CS 81 Course Syllabus

Pomona College, Spring 2018

Time/location:	T/Th 9:35-10:50, Millikan 2131
Instructor:	Prof. Yuqing Melanie Wu Office: Edmunds 221 Phone: 909-607-9202 Email: Melanie.Wu@pomona.edu Office hours: M/T/W/Th 14:30-16:00 and by appointment
Student Mentors:	Manon Audebert, Wentao Guo, Alice Tan, Ross Wollman Mentoring hours: Tuesday/Wednesday/Thursday 8-10pm

Table of Contents

CS81 Overview

Course 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.

Prerequisites

CSCI52 PO and CSCI60 HM; and one of the following: CSCI55 PO, Math55 HM or Math103 PO.

Aim and Objectives

Computability refers to the study of the mathematical foundations of computation: what is an appropriate mathematical model of a computer, what types of computations are possible in the model, what types are not, the inherent complexity of certain computations and so forth. Perhaps surprisingly, many concepts from the theory of computation have become of fundamental importance in other areas of computer science, such as computational linguistics, compiler design, hardware design, object-oriented design, artificial intelligence, and even the syntax of the UNIX grep and awk commands.

In this course we will investigate the interaction between various models of computation. Along the way the intimate connection between computation and language recognition will be developed. We will study several classes of abstract machine including finite automata, push-down automata and Turing machines along with several classes of languages such as regular and context-free languages. In addition, we will examine some of those problems, such as the Halting Problem, which are not amenable to computer solution.

We will also investigate the use of formal logics in computer science, defining a variety of logics and examining their strengths and weaknesses in expressiveness and the difficulty of creating proofs. We will examine logics that are particularly useful in proving programs (in both software and hardware) correct.

By the end of this course, you should be able to:

1. Describe and use formal systems to model real phenomena

2. Understand the differences between and use of formal syntax and semantics.

3. Understand and be able to write proofs in several formal logic systems.

4. Understand the importance of the notions of soundness and completeness for formal logics.

5. Be able to use formal logics to prove the correctness of computer programs.

6. Understand the uses of and differences between finite automata, pushdown automata, and Turing machine models of computation.

7. Understand and use the correspondence between formal grammars and the machines that can recognize them.

8. Understand the Church-Turing thesis and several models of universal computation.

9. Understand and be able to show that a variety of interesting problems involving computation are

undecidable.

Course materials

- *Automata, Computability, and Complexity* by Elaine Rich
- *Logic in Computer Science,* 2nd Edition by Michael Huth & Mark Ryan

Course outline

- Topic 1: FSA and Regular Languages
- Topic 2: PDA and Context-free Languages
- Topic 3: Proposition Logic
- Topic 4: Predicate Logic
- Topic 5: Program Logic
- Topic 6: Model Checking
- Topic 7: Turing Machine and Decidability

Grading Scheme

- Assignments:
  - 12 HW assignments
  - 25% of total grade
- Exams:
  - Two 24-hour take-home exams
  - 70% of total grade
- Participation:
  - 5% of total grade
  - Students are expected to actively participate in lectures and discussion.
  - The participation grade is assigned based on the value your participation adds to the lecture as well as your attendance.

Course Schedule

- *Please notice that the schedule may change slightly during the semester based on students' feedback. An up-to-date course schedule will be shared via Sakai.*
- *In the table below, R stands for the Automata textbook by Rich, while HR stands for the logic text by Huth and Ryan.*
- *Readings of the form n.m means section m from chapter n.*

Week	Day	Date	Topic	Details	Readings	Exercises	Deadlines
1	T	1/16	1	Introduction, Finite Automata, NDFA	R 1-4, 5.1-5.3	R 2.7, 4.4, R 5.2 (b,c), 5.3	
1	Th	1/18	1	NDFA, Minimization	R 5.4-5.8	5.5, 5.6(b,c), R 5.8(a), 5.11(a)	
2	T	1/23	1	Regular expressions & grammars, Pumping Lemma	R 6.1-6.2, 7.1-7.2, 8	R 6.2(b,d), 7.1(a,b)	

