PREFACE

This book is an introduction to the theory of computation. After a chapter presenting the mathematical tools that will be used, the book examines models of computation and the associated languages, from the most elementary to the most general: finite automata and regular languages; context-free languages and pushdown automata; and Turing machines and recursively enumerable and recursive languages. There is a chapter on decision problems, reductions, and undecidability, one on the Kleene approach to computability, and a final one that introduces complexity and *NP*-completeness.

Specific changes from the third edition are described below. Probably the most noticeable difference is that this edition is shorter, with three fewer chapters and fewer pages. Chapters have generally been rewritten and reorganized rather than omitted. The reduction in length is a result not so much of leaving out topics as of trying to write and organize more efficiently. My overall approach continues to be to rely on the clarity and efficiency of appropriate mathematical language and to add informal explanations to ease the way, not to substitute for the mathematical language but to familiarize it and make it more accessible. Writing "more efficiently" has meant (among other things) limiting discussions and technical details to what is necessary for the understanding of an idea, and reorganizing or replacing examples so that each one contributes something not contributed by earlier ones.

In each chapter, there are several exercises or parts of exercises marked with a (†). These are problems for which a careful solution is likely to be less routine or to require a little more thought.

Previous editions of the text have been used at North Dakota State in a two-semester sequence required of undergraduate computer science majors. A one-semester course could cover a few essential topics from Chapter 1 and a substantial portion of the material on finite automata and regular languages, context-free languages and pushdown automata, and Turing machines. A course on Turing machines, computability, and complexity could cover Chapters 7–11.

As I was beginning to work on this edition, reviewers provided a number of thoughtful comments on both the third edition and a sample chapter of the new one. I appreciated the suggestions, which helped me in reorganizing the first few chapters and the last chapter and provided a few general guidelines that I have tried to keep in mind throughout. I believe the book is better as a result. Reviewers to whom I am particularly grateful are Philip Bernhard, Florida Institute of Technology; Albert M. K. Cheng, University of Houston; Vladimir Filkov, University of California-Davis; Mukkai S. Krishnamoorthy, Rensselaer Polytechnic University; Gopalan Nadathur, University of Minnesota; Prakash Panangaden, McGill University; Viera K. Proulx, Northeastern University; Sing-Ho Sze, Texas A&M University; and Shunichi Toida, Old Dominion University.

I have greatly enjoyed working with Melinda Bilecki again, and Raghu Srinivasan at McGraw-Hill has been very helpful and understanding. Many thanks to Michelle Gardner, of Laserwords Maine, for her attention to detail and her unfailing cheerfulness. Finally, one more thank-you to my long-suffering wife, Pippa.

What's New in This Edition

The text has been substantially rewritten, and only occasionally have passages from the third edition been left unchanged. Specific organizational changes include the following.

1. One introductory chapter, "Mathematical Tools and Techniques," replaces Chapters 1 and 2 of the third edition. Topics in discrete mathematics in the first few sections have been limited to those that are used directly in subsequent chapters. Chapter 2 in the third edition, on mathematical induction and recursive definitions, has been shortened and turned into the last two sections of Chapter 1. The discussion of induction emphasizes "structural induction" and is tied more directly to recursive definitions of sets, of which the definition of the set of natural numbers is a notable example. In this way, the overall unity of the various approaches to induction is clarified, and the approach is more consistent with subsequent applications in the text.
2. Three chapters on regular languages and finite automata have been shortened to two. Finite automata are now discussed first; the first of the two chapters begins with the model of computation and collects into one chapter the topics that depend on the devices rather than on features of regular expressions. Those features, along with the nondeterminism that simplifies the proof of Kleene's theorem, make up the other chapter. Real-life examples of both finite automata and regular expressions have been added to these chapters.
3. In the chapter introducing Turing machines, there is slightly less attention to the "programming" details of Turing machines and more emphasis on their role as a general model of computation. One way that Chapters 8 and 9 were shortened was to rely more on the Church-Turing thesis in the presentation of an algorithm rather than to describe in detail the construction of a Turing machine to carry it out.
4. The two chapters on computational complexity in the third edition have become one, the discussion focuses on time complexity, and the emphasis has been placed on polynomial-time decidability, the sets P and NP, and NP-completeness. A section has been added that characterizes NP in terms of polynomial-time verifiability, and an introductory example has been added to clarify the proof of the Cook-Levin theorem, in order to illustrate the idea of the proof.
5. In order to make the book more useful to students, a section has been added at the end that contains solutions to selected exercises. In some cases these are exercises representative of a general class of problems; in other cases the

solutions may suggest approaches or techniques that have not been discussed in the text. An exercise or part of an exercise for which a solution is provided will have the exercise number highlighted in the chapter.

PowerPoint slides accompanying the book will be available on the McGraw-Hill website at http://mhhe.com/martin, and solutions to most of the exercises will be available to authorized instructors. In addition, the book will be available in e-book format, as described in the paragraph below.

John C. Martin

Electronic Books

If you or your students are ready for an alternative version of the traditional textbook, McGraw-Hill has partnered with CourseSmart to bring you an innovative and inexpensive electronic textbook. Students can save up to 50% off the cost of a print book, reduce their impact on the environment, and gain access to powerful Web tools for learning, including full text search, notes and highlighting, and email tools for sharing notes between classmates. eBooks from McGraw-Hill are smart, interactive, searchable, and portable.

To review comp copies or to purchase an eBook, go to either www.CourseSmart.com <http://www.coursesmart.com/>.

Tegrity

Tegrity Campus is a service that makes class time available all the time by automatically capturing every lecture in a searchable format for students to review when they study and complete assignments. With a simple one-click start and stop process, you capture all computer screens and corresponding audio. Students replay any part of any class with easy-to-use browser-based viewing on a PC or Mac.

Educators know that the more students can see, hear, and experience class resources, the better they learn. With Tegrity Campus, students quickly recall key moments by using Tegrity Campus's unique search feature. This search helps students efficiently find what they need, when they need it, across an entire semester of class recordings. Help turn all your students' study time into learning moments immediately supported by your lecture.

To learn more about Tegrity, watch a 2-minute Flash demo at http://tegritycampus.mhhe.com