in their joint paper, which earned them the Turing Award, introduced the notion of *nondeterministic* finite-state machines and presented the *subset construction* to constructively prove the equivalence of deterministic and nondeterministic finite-state machines.

Compiler writers have recognized quite early that different types of automata were ideally suited to understand and to implement several tasks in compilation: finite state machines for lexical analysis, pushdown automata for syntax analysis, and tree automata for code selection. The correspondence of these automata types to concepts on the formal language side was also recognized early. For the compiler writers this meant that automatic methods for the implementation of compiler tasks from given specifications was possible.

At some point in the development of computer hardware, engineers were overwhelmed by the task of verifying their hardware designs when feature size became smaller and the number of integrated transistors exceeded hundreds of thousands. Automatic or at least semi-automatic methods for the verification and the test of designs were needed. Automata theory profited from its connection to logic, in particular to temporal logic, imported into this field by Amir Pnueli in 1977. Model checking based on this connection was invented as a method of choice independently by Queille and Sifakis and by Clarke and Emerson at the beginning of the eighties. However, the problem of state-space explosion limited its applicability, despite the invention of ingenious data structures such as BDDs for the symbolic representation of the space. At the same time, automata with acceptance criteria defined by Büchi moved into the focus of research on automata. They were needed to reason about systems with infinite computations. In order to make model checking feasible, abstraction to systems with a sufficiently small number of states was necessary. The abstraction should preserve essential properties such that verifiable properties of the abstracted system would definitely hold on the original system. Much progress has been made here in recent times.

The connection to logic has always been a fruitful ground for research. Decidability, complexity, and expressiveness problems had to be solved. Characterizations of variants of automata in terms of logic or formal languages had to be found.

Modern applications of automata theory go far beyond compiler techniques or hardware verification. Automata are widely used for modelling and verification of software, distributed systems, real-time systems, or structured data. They have been equipped with features to model time and probabilities as well.

This volume tackles some of the challenges such as countering ever-growing state spaces by using finite-state abstractions, translations between several temporal logics and Büchi automata, equivalence between several types of automata and different types of regular and graph languages, decidability of the model-checking problem for infinite-transition systems, and the connection of variants of tree automata to monadic second order logic. As a practitioner, I would consider these to be *applications within theory.* However, this monograph also considers applications to *real-world problems.* These concern navigation in XML and type checking of XSLT, system verification, coordination in distributed systems, expression of concurrency, compression of tree-structured data and digital images, and access control in smartcard systems. I am happy to note that many of the contributors to this volume are leading experts in the areas mentioned above.

Dear Reader, have you noticed how many names of Turing Award winners were mentioned in this short preface? If not, here is the list: Rabin, Scott, Pnueli, Sifakis, Clarke, Emerson. This at least supports my claim that automata theory is considered a core part of computer science.

I congratulate the editors and the Indian Institute of Science for this fine collection of articles on modern applications of automata. I am happy to learn that the institute has recently celebrated its centenary, and that this volume is among the books published to mark this event.

Reinhard Wilhelm
Universität des Saarlandes.

Preface

Automata theory is perhaps one of the oldest and most researched areas in Computer Science. Over the past fifty years or so, numerous applications of automata have been developed in a wide spectrum of areas with a corresponding evolution of a variety of theoretical models. Early applications of automata theory included pattern matching, syntax analysis and software verification, where elegant theory was applied to real world problems, resulting in the generation of useful software tools significantly in the area of compilers. Deep connections between automata and logic continue to be discovered, and newer models of automata, for example, timed automata, hybrid automata, distributed automata and weighted automata have been proposed, all driven by specific applications. The main focus of this book is on verification which is, without a doubt, the crowning achievement of the area in recent years. Since a comprehensive coverage of applications in this area is not possible we have selected a few topics that are representative of recent trends in verification. In addition, a couple of chapters on the application of automata to the problems of image and tree compression have been included.

The mathematical prerequisites for understanding the contents of this book are relatively modest (mainly undergraduate courses on automata theory, formal languages and logic). Introductory chapters on advanced topics not typically covered in undergraduate courses have been added. We hope that the material in this book will be useful for an advanced course in automata theory and its applications.