2	Th	1/25	1	Pumping Lemma, Decision Procedure for regular languages	R 8, 9	R 8.1(a,b,k), R 9.1(a,i)	HW1 due Jan 25, 10pm
3	T	1/30	2	Context-free Grammars, PDA	R 11.1-11.8, R 12.1-12.3	R 11.5, 11.6(d,f), R 12.1(c,f)	
3	Th	2/1	2	PDA, Pumping Lemma	R 12.1-12.3, R 13.1-13.3	R 12.2, 13.1(a,f)	HW2 due Feb 1, 10pm
4	T	2/6	2	Pumping Lemma, closure properties	R 13.4	R 13.4	
4	Th	2/8	2	Algorithms for CFL, Parsing	R 14, 15.1-15.2	R 13.17(a), R 14.1(a)	HW3 due Feb 8, 10pm
5	T	2/13	3	Propositional Logic	HR 1.1, 1.3, 1.4.1	HR 1.4.2(g,h,i)	
5	Th	2/15	3	Natural deduction	HR 1.2	HR 1.2.3(d,f,g)	HW4 due Feb 15, 10pm
6	T	2/20	3	Natural deduction	HR 1.3, 1.4.2-1.4.3	HR 1.2.3(c,d)	
6	Th	2/22	3	Soundness and completeness	HR 1.4.4	HR 1.4.15	HW5 due Feb 22, 10pm
7	T	2/27	4	Predicate Logic, review for exam	HR 2.1-2.2	HR 2.1.4(c,g,h,k)	
7	Th	3/1	4	Proof theory of predicate logic	HR 2.2, 2.3	HR 2.2.4(a.b.c), HR 2.3.11(a)	HW6 due Mar 1, 10pm
8	T	3/6	4	Proof theory of predicate logic	HR 2.4-2.5	HR 2.4.5	Exam 1 Mar 5-6
8	Th	3/8	4	Semantics of predicate logic, Compactness and expressiveness	HR 2.5-2.6	HR 2.5.1(b)	HW7 due Mar 8, 10pm
9	T	3/13		Spring Break			
9	Th	3/15		Spring Break			
10	T	3/20	5	Program logic, Program verification	HR 4.1-4.3	HR 4.2.1, HR 4.3.5(b,c)	
10	Th	3/22	5	Hoare logic, Model Checking	HR 4.3-4.4, HR 3.1	HR 4.3.10	HW8 due Mar 22, 10pm
11	T	3/27	6	Model Checking	HR 3.1-3.2	HR 3.2.4	
11	Th	3/29	6	Model Checking	HR 3.3-3.4	HR 3.3.3, HR 3.4.2(a,f)	HW9 due Mar 29, 10pm
12	T	4/3	7	Turing Machine, TM variants	R 17.1-17.3	R 17.2(a,c), 17.4	
12	Th	4/5	7	Universal TM, Church-Turing	R 17.6-17.7, R18	R 17.14, R 18.1(a)	HW10 due Apr 5, 10pm
13	T	4/10	7	Halting, Decidability	R 19, R20	R 19.1	
13	Th	4/12	7	Decidability, semi-	R 20, R 21.1-	R 20.2(c,e), R	HW11 due

				decidablity, Reducibility	21.3	21.1(f, m)	Apr 12, 10pm
14	T	4/17	7	Reducibility, Rice's theorem	R 21.4	R 21.10	
14	Th	4/19	7	Decidability of language problems	R 22.5 (review R 9, 14)	R 22.7(a,c)	HW12 due Apr 19, 10pm
15	T	4/24	7	Undecidablity	R 21.5-21.7	R 21.20	
15	Th	4/26	7	Applications and other Formal Systems			HW13 due Apr 26, 10pm
16	T	5/1		Review for Exam			
16	Th	5/3		Reading day			Exam 1 May 3-8

- Other important dates
  - Jan 29. Last day to add a course
  - Mar 8. Last day to drop a course
  - Apr 5. Last day to choose pass/no credit grading option.

Course Policies

The information contained herein is not optional. The students are responsible for conducting themselves in accordance with the rules below. Ignorance of a rule stated or referred to here is not grounds for excusal. If you have questions regarding a policy, seek advice from the instructor or student mentors ***BEFORE*** you act.

