CS081: Computability and Logic

Instructor: Joe Vanderwaart (Andrew 263/x70486/`joev@cs.pomona.edu`)

Text: James L. Hein, *Discrete Structures, Logic, and Computability, Second Edition,* 2002.

Lectures: Tuesday and Thursday, 1:15–2:30, Millikan 218

Office Hours: Tuesday 2:30-4:30, Wednesday 4:00-5:00 and by appointment.
(In other words, if you warn me you're coming I can probably be there.)

Final Exam: Tuesday, December 13 at 2:00 p.m.

Catalog Description

> An introduction to formal systems, mathematical models of machines, and computability. Topics include predicate logic, regular languages, context-free languages, and recursive and recursively enumerable sets. Students will learn to understand and construct formal proofs.

Course Overview

One of the remarkable things about theoretical computer science is that its foundations were laid by mathematicians and logicians investigating the limits of mechanical calculation long before anyone knew how to build calculating machines. What is even more remarkable is how useful the abstract models of computation those mathematicians studied are in computer science today, especially considering how much they appear to differ from the digital electronic computers we are familiar with. This course will explore some of these formal notions, their properties, and their applications.

Our investigation will focus first on a very basic computational problem: deciding whether a given string is a member of a given infinite set or not. We'll learn some ways to describe infinite sets of strings, and see some abstract machines for testing membership, called *automata*, that go along with them. Eventually we will discover a kind of automaton (the *Turing machine*) that seems to be able to solve every problem that we solve with *any* machine, including real computers. Even Turing machines have their limitations, however, and we will see several examples of problems that no Turing machine, and thus no known form of calculating machine, can solve.

Another way of approaching the question of how much mathematics a machine can do is to look at how well machines can handle logical reasoning. The second part of the course will be spent on the study of logic, where statements and proofs, normally the very language of mathematical discourse, become objects of discussion in their own right. After formalizing the structure of logical statements, we will ask what it means, formally, for a statement to be true, and give a formal description of proofs. We will then be in a position to answer questions like whether every true statement is provable (yes, in a certain sense) and whether there is an algorithm to test whether a statement is true (no).

Calculation of Grades

Final grades will be based on the weighted average of homework grades, exam scores, and class participation, as follows:

- Homework, 35%
- Midterm Exam, 30%
- Final Exam, 30%
- Participation, 5%

Letter grades will be determined from this numerical score, but the range of scores corresponding to each letter will depend on the distribution of scores in the class. At least one student will get an A or an A+, and at least one will not.

Homework Policies

There will be weekly homework assignments, due on Thursdays. Homework must be legible; if I can't understand it, I can't give you credit for it. Using a computer and your preferred document preparation software to write up your homework is probably the most reliable way of ensuring legibility, but the choice is yours.

Homework is due at the start of class on the due date; students arriving late to class may hand in their homework on their arrival with no penalty. Any homeworks received after the end of class, however, will be penalized by multiplying the numerical score by $(0.8)(0.9)^n$, where n is the number of calendar days it is late. Thus the maximum score possible on a homework submitted after class on Thursday is 80%, the maximum for homework submitted on Friday is 72%, and so on.

Collaboration on homework assignments *is allowed*, subject to the following guidelines:

- Anything handed in for credit must be the primarily the work of the student submitting it. Sharing complete or nearly complete answers with fellow students is expressly prohibited, as is receiving answers from fellow students or any other source.
- Each student who discusses the content of a homework assignment with one or more others must acknowledge those others by name on the writeup he or she hands in.
- I reserve the right to ask particular groups of students not to collaborate with each other anymore.

This policy is subject to changes and exceptions; any rules specified in a given homework assignment are the ones that count for that assignment.

Projected Schedule

This entire schedule is tentative. All readings are from the textbook; "to" means "up to and including".

Date	Lecture #	Topic	Reading
8/30	1	Introductions; logistics; course overview; alphabets and strings	
9/1	2	Inductive definitions and proofs; basic string operations; languages; language operations; regular languages	1.3.3, 3.1
9/6	3	Regular expressions; Deterministic Finite Automata; DFA's and regular languages	11.1, 11.2.1
9/8	4	Nondeterministic Finite Automata; NFA's and regular languages; DFA's and NFA's	11.2.2–11.3.2
9/13	5	Applications of regular languages	11.2.6
9/15	6	Decision problems about regular languages	
9/20	7	Distinguishability, The Myhill-Nerode Theorem	11.3.3
9/22	8	Non-regular languages; pumping lemma for regular languages	11.4.2–11.5
9/27	9	Context-free grammars; context-free languages; regular grammars and NFA's	11.4.1, 12.1
9/29	10	Push-down automata; PDA's and CFG's	12.2
10/4	11	Deterministic CFL's; parsing	12.3
10/6	12	Non-context-free languages; pumping lemma for CFL's.	12.4.2–12.5
10/11	13	Review	
10/13	14	Midterm Exam	
10/18	–	*Fall Recess*	
10/20	15	Turing machines; recursively enumerable languages; recursive languages	13.1.1–13.1.2
10/25	16	Chomsky hierarchy; variations on Turing machines	13.1.3, 14.2
10/27	17	Computable functions; models equivalent to Turing machines; Church's Thesis	13.2 (to 13.2.2)
11/1	18	Undecidability; the halting problem	14 (to 14.1.3)
11/3	19	Reductions; undecidable problems	14.1.4
11/8	20	Introduction to Logic; propositional logic syntax and semantics; natural deduction and proofs	6
11/10	21	Soundness and completeness of propositional natural deduction	
11/15	22	First-order logic syntax; free and bound variables; renaming and substitution	7 (to 7.1.3)
11/17	23	First-order logic semantics; interpretations and models; validity, satisfiability and entailment; theories	7.1.4–7.2.1
11/22	24	First-order natural deduction; soundness and completeness of first-order logic; Is truth decidable?	7.3–7.4
11/24	–	*Thanksgiving*	

11/29	25	Reasoning about equality; higher-order logics	8
12/1	26	Automated reasoning; sequent calculus	9.1
12/6	27	Review	