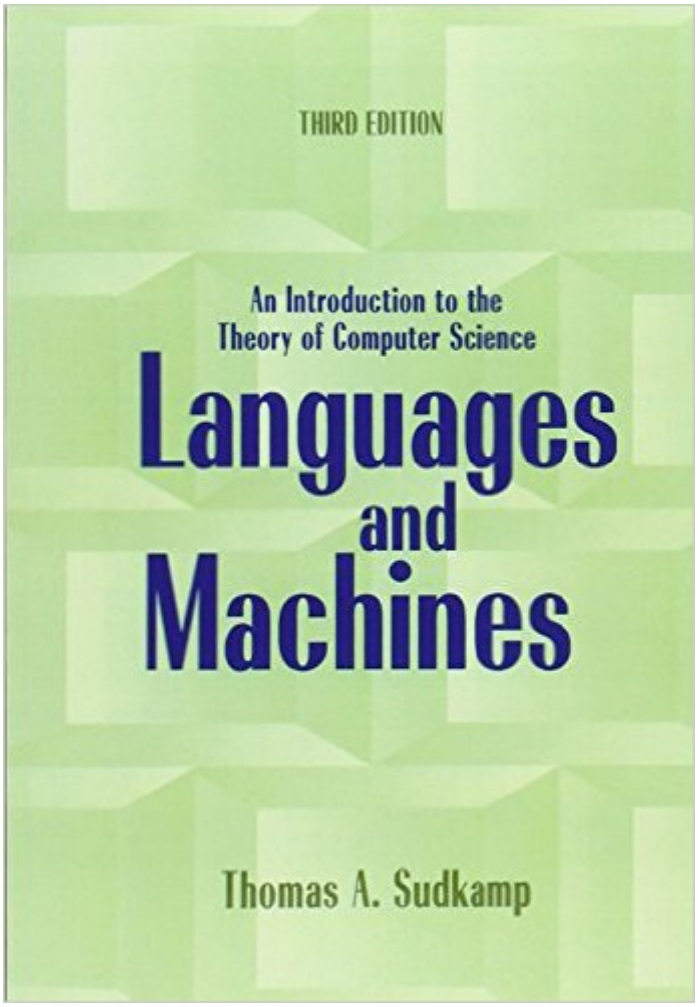
Languages And Machines: An Introduction To The Theory Of Computer Science (3rd Edition)



Synopsis

The third edition of Languages and Machines: An Introduction to the Theory of Computer Science provides readers with a mathematically sound presentation of the theory of computer science.Â The theoretical concepts and associated mathematics are made accessible by a "learn as you go" approach that develops an intuitive understanding of the concepts through numerous examples and illustrations.Â

Book Information

Paperback: 672 pages

Publisher: Pearson; 3 edition (February 24, 2005)

Language: English

ISBN-10: 0321322215

ISBN-13: 978-0321322210

Product Dimensions: 6.4 x 1.4 x 9.1 inches

Shipping Weight: 1.6 pounds (View shipping rates and policies)

Average Customer Review: 3.3 out of 5 starsÂ Â See all reviewsÂ (18 customer reviews)

Best Sellers Rank: #254,954 in Books (See Top 100 in Books) #86 inÂ Books > Computers & Technology > Computer Science > Information Theory #136 inÂ Books > Science & Math > Mathematics > Pure Mathematics > Logic #1794 inÂ Books > Computers & Technology > Software

Customer Reviews

This is one of the better books that I read on languages and machines. This book is great for someone who is interested in parsing, compilers or pattern matching. The book covers a lot of theory on computation and is not for a beginner. I would recommend that one be well grounded in set theory, recursion and mathematical induction before attempting to read this book. I did not read all the chapters; I only read those that were relevant to my project and I had not seen before in other texts. The 1st chapter get you upto speed with a good review of set theory followed by a quick review of induction and recursion. The 2nd chapter gives an excellent introduction to strings, languages and regular expressions along with relations on regular expressions. Chapter 3 is where the rubber hits the road. It covers context-free and regular grammars. I feel this chapter covers the subjects very well. Chapter 4 gives a good description of parsing and methods of parsing. Chapter 6 covers Finite Automata. This chapter describes deterministic finite state machines, nondeterministic finite state matchines and nondeterministic finite state matchines with lambda transitions. The presentation of the subject in this chapter was excellent. Chapter 7 presents Regular Languages

and Sets. This chapter gives a good presentation of how to put together different types of machines from different languages and build languages from machines. I found it best not to read the chapters in orders, instead I read them in the following order which helped to understand the material better; 1,2,6,7,3,4,11,12My only complaint: It would have helped if the author could have gave answers to some of the problems at the end of the chapters.