Academic involvement

- Sakai is used as the resource-sharing platform for this course. Students are expected to consult the board regularly for announcements and updates in reading and assignments. You may want to double check your Sakai setup such that you will receive email notification whenever new accouchements are posted.
- Detailed topic syllabus will be made available via Sakai before we enter each topic. Students are responsible for assigned readings ***PRIOR*** to lectures. You will find lots of examples and discussion of the materials in the textbook. We will not attempt to cover all materials in lecture. Instead, we will cover the highlights, additional examples, and particularly difficulty materials (some not from the text). It is expected that you come to the lecture already familiar with the simpler aspects of the materials so we can work on the difficulty parts together.
- Students are expected to practice materials covered in each lecture using the practice homework listed with each lecture. Those exercises are picked to help you test your understanding of the course materials and to prepare you for the weekly homework assignments. These exercises will not be turned in. We encourage you to work with your classmates to finish these exercises or to check your solutions.

- Course slides will be made available via Sakai before each lecture. Slides are designed to help students take lecture nodes. Students are responsible for downloading and printing the slides for note-taking during lecture.

Academic Honesty

Students should make themselves familiar with the Pomona College Student Code and Academic Honesty Policy of Computer Science Department. Please notice that conducting academic misconduct and facilitating academic misconduct both violate the student code and the academic honesty policy.

In this class the default penalty for any sort of academic dishonesty shall be failure in the course. Please do not put yourself or any of your classmates in this position.

Exams

- Missed exams can ***NOT*** be made up. A student who is late for checking out take-home exam is expected to submit the exam paper at the arrange ending time of the exam.
- Group work is *NOT* permitted on the exams. The exams must be entirely your own work.

Homework Assignments

General guideline

An important criterion in grading HW assignments will be clarity of solution. Thus you should attempt to explain your solutions as clearly as possible. This is especially important for proofs, as we already know the result is true. We are looking to see if you can formulate a complete and correct proof.

Homework assignment must be **individual work** of a team (usually student pair).

- The team is required to work on all problems together. Each team member is required to write part of the solutions.
- Between teams, discussion is encouraged, but please avoid taking any written notes from discussion; do not show or send your edited work or written document to other team; and do not copy from books and other resources.
- Please clearly list all collaboration and references in your submission.

Editing guideline

Homework assignments should be edited using LaTeX and be submitted in .pdf format.

- Please use the homework template, which will be posted via Sakai, when composing your homework submissions. If your document has more than one page, please clearly list the page # at the bottom of the page.
- For HW assignments, please name your file hw*-xxx.pdf, where * is the assignment # (in double digits) and xxx is your last name. For example, I would name my answer to HW 1 as hw01-Wu.pdf. For HW assignment that are to be done in pairs, the name should be hw*-xxxyyy.pdf, with both students' last names in alphabetical order. For example, if I team up with Chris Brown for HW2, the file name should be hw02-BrownWu.pdf.

- To facilitate team grading, you will be asked to arrange every single question on a separate page. For certain questions, answers may spread out on more than one page. In such case, it will be indicated in the assignment how many pages you should reserve this the questions. If you use less pages than indicated, please leave empty pages.
- To make it easier for you to compose our solution, HM assignments will be posted with both LaTeX source code and .pdf file.

Submission guideline

All HW should be submitted via the course gradescope site. All deadlines are strictly enforced.

- The timestamp of gradescope is used as the indicator of the time of submission.
- Each student will have three "late days" during the course of the semester. A one-day delay in submission costs one "late-day". In the case of pair HW, it will cost each partner one "late day". The "late days" are intended to be used for illness and unexpected emergencies, rather than to make up for a late start on your HW. Once the late days are consumed, late submission will not be accepted any more except for extenuating circumstances. Extenuating circumstances will normally include only serious medical or family emergencies documented with a doctor's note or notes from dean's office.

Incomplete grade

- An incomplete ("I") final grade will be given only by prior arrangement in exceptional circumstances conforming to college policy.

Special need & religious observance

- We will follow the policies of Pomona College to accommodate students' special needs and religious practice. Students with such needs should contact the Dean of Students and bring their instructions to the instructor for accommodation.

