Problem Solving In Automata, Languages, And Complexity

Synopsis

Automata and natural language theory are topics lying at the heart of computer science. Both are linked to computational complexity and together, these disciplines help define the parameters of what constitutes a computer, the structure of programs, which problems are solvable by computers, and a range of other crucial aspects of the practice of computer science. In this important volume, two respected authors/editors in the field offer accessible, practice-oriented coverage of these issues with an emphasis on refining core problem solving skills.

Book Information

Hardcover: 408 pages

Publisher: Wiley-Interscience; 1st edition (September 15, 2001)

Language: English

ISBN-10: 0471439606

ISBN-13: 978-0471439608

Product Dimensions: 6.4 x 1 x 9.4 inches

Shipping Weight: 1.5 pounds (View shipping rates and policies)

Average Customer Review: 2.5 out of 5 stars 5 customer reviews

Best Sellers Rank: #477,299 in Books (See Top 100 in Books) #211 inÂ Books > Science & Math > Mathematics > Pure Mathematics > Logic #1692 inÂ Books > Business & Money > Management & Leadership > Decision-Making & Problem Solving #1848 inÂ Books > Textbooks > Computer Science > Programming Languages

Customer Reviews

"Its notional richness, while challenging, when combined with the instructional narrative is quite engaging-I found myself drawn into the textâ |" (IEEE Circuits & Devices, July/August 2004) "...uses a problem-solving approach that makes these computer languages concrete..." (SciTech Book News, Vol. 25, No. 4, December 2001)

A practical introduction to essential topics at the core of computer science Automata, formal language, and complexity theory are central to the understanding of computer science. This book provides, in an accessible, practically oriented style, a thorough grounding in these topics for practitioners and students on all levels. Based on the authors' belief that the problem-solving approach is the most effective, Problem Solving in Automata, Languages, and Complexity collects a rich variety of worked examples, questions, and exercises designed to ensure understanding and

mastery of the subject matter. Building from the fundamentals for beginning engineers to more advanced concepts, the book examines the most common topics in the field, including: Finite-state automata Context-free grammars Turing machines Recursive and recursively enumerable languages Computability theory Complexity classes NP-completeness Focused, practical, and versatile, Problem Solving in Automata, Languages, and Complexity gives students and engineers a solid grounding in essential areas in computer science.

This book has some of the worst explanations every. Symbols used are very inconsistent and the examples are more confusing than the explanations. There are no bold or italicized fonts to help distinguish between sections and titles of topics for the chapter. There are grammatical errors as well. ONLY REASON TO BUY THIS BOOK IS IF YOU'RE FORCED TO GET IT FOR A CLASS.

Ding-Zhu Du was actually a professor of mine, for undergraduate Automata Theory. This is a thorough text concerning some very advanced concepts in computer science. However, it is overly technical and anyone who is not a graduate student studying university-level computer science will find this COMPLETELY inaccesible. Almost no attempt is made to ease the reader into this extremely complex material. There are typos and grammatical errors, but they are not frequent or major enough to impair the reader's understanding of the what is being said. In truth the actual english text in this book is sparse anyway and I sincerely doubt most readers would have understood the descriptions of the material any more clearly had these print mistakes not existed. Example problems are abundant, but explanations of solutions are short, to the point, and rarely comprehensible. As I've stated already, this material is very abstract and complex, and as such these problems BEG for clear, step-by-step descriptions in plain english. I can gaurentee that any undergraduate or self-taught student will be utterly lost in the myriad of symbols and lemmas filling these pages. Chapter introductions are typically a few short sentences and give no perspective as to just what the hell is really going on in the examples. As a companion to solid professional instruction in this material, this text serves its purpose I suppose. I pity any student who misses a lecture and expects this text to fill in any questions he may have.Honestly, unless you are a professor or engaged in some high-level research, you will gain nothing from this text. As a fourth-year CS student at an accredited state university, I found it intimidating and unreadable (this coming from someone who is well acquainted with needlessly complex mathematical jargon). If you have a legitimate reason to be exploring all the intricate complexities of NP-complete problems, and you already have a solid familiarity with finite automata and computational complexity, you may