The material in the book has been organised into four parts, based upon the kind of applications of automata discussed. We begin with the introductory chapters, followed by applications related to verification, then applications in logic, and finally applications in compression. We proceed below to describe the chapters in the light of this classification.

Part I comprises the introductory chapters and begins with the chapter *An Introduction to Finite Automata and their Connection to Logic* by Straubing and Weil. Apart from the basic results on finite-state automata like closure properties, the pumping lemma, and the Myhill-Nerode theorem, the chapter covers the important connections to logic, including Büchi's characterisation of regular languages via monadic second order logic (MSO), and the subsequent McNaughton–Schutzenberger characterisation of first-order definable languages. The chapter also introduces the algebraic view of regularity and the notion of the syntactic monoid of a language.

Chapter 2 on *Finite-State Automata on Infinite Inputs* by Mukund introduces automata that run on infinite words. Originally proposed by Büchi in 1960 to decide the truth of MSO-definable properties of discrete linear orders, these automata play an important role in automated verification, particularly with the development of temporal logic as a formalism for specifying and verifying properties of programs. The chapter contains a proof of Büchi's characterisation of the class of languages accepted by these automata in terms of MSO-definable properties, as well as a detailed account of Safra's construction to complement Büchi automata.

Chapter 3 titled the *Basics on Tree Automata* by Löding is a comprehensive introduction to the theory of automata over trees. The foundations of this theory were developed by Thatcher and Wright in the late sixties and by Doner in 1970. The chapter focuses mainly on ranked trees but also describes hedge automata for unranked trees and tree walking automata, the last two being of interest in the context of XML. The author demonstrates that much of the theory of automata over finite words can be adapted to the setting of ranked trees, in particular automata constructions for boolean operations on languages, minimization of automata, and the relation to monadic second-order logic.

Chapter 4 on *An Introduction to Timed Automata* by Pandya and Suman is about automata that run on "timed words," which are classical words along with a real-valued time-stamp for each letter. Timed automata were introduced by Alur and Dill in their seminal paper of 1994 to model and reason about the behaviour of real-time systems. The chapter gives a detailed account of the region construction for showing the regularity of the "untiming" of the language accepted by a timed automaton, closure and non-closure properties, and the undecidability of the universality problem for timed automata.

Moving on to Part II of this volume, we collect together chapters whose common theme is the verification problem. Simply stated, the problem of verification is about modelling and reasoning about the correctness of system models, in terms of satisfying various kinds of properties ranging from unreachability of certain configurations to patterns of behaviour specified by temporal logics. The chapters have been ordered according to the kind of system behaviour they consider, ranging from finite words to continuous dynamics.

This group of chapters begins with Chapter 5 titled *A Language-Theoretic View of Verification* by Lodaya, which views system behaviour as a finite sequence of actions or events, represented as finite words, and discusses the application of formal language theory to solve problems related to the verification of such systems. Chapter 6 by Chaturvedi et al. on *A Framework for Decentralized Access Control using Finite State Automata* continues this view of system behaviour, where system events comprise access requests and grants, and uses classical finite-state automata to implement MSO-based access control policies. In Chapter 7 by Chakraborty on *Reasoning about Heap Manipulating Programs using Automata Techniques*, a program's behaviour is viewed as a sequence of "states" that a program goes through during execution. Each state is modelled as a finite word representing the contents of the program's heap memory. In three different techniques the author explores the use of extensions of classical automata like transducers and counter automata to reason about the set of reachable states of a program. Chapter 8 titled *Chop Expressions and Discrete Duration Calculus* by Ajesh Babu and Pandya considers system behaviour as a sequence of states where each state is a valuation to a finite set of propositional variables. Properties are expressed in the Discrete Duration Calculus (DDC^*). The chapter studies a regular-expression like formalism called chop expressions into which DDC^* specifications can be compiled, enabling one to check equivalence and implication between such specifications.