Abstract language theory is hard, but Languages and Machines does a very good job of explaining the subject step by step. The topics are covered extremely thoroughly and with just the right amount of rigor. As for those who claim it’s not exciting enough, you can’t get blood out of a stone. Only the most dedicated computer scientist and mathematicians will find this topic interesting. Even so, this book does a superb job of tying theory to application (e.g., the machines one can use language theory to build) for even the most obscure concepts (like the Greibach Normal Form).That being said, there are a few problems. First, the author’s claim that this is a book for undergrads is not credible (except perhaps at MIT or CalTech). Even my graduate students have to read sections multiple times to "get it". Second, the author needs to provide solutions to selected problems at the back of the textbook. Most theory books do this, but not this one. This is a major weakness, especially given the difficulty of the material. Lastly, Sudkamp’s proofs are extremely dry and very difficult to follow. He should take a cue from Sipser’s "Intro to the Theory of Computation" book (which I do not recommend as it is generally too abstract for most students) and introduce "proof ideas" to give the big picture for important proofs.

Hey,I was fortunate enough to learn this course from the author of the book. The book by itself might seem tough. The fault lies in the fact that subject matter is not altogether too simple to understand without someone teaching it to you!With the help of the instructor, we did learn a lot about formal languages, finite automaton, regular grammer, etc.The key to understanding this material (and using this book effectively) is solving as many problems as possible, preferably in a group setting so that solutions can be discussed.Note: For most problems, there exists multiple solutions, and the approach is what needs to be learned and discussed.Recommended, with some reservations...Good luck!

This is a tough book. The book is a great resource to utilize, but there are a distinct lack of examples. You’ll absolutely need to supplement this book with something else--educational videos online are a great resource.I used this book in a Theory class, and it was really difficult. Going from

DFAs to NFAs to PDAs were pretty simple, but the jump to Turing Machines was incredibly difficult... I had trouble learning it from this book and had to find other places.If my teacher did not assign problems from the book, I would not have not have bought it. I don't think it impacted how I learned the subject in any way, so save your money if it's an optional buy.

Besides the fact that the book is "dry", in which most Math theory based books are, the examples are just the basis step towards solving a problem. I equate it to teaching a child how to add, and just giving them the example "1 + 0", then assume they can figure out the rest. There are no answers, either in the back of the book for particular exercies, nor was a study guide made available. What is really shocking is that it's the most expensive book out there! Not to mention that there isn't any programming steps made available. Great text for a Math major ... horrible textbook for Computer Science Majors, mainly because computer science majors would want to see programming examples and may not be as strongly math oriented as a Math major would be.

Through OpenGL: From Theory to Experiments (Chapman & Hall/CRC Computer Graphics, Geometric Modeling, and Animation) Finite State Machines in Hardware: Theory and Design (with VHDL and SystemVerilog) (MIT Press) Computer-Controlled Systems: Theory and Design (3rd Edition) Mathematics and Computer Science in Medical Imaging (Nato a S I Series Series III, Computer and Systems Sciences) Hacking: Beginner to Expert Guide to Computer Hacking, Basic Security, and Penetration Testing (Computer Science Series) Computer Analysis of Images and Patterns: 7th International Conference, CAIP ’97, Kiel, Germany, September 10-12, 1997. Proceedings. (Lecture Notes in Computer Science) Computer Analysis of Images and Patterns: 8th International Conference, CAIP’99 Ljubljana, Slovenia, September 1-3, 1999 Proceedings (Lecture Notes in Computer Science)