COS 451: Automata, Computability, and Languages
and
COS 550: Theoretical Computer Science

Sudarshan S. Chawathe

University of Maine

Fall 2024

This course is an introduction to the theory of computation. Some big questions: What is a computer? How may we model computers and computation? What are the theoretical and practical limits of computation? What do we know about what can, and cannot, and may or may not be computable and efficiently computable? Some more details, from the course catalog: Fundamentals of formal languages and the mathematical theory of computation; finite-state automata, nondeterminism, regular expressions, and Kleene's Theorem; context-free grammars, pushdown automata, the correspondence theorem and the pumping lemma; computability, Turing machines, and the halting problem.

Prerequisites: COS 301; programming maturity.

COS 550 is the graduate version of this course, which shares most class meetings and coursework with the undergraduate version (COS 451) but that includes additional coursework and is assessed to a higher standard.

News and Reminders:

- Some sections below point to material in separate documents that are found on the class Web site, linked from the online version of this document.
- The most recent version of this document may be found at `http://chaw.eip10.org/cos451/`.
- Please use the PDF version of this document for printing and reference: `cos451.pdf`
- Brightspace site (access limited): `https://courses.maine.edu/d2l/home/360915`.

Goals and Learning Objectives

Goals

- Study various automata, such as deterministic and nondeterministic finite-state machines, pushdown automata, and Turing machines.
- Study formal languages of different kinds, such as regular and context-free languages.
- Understand the connections between languages and automata, and related algorithms for transformations.
- Understand the basic results on computability, including undecidable problems such as the halting and Post correspondence problems, and their significance.
- Study the basics of intractability, including NP-completeness and related topics.
- Make connections between theoretical results and topics in practical software development, such as finite automata and regular-expression libraries.

- Improve programming skills, with emphasis on connections between theoretical results and practical software.

Student Learning Outcomes

Students should be able to

- determine the detailed action of given automata on given inputs (e.g., determine whether a given DFA accepts a given string).
- devise simple automata to satisfy given properties (e.g., devise a pushdown automaton to recognize a given language).
- perform tasks analogous to the above for grammars and other linguistic formalisms (e.g., devising a formal grammar for a language described in English).
- use standard algorithms to transform automata and languages in various ways (e.g., mapping context-free grammars to pushdown automata).
- map instances of problems using standard reductions (e.g., 3-SAT to CLIQUE).
- demonstrate understanding of the above by writing suitable programs.

Contact Information

Class meetings:

Time: Mondays, Wednesdays, and Fridays, 09:00–09:50 a.m.
Location: Neville Hall 227.

Instructor: Sudarshan S. Chawathe

Office: Boardman Hall, Room 329.
Office hours: (Please check for changes.)
Mondays, Wednesdays, and Fridays; 8:30 a.m.–8:50 a.m. and 11:00 a.m.–11:30 a.m. (An appointment is *not required*, but advance notification is *useful*.)
Others by appointment, possibly remote/online.
Phone: +1-207-581-3930.
Please avoid calling except for truly urgent matters.
Email: `sudarshan.chawathe@maine.edu`
Use email only for messages unsuitable for the discussion forum. (See below.) Please use only this email address and put the string *COS451* near the beginning of the Subject header of the message. *Responses to all other messages may be very significantly delayed.*
Web: `http://chaw.eip10.org/`.

Online Resources

Class Web site:
`http://chaw.eip10.org/cos451/`
We will use the class Web site for posting assignments, readings, notes, and other material. Please monitor it.

Class discussion forum: We will use the university's *Brightspace* installation for class discussions outside class meeting times.

Class mailing list: *Please make sure you are on the class mailing list.* The mailing list will use the email address for each student as recorded in the official university records (*MaineStreet* system). We will use this mailing list only for urgent messages because all other messages will go on the class discussion forum. I anticipate fewer than a dozen messages on this list over the semester.

Grading Scheme

Grade components: *Students are expected to complete and submit all assigned coursework in good faith; those who fail to do so will earn a* ***failing grade, regardless of overall numerical score****.* Cut-offs for final letter grades D, C, B, A are, respectively, 35, 55, 70, and 85. Actual grades may be higher; these cut-offs provide lower bounds.

component	% of grade
class participation & exercises	10
homeworks	20
two quizzes	20
two midterm exams	30
final exam	20

Attendance: Attending class is required. Each student is granted three penalty-free absences for the semester, no questions asked. Beyond those three, each absence results in a *loss of one overall grade percentage point*. Absences for valid reasons (e.g., medical, family, religious, academic, athletic) may be excused *only if* a request is made very soon after the circumstances are known.

Class participation: Students are expected to contribute to learning by asking questions and making relevant comments in class and participating in the specified online components of the class. Quality is more important than quantity. Disruptive activity contributes negatively. See policies below.

Classroom exercises: Our work in the classroom may include some short individual and/or group exercises, meant to solidify understanding of the concepts being discussed. The exercises will be graded primarily for effort, group work, and other contributions, and less so for simple correctness.

Homeworks: Homeworks include programming and non-programming ones, often mixed. No collaboration is permitted. Everyone is encouraged to discuss the problems and solution strategies *at a high level*, but the final solution and details must be individual work. If the boundary between permissible and non-permissible interactions is unclear, please ask for clarifications.