The next couple of chapters address the verification of distributed systems, where it is convenient to view system behaviour as partially-ordered sets of events. Chapter 9 titled *Automata on Distributed Alphabets* by Mukund introduces asynchronous automata that accept such partially-ordered behaviours represented as "trace-closed" languages of finite words. The chapter contains a first-hand account of the important distributed time-stamping or "gossip" algorithm, which is then used to prove Zielonka's theorem characterising the class of languages accepted by asynchronous automata as trace-closed regular languages. In effect the chapter gives a way to synthesize a distributed finite-state implementation of any regular set of trace-closed behaviours. Chapter 10 on *The Theory of MSC Languages* by Narayan Kumar, is a comprehensive survey of formalisms to generate and reason about Message Sequence Charts which are a popular visual representation of patterns of distributed communication.

From partial-orders we move on to trees. Programs that transform XML documents can be viewed as tree walking transducers that take as input labelled trees of a certain type and output labelled trees of another type. Chapter 11 on *Type Checking of Tree Walking Transducers* by Maneth et al. considers the problem of verifying the type-correctness of such transducers. While the problem in its generality is computationally expensive to solve, the authors provide an efficient solution based on forward type inferencing for a subclass of tree walking transducers.

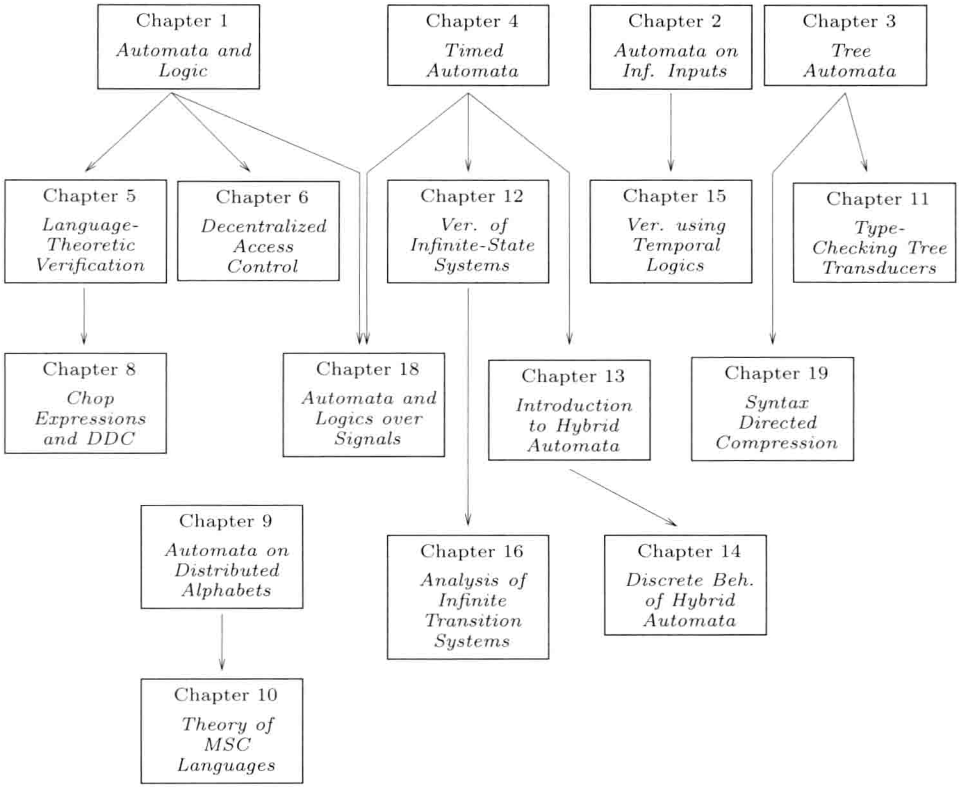
Chapter 12 titled *Three Case Studies on Verification of Infinite-State Systems* by Esparza and Kreiker once again views behaviour as a sequence of states, though now the number of states are potentially infinite, containing for example real-valued valuations for clock variables. Three example systems are considered which are modeled using timed, pushdown, and linear automata respectively. The properties considered are phrased in terms of reachability of certain states. Algorithms to solve the reachability problem for each of these models are described and then applied to the verification problem for the example systems.

