CSC 544 Fall 2022

Theory of Computation

Prof. Noah M. Daniels

noah_daniels@uri.edu
http://www.cs.uri.edu/~ndaniels/
Tyler 250

Changes to course policies or schedule may occur in response to unforeseen circumstances. I will notify the class of any changes immediately.

Course Description: What's this class about and why am I taking it?
To me, Theory of Computation represents the greatest mathematical achievement of the 20th century, and continues into the 21st with important open questions around P vs. NP (and more generally, the polynomial-time hierarchy), randomness, and quantum computing. I consider this class to be firmly at the intersection of computer science, mathematics, philosophy, and even linguistics.
Prerequisite(s): CSC440 (Design & Analysis of Algorithms) or equivalent.
Texts:

- *The Nature of Computation*, Cris Moore and Stephan Mertens (recommended)
- *Quantum Computing since Democritus*, Scott Aaronson (recommended)
- *Computability and Logic, fifth ed.*, George S. Boolos, John P. Burgess, Richard C. Jeffrey (recommended)

Should you buy the books? Moore and Mertens is an excellent foundation, though the earlier chapters are more appropriate to an undergraduate algorithms course (e.g. CSC 440). Still, it is an excellent coverage of the theory of computation, complexity theory, and even some areas that might be of interest even though we may not touch on them this semester (such as spin glasses and other things motivated by physics). The authors are physicists, and yet this book is one of the two best introductions to theoretical computer science I have found. It is one of the few hardbound textbooks that I have found ***truly enjoyable*** to read.
Scott Aaronson's book is quite a fun read, though it somewhat glosses over the more foundational material in order to present a notion of quantum probability and introduce quantum computing. I will teach some of the material from this book, but will not assign reading from it; you might enjoy it, however.
Boolos, et al. is a good foundation, but it lacks any mention of other important models of computation, namely Alonzo Church's Lambda Calculus, and Emil Post's rewrite rules. It also lacks in-depth coverage of the polynomial-time hierarchy and P vs. NP.
Rather than overly burden you with textbooks, we will also read a variety of papers that I will make available.

Course Goals:

1. Become conversant in formal models of computation: Turing machines, the Lambda calculus, Recursively Enumerable functions, and Post tag machines
2. Obtain fluency in the terminology of computation theory
3. Know some of the history of theory of computation and its connections to other disciplines

At the end of this course you will have an understanding of the jargon and terminology of computational theory. You will have an understanding of the major models of computation and complexity theoretical issues. You will feel comfortable writing proofs involving computational models.

Student Learning Outcomes:

Upon successful completion of this course of this course, students will be able to:

1. Write proofs involving computational models
2. Write an automaton to recognize a grammar
3. Prove the computational complexity of a problem by way of a reduction

Grade Distribution:

Class Participation	40%
Homework Assignments	60%

Letter Grade Distribution:

>= 93.00	A	80.00 - 82.99	B-	67.00 - 69.99	D+
90.00 - 92.99	A-	77.00 - 79.99	C+	63.00 - 66.99	D
87.00 - 89.99	B+	73.00 - 76.99	C	60.00 - 62.99	D-
83.00 - 86.99	B	70.00 - 72.99	C-	<= 59.99	F

Course Policies:

- **Attendance**
  - You are expected to attend class. I do not take attendance *per se*, but in-class discussion is an important component of your grade in the form of class participation.
- **Grades**
  - We will use the Perusall web platform for reading and discussion of assigned papers. Discussion here will be a component of class participation.
  - **Homework assignments** will be submitted using the Gradescope online system.
- **Cheating**
  - All homework solutions must be your own work.
  - While you may discuss general solutions and approaches with classmates (indeed, this is encouraged) your writeup must be your own.
  - **Any violation of these rules may result in a grade of 0 on the assignment. In addition, you may be reported to the Dean and the Office of Student Life.** See the University Manual for more information about the potential consequences of cheating `https://web.uri.edu/manual/chapter-8/chapter-8-2/`.

- **Exams**
  - There will be no exams. Your grade is based on homework and class participation (including paper discussions).

Students with Disabilities

Any student with a documented disability is welcome to contact me as early in the semester as possible so that we may arrange reasonable accommodations. As part of this process, please be in touch with Disability Services for Students Office at 302 Memorial Union, Phone 401-874-2098.

Academic Honesty Policy:

All submitted work must be your own. If you consult other sources (class readings, articles or books from the library, articles available through internet databases, or websites) these MUST be properly documented, or you will be charged with plagiarism and will receive an F for the paper. In some cases, this may result in a failure of the course as well. In addition, the charge of academic dishonesty will go on your record in the Office of Student Life. If you have any doubt about what constitutes plagiarism, visit the following websites: the URI Student Handbook, and Sections 8.27.10 – 8.27.21 of the University Manual (web.uri.edu/manual/).

Programming assignments will be done in pairs. For the purposes of pair programming assignments, "your own" means that the work is the product of you and your partner, together.

Attendance

Students are expected to attend class and classroom activities. Occasionally, students may miss class activities due to illness, severe weather, or sanctioned University events. If ill, students should not attend class and should seek medical attention especially if they have a communicable disease. Students should not attend class when the University announces classes are cancelled due to severe weather. Also, it is the policy of the University of Rhode Island to accord students, on an individual basis, the opportunity to observe their traditional religious holidays. Students desiring to observe a holiday of special importance must inform each instructor and discuss options for missed classes or examinations. See Sections 8.51.11 – 8.51.14 of the University Manual for policy regarding make-up of missed class or examinations.

Tentative Course Outline:

Note: This is organized by **module**, not by week. Invariably, the schedule will change. This schedule roughly follows a historical path, beginning with late 19th century mathematical logic, journeying through the formalization of computation in the 1930s and 1940s, and culminating in modern complexity theory and quantum computing.

- **Mathematical foundations: Frege, Russel, Peano, Hilbert**

  Sets, Axioms, and Arithmetic

- **Gödel**

  Incompleteness

- **Finite Automata**

  Generators and Recognizers, Grammars

- **Turing**

  Turing Machines, the Halting Problem, Undecidability

- **Péter**

  Recursion (Primitive, Total, Recursive Enumerability)

- **Church**

  The Lambda Calculus

- **Post**

  The Tag Machine

- **Cook, Levin, Karp**

  Complexity, the Polynomial-Time Hierarchy, P vs. NP

- **Markov, Chernoff**

  Randomness

- **Feynmann, Shor, Aaronson**

  Quantum Probability and Computation