From: Anderson, Claude W
Sent: Monday, May 16, 2016 4:39 PM
Subject: MA/CSSE 474 summer textbook, prerequisite background, pre-course reading, and other issues

Automata, Computability and Complexity: Theory and Applications

Elaine A. Rich

ISBN-10: 0132288060
ISBN-13: 9780132288064
Publisher: Prentice Hall
Copyright: 2008
Format: Cloth; 1120 pp

I think told all of you whom I talked with in person (and sent an email to all math and CSSE majors in March) that the course is mainly going to consist of self-paced (but with deadlines to make sure you don't save most of the work until the end and I don't have to grade most of the work at the end) reading the textbook, doing the problems (with help from me by email), and getting feedback/grades from me. My PowerPoint slides from last time I taught the course will be available, I have done some videos, and I hope to do some more.

I will make some changes, but many of the assigned problems (especially at the beginning of the course) will be the same as last time I taught the course; the HW from last time should be a good starting place **if you want to work ahead in the course**. In case you want to work ahead, I will say that all of the problems from 2015-16 Winter's HW 1 and HW2 (http://www.rose-hulman.edu/class/csse/csse474/201620/Schedule/Schedule.htm)will be part of the Summer's assignments (however, I may rearrange the assignments or add some new problems). Details of how the course is presented (and of course the syllabus) may be very different, but the reading material and early problems from this past winter term will be relevant to the summer course.

About the course
I think this is a very enlightening course. As the name implies, it is a theory course. There are many applications of this theory, and I chose a textbook that talks about many of them, but the theory itself will be the emphasis of this course.

I believe that this course that is required for CS majors has some things in common with the Reals course that is required of math majors:

* It is where you REALLY learn to do complex proofs (or you die!).
* A high percentage of the HW and exam problems involve doing proofs.
* Some of the theorems from the course are counter-intuitive.
* Notation and terminology are crucial to understanding and communicating about the material.
* It is probably the most intellectually challenging course in the CSSE department.

It is no accident that this course can be taken with a MA number or a CSSE number. It really is a mathematical study of computation. How do we model computation mathematically? Given a specific model, what kinds of computations can be done, and which ones can't be done?

Expected student background
The main background for this course is from the two discrete math courses (MA 275 is probably more important than 375). For some of you, it has been a long time since you took those courses, or you did not get that material very well when you took them.

The Rich book has extensive background help for you in Appendix A (Sections 1-7, about 45 pages). Appendix A is a review of the Mathematics needed to understand this textbook. I suggest that you read it carefully before the course starts. Almost all of this appendix should be things that you saw in Disco I & 2 courses. But there may be some things that were not emphasized by your instructor or that did not stick in your long-term memory. And this appendix is good way to to get into this author's use of terminology, etc. before reading the book. Pay special attention to the section on proof techniques.

The contents of that appendix are approximately the background that I expect you to have as you come into this course. We will spend a little bit of time at the beginning of the course reviewing some highlights of Appendix A.

You will be doing a number of proofs, including proofs by induction. If your previous courses did not bring you to a high comfort level with writing inductive proofs, I especially recommend that you work on that before the course begins. Many of the other proof techniques from Appendix A will be useful for the course also, so you should review all of them.

If your background may be weak (for example, if you did not get at least a B in MA275 and MA375 (or if you did not take MA375 yet) you probably ought to get a head start on reviewing for this course as soon as the Spring term is over. Because of the flexible schedule of the course, it will be okay if that review takes you a little bit past the first official day of the course, June 3.

To give you an idea of how much reviewing you need to do in order to be able to get a good start in the course, list some of the topics from Appendix A. If a *few* of them are fuzzy or unfamiliar to you, a little extra work after the term starts should suffice to catch you up. If a *lot* of them are fuzzy or unfamiliar, you really should carefully read this appendix before the term begins.

- Logic
    - Boolean propositional logic
        - Well-formed formulas and propositions
        - Truth tables
        - Axioms and proofs
        - Modus ponens, modus tolens
    - First-order logic
        - Predicates, terms, expressions, free and bound variables,
        - Universal and existential quantifiers
        - Interpretations and models; valid, satisfiable, and unsatisfiable formulas
        - Quantifier exchange, universal instantiation, existential generalization Sets
    - Enumeration of a set
    - Finite, countable, and uncountably infinite sets
    - Subset, intersection, union, difference, power set
    - Partitions of a set
- Relations
    - Cartesian Product of two sets
    - Inverse of a relation, graph of a relation
    - Reflexive, transitive, symmetric, antisymmetric, equivalence relation, equivalence classes
    - Orderings and partial orderings
- Functions
    - Domain, range, arity, total and partial functions
    - Commutativity, associativity, distributivity, identity, inverse elements
    - One-to-one and onto functions
- Closure
    - What it means for a set to be closed under a property
    - Transitive and reflexive closures
    - Closure under functions
- Proof techniques.
    - Proof by:
        - Construction
        - Contradiction

Counterexample
Case Enumeration
Mathematical induction
Pigeonhole principle
Showing that two sets are equal
Showing that a set is finite or countably infinite
Diagonalization: Showing that a set is uncountable (this is not a prerequisite for 474) Reasoning about programs
Proving correctness (not a 474 prerequisite)
Analyzing complexity (big-O and its cousins)

You should also read pages xii-xv in the preface. First actual reading assignment after the course starts: Chapters 1 and 2 (both are very short).

More details about course requirements, due dates, etc. will be available on or before June 3.

I look forward to working with you this summer.

If you have any questions, feel free to email me or come to my office.

Claude Anderson

Return policy. Product Highlights. Automata, Computability and Complexity: Theory and Applications See More Info. Tell us if something is incorrect. About This Item. We aim to show you accurate product information. Manufacturers, suppliers and others provide what you see here, and we have not verified it. See our disclaimer. Hardcover, Prentice Hall, 2007, ISBN# 0132288060 Automata, Computability and Complexity: Theory and Applications. Specifications. Publisher. Algorithms, Complexity Theory and Optimisation. Economics and Computation. Networks and Distributed Computing. Artificial Intelligence. Research Impact.Â Our group works on fundamental questions about what can be computed in principal (computability theory) and what amount of computational resources such as time and space are required to perform those computations (computational complexity theory). The study of abstract machines and automata plays essential role in theoretical analysis of computations and therefore is important integral part of this stream of research. Computational complexity theory focuses on classifying computational problems according to their inherent difficulty, and relating these classes to each other. A computational problem is a task solved by a computer. A computation problem is solvable by mechanical application of mathematical steps, such as an algorithm. A problem is regarded as inherently difficult if its solution requires significant resources, whatever the algorithm used. The theory formalizes this intuition, by introducing