We finally move to behaviours exhibited by continuous dynamical systems. Chapter 13 titled *Introduction to Hybrid Automata* by Gopinathan and Prabhakar, introduces hybrid automata for modelling dynamical systems with multiple modes of continuous evolution. The chapter describes algorithms for verifying reachability properties for the well-known subclasses of initialized rectangular hybrid automata and o-minimal hybrid automata. Chapter 14 by Agrawal et al. titled *The Discrete Time Behavior of Restricted Linear Hybrid Automata* studies the discrete time behaviour of a class of hybrid automata with a restricted form of linear dynamics. The authors show that if we consider the discrete time, finite precision semantics of this class of hybrid automata, the resulting language is regular and effectively computable. The results can be used to verify reachability properties of continuous systems modelled as such hybrid automata.

We now come to Part III which groups together material describing applications of automata in logic. Chapter 15 titled *Specification and Verification using Temporal Logics* by Demri and Gastin shows how to solve the satisfiability and model-checking problems for temporal logic. The authors describe in detail an elegant and simple translation from temporal logic to Büchi automata. Chapter 16 by Thomas on *Finite Automata and the Analysis of Infinite Transition Systems* explores an unusual application of automata. The author considers infinite-state transition systems that arise from automata-definable transition relations, and shows that the model-checking problem for these systems with respect to natural first-order and MSO logics is decidable. Chapter 17 titled *Automata over Infinite Alphabets* by Manuel and Ramanujam deals with automata and logics over "data words" which arise in modelling systems with unbounded data. The authors consider a range of automata models over data words with decidable emptiness problems, and use one of these classes of automata to show the decidability of a fragment of an MSO logic interpreted over data words. Chapter 18 by Chevalier et al. on *Automata and Logics over Signals* provides an application of automata to decide satisfiability and characterize the expressiveness of natural logics over finitely-varying discrete-valued functions or "signals."

Finally in Part IV we have two chapters on the application of automata for data compression. Chapter 19 by Shankar on *Syntax Directed Compression of Trees Using Pushdown Automata* describes the compression of ranked and unranked trees using pushdown automata to control the multiplexing of models for adaptive compressors. The first part of the chapter shows how ranked trees defined by a regular tree grammar can be compressed by using a modification of the LR parsing algorithm to achieve multiplexing. The second part describes a scheme for the compression of XML files (which are essentially unranked trees) using devices called recursive finite automata. Chapter 20 titled *Weighted Finite Automata and Digital Image Representation* by Krithivasan and Sivasubramanyam is on the application of weighted finite automata to the problem of image compression. Weighted finite automata were introduced in the nineties by Kulik and Kari for representing images. This chapter describes how such automata can be used in the representation, transformation, and compression of digital images.

We now address the issue of dependencies between chapters. While the chapters are largely self-contained, some chapters do make use of some concepts or results that are dealt with in more detail in other chapters. We have tried to capture some of these essential dependencies in the figure below.



We will be happy to receive feedback from readers on typographical and other errors in this book. A list of errata will be maintained at http://www.worldscibooks.com/compsci/7237.html

Acknowledgments

We owe a huge debt of gratitude to all the authors, many of whom are well-known experts in their areas, for readily agreeing to contribute to this volume, and for having devoted a great deal of their time and energy to present their knowledge, experience, and insights in the chosen topics. We would also like to thank the authors for participating in a careful review of the chapters. We would like to specially thank Reinhard Wilhelm for readily agreeing to write the foreword for this volume. We are indebted to Eugene Asarin, Anca Muscholl, and V. Vinay for their painstaking and useful reviews that have helped to make the material in this book more complete. We would also like to thank Vijay Natarajan and Shantanu Choudhary for their help with generating the image for the book cover.

We would like to express our gratitude to our institute Director P. Balaram for initiating this series of books to commemorate the Centenary of the institute, and for suggesting that we contribute a volume to it. Our thanks are also due to the series editors, Gadadhar Misra and K. Kesava Rao, for their editorial advice. Finally we would like to thank Ranjana Rajan, Kwang Wei, Patrick Tay Zi Dong and the others at World Scientific for their close cooperation in putting together the material in the book.

Deepak D'Souza and Priti Shankar

Contents

Part I

Basic Chapters