want to pick this book up. The hundreds of examples may be worth something to a person who is already familiar with these types of problems. If, on the other hand, you have no idea what the hell a nondeterministic turning machine is, you'd better turn your attentions somewhere else. This book isn't going to be the one to explain it to you.Let me be clear though, this is NOT A BAD TEXT. Rather, due to the extremely difficult technical nature of this material, the text should have been presented at a much slower pace, descriptions of problem solutions should have been much clearer and more detailed, and the text overall should've been much less technical and more approachable if it were indeed meant as an introduction to these concepts for undergraduates. Or, it should simply not be read by undergraduates or self-study students of the discipline.If this is your required text for an undergraduate computer science course, all I can say is, don't skip your lectures.

I like this book and wished that I had more time studying it. There were a couple of things that I thought the authors might want to consider improving:o Maintain consistencies for notations and symbols across the entire booko Use larger fonts and leave some more space for writing comments for the booko Simplify some of the zero block, blank, and BBB notations for Turing machines; they seem not that necessary for us to understand Turing machineso Provide hints for the problem/exercise setI was in awe when learning from the chap 4 and 5, which laid the foundation for NP problems. Turing is such a genius. Strong points of this book:o Math: Though sometimes I felt there was a lot more to be covered for a complete understanding. For example, bounded and unbounded minimizations and the operators.o Abundant examples and brief proofsThings the authors/instructors might want to add or improve:o All math related to this text. List right after each section for the reader to review or catch up. For example, cardinality, math logic, etc.o Explain how parsing and parser generators are actually done. I like LR or LALR parser and grammars better than the LL-like grammar in the texto Structure the proofs to allow better understanding and grasp of the concept.o Indicate clearly when the notation is for set, function, etc.o May be more for Chap 7 and beyond, which was not covered in the class

This book was terrible! It was very boring, overly-technical, and had spelling and grammar errors all over the place! Not very professional... If you are a self-teaching programmer I would recommend that you find a different book because you won't learn much from this one!

I have never seen such a poorly written book in any CS subject. Everything from the way the material is presented to the font chosen for the book are the worst of the worst.

Problem Solving in Automata, Languages, and Complexity CRITICAL THINKING: A Beginner's Guide To Critical Thinking, Better Decision Making, And Problem Solving ! (critical thinking, problem solving, strategic thinking, decision making) Clinical Problem Solving in Orthodontics and Paediatric Dentistry, 2e (Clinical Problem Solving in Dentistry) Clinical Problem Solving in Orthodontics and Paediatric Dentistry - E-Book (Clinical Problem Solving in Dentistry) Clinical Problem Solving in Periodontology and Implantology, 1e (Clinical Problem Solving in Dentistry) An Introduction to Formal Languages and Automata, 5th Edition Introduction to Automata Theory, Languages, and Computation (3rd Edition) Introduction to Automata Theory, Languages, and Computation (2nd Edition) Groups, Languages and Automata (London Mathematical Society Student Texts) An Introduction to Formal Languages and Automata Simply Complexity: A Clear Guide to Complexity Theory The Atlas of Languages: The Origin and Development of Languages Throughout the World (Facts on File Library of Language and Literature)**OUT OF PRINT** Learn Languages & Spanish, 2 Books in 1!: Learning Languages and Learn Spanish The Languages of Tolkien's Middle-Earth: A Complete Guide to All Fourteen of the Languages Tolkien Invented Paper Models That Move: 14 Ingenious Automata, and More (Dover Origami Papercraft) (English and English Edition) Guide to Law and Literature for Teachers, Students, and Researchers: Companion Text to Literature and Legal Problem Solving : Law and Literature As Ethical Discourse Playing Their Parts: 19th Century Automata, Musical Boxes and Singing Birds The Wonderland Of Music Boxes And Automata Cabaret Mechanical Movement: Understanding Movement and Making Automata Automata and Computability (Undergraduate Texts in Computer Science)