Academic Honesty Policy

Computer Science Department, Pomona College

Adopted July 1, 2014

The Computer Science Department seeks to create a friendly and supportive learning environment. We encourage students to work in groups to review material from the lectures and readings, to work practice problems from the text, to study for exams, and to discuss the general ideas and approaches to assignments. However, work submitted for a course must be done independently, unless collaboration on a particular assignment is explicitly permitted. Effective learning is compromised when this principle is violated. As explained in the Pomona College Student Handbook, this means that the work you turn in must represent only your own work. It must not be based on help from others or information obtained from sources other than those approved by the instructors.

The following discussion reflects our general understanding of academic honesty in the Computer Science Department. Any exceptions or differences will appear in the course syllabus or the instructions for an assignment. Ask your instructor if you are ever unsure about what constitutes acceptable behavior.

1 Collaboration policies

The types of work and the level of expected collaboration vary from course to course and assignment to assignment. In this section, we describe some typical expectations. Instructors will often indicate that an assignment falls into a particular category, occasionally with additional remarks about the use of specific materials or sources. Students may freely use any resource that is provided by the instructor for an assignment.

1.1 Individual Work

Most work in our courses is to be completed individually. In general, the work that that is submitted for an assignment must be the student's own. Students may not submit work under their own name that is done by, or in collaboration with, someone else. Copying solutions from any source, including the web or students in previous offerings of the course, is not allowed.

Students should not read or possess copies in any form—physical or electronic—of another student's work. There is no legitimate reason for a student to possess a copy of another student's assignment, to send a copy of student work from one computer account to another, or to be logged-on to another student's account. Providing one's own work to another student is also a violation of these policies.

We routinely use software and other tools to detect similarities between submissions. Identical, or nearly identical, submissions will be considered conclusive evidence of plagiarism.

- For programming assignments, students may normally discuss general approaches to assignments, and they may give or receive "consulting" help for specific problems with software or computer programs. A student may look at another student's work only when help is requested. In that situation the student takes on the role of mentor, and the interaction must be limited to the immediate problem. Two students sitting side-by-side and working through a program step-by-step will certainly produce work that will be considered evidence of illegal collaboration.
- On problem sets, group discussion of the general ideas and approaches is permitted, provided the group members are noted on the submitted solutions. However, each student must write the solutions apart from the group, without consulting notes or other artifacts from the discussion.
- Although papers are less common in computer science classes, when they are assigned they must adhere to the usual levels of academic integrity. The prose must be the student's own, and all external sources must be properly cited.

1.2 Group Projects

Sometimes assignments are to be done by small teams of students. In these situations, the team takes on the role of an individual in the preceding discussion. The members of a team may communicate with one another, but collaboration with members of a different team is not permitted.

1.3 Exams and Test Programs

As stated in the Pomona College Student Handbook, "Students neither give nor receive assistance with examinations." Each examination will have a clear statement of what resources are permitted. Any use of material beyond those limits is not allowed. Take-home examinations will have time limits and similarly explicit rules; they are subject to the same policies.

During examinations students may ask the instructor questions of clarification. The instructor will decide how complete an answer can be given.

Some courses have "test programs" which are programming assignments that are to be treated in the same way as take-home examinations.

1.4 Practice Exercises

Some assignments are intended to give students comfort with a programming language feature or software environment. On these, any kind of assistance is permitted. The point is to get the work done.

2 Use of Course Materials

Course materials that are distributed in class, on Sakai, on the web, or by other means are provided solely for students in the class. Students are encouraged to use them to the fullest extent, but they are not to publish or distribute them to other people or organizations.

3 Responsibility of Mentors and Graders

Course assistants are routinely provided with solution sets to assignments. The solutions are intended to be an aid to effective mentoring and grading. Course assistants are not to distribute the solutions, in whole or in part, at any time.

Graders who encounter suspicious similarities between submissions must report those instances to the instructor in the course.

4 Penalties

Failure to abide by our rules will be considered a violation of the college's academic honesty policy and will result in severe penalties. Instances of plagiarism are easy to identify and will be handled promptly. The first offense typically results in failure in the course and is always reported to the Dean of Students Office. A second offense is automatically referred to the College's Board of Academic Discipline. See the Academic Honesty Policy in the Pomona College Student Handbook for further information. Students from other Claremont Colleges will be treated according to the procedures of their home campus. Please do not put us, yourself, or anyone else in an unpleasant situation.