Exams and quizzes: All exams and quizzes are *open book, open notes*. The use of computers and similar devices (tablets, phones, etc.) during exams is *not allowed*, A special exception is made for devices that are used purely as e-book readers, subject to important restrictions discussed in class. (Briefly, only those things are permitted that are just as easily done using a physical copy of a book.) Communications of any sort (electronic or other) are *not allowed*, except for communication between a student and the exam proctor. (In particular, Internet access is not allowed, even to access an e-book; so please ensure that you have locally-saved offline copies of your books.)

Policies

Due dates: All due dates and times, as announced in class, are strict, to the second. If you believe your work was delayed by truly exceptional circumstances, let me know as soon as those circumstances are known to you and I will try to make a fair allowance. However, *the default is that you get a zero if you don't turn in the work on time*, and fail the class if you don't turn it in at all (cf. Grade Components above). A *very limited* late-submission option *may* be announced in class later.

Attendance: It is very important to attend all class meetings, for many reasons, but the grade component provides additional motivation to those who may need it. If you have a valid reason for missing a class, let me know early and I will try to help you make up the class. There will be no make-up exams or quizzes. A missed test earns zero credit. If you have a valid reason for missing a test, let me know as early as that reason is known to you and I will make a fair allowance but there will be no make-up tests in any case.

Classroom activities: This course is based on an active learning format, so effective classroom activities are critical to its success. Students are expected to contribute to their own learning and to that of their classmates, and to devote 100% of their attention to these activities while in class. For this reason, electronic and other distractions (computers, phones, assorted gizmos, etc.) may be required to be be completely silenced and put away for some or all of the duration of class meetings. (Students who need any such devices for disability accommodations should follow the appropriate procedures, or contact the instructor if in doubt. Others who need any accommodation in this regard due to special circumstances should make advance arrangements with the instructor.) No food or drink is allowed in class, other than water, tea, coffee, and similar, in a spill-proof container. (As a general guideline, we will follow "library rules" in this regard.) Students who violate these rules or otherwise cause distractions in class will be asked to leave with *no warning*; habitual violators will face disciplinary action.

Office hours: All students are encouraged to make use of office hours to further their learning, obtain assistance on homework assignments, obtain feedback on their class performance, etc. However, office hours are not to be used as a substitute for attending and participating in class meetings (see above). Similarly, assistance with homework assignments will be limited to what is appropriate based on fairness to all; students are expected to demonstrate substantial effort on the assignment before seeking assistance.

Make-up classes: I may have to reschedule a few classes due to my other professional commitments. I will make every attempt to minimize the number of such occurrences and to reschedule for a time that works for most students. Further, I will make sure no student is penalized by such occurrences.

University of Maine administrative policy statements: [Verbatim, standard wording from `https://umaine.edu/citl/teaching-resources-2/required-syllabus-information/`. Please refer to that site for further details.]

Academic Honesty Statement Academic honesty is very important. It is dishonest to cheat on exams, to copy term papers, to submit papers written by another person, to fake experimental results, or to copy or reword parts of books or articles into your own papers without appropriately citing the source. Students committing or aiding in any of these violations may be given failing grades for an assignment or for an entire course, at the discretion of the instructor. In addition to any academic action taken by an instructor, these violations are also subject to action under the University of Maine Student Conduct Code. The maximum possible sanction under the student conduct code is dismissal from the University.

Students Accessibility Services Statement If you have a disability for which you may be requesting an accommodation, please contact Student Accessibility Services, 121 East Annex, 581.2319, as early as possible in the term. Students who have already been approved for accommodations by SAS and have a current accommodation letter should meet with me (the instructor of the course) privately as soon as possible.

Course Schedule Disclaimer (Disruption Clause) In the event of an extended disruption of normal classroom activities, the format for this course may be modified to enable its completion within its programmed time frame. In that event, you will be provided an addendum to the syllabus that will supersede this version.

Observance of Religious Holidays/Events The University of Maine recognizes that when students are observing significant religious holidays, some may be unable to attend classes or labs, study, take tests, or work on other assignments. If they provide adequate notice (at least one week and longer if at all possible), these students are allowed to make up course requirements as long as this effort does not create an unreasonable burden upon the instructor, department or University. At the discretion of the instructor, such coursework could be due before or after the examination or assignment. No adverse or prejudicial effects shall result to a student's grade for the examination, study, or course requirement on the day of religious observance. The student shall not be marked absent from the class due to observing a significant religious holiday. In the case of an internship or clinical, students should refer to the applicable policy in place by the employer or site.

Sexual Violence Policy Sexual Discrimination Reporting

The University of Maine is committed to making campus a safe place for students. Because of this commitment, if you tell a teacher about an experience of sexual assault, sexual harassment, stalking, relationship abuse (dating violence and domestic violence), sexual misconduct or any form of gender discrimination involving members of the campus, your teacher is required to report this information to the campus Office of Sexual Assault & Violence Prevention or the Office of Equal Opportunity.

If you want to talk in confidence to someone about an experience of sexual discrimination, please contact these resources:

For confidential resources on campus: Counseling Center: 207-581-1392 or Cutler Health Center: at 207-581-4000. For confidential resources off campus: Rape Response Services: 1-800-310-0000 or Partners for Peace: 1-800-863-9909.

Other resources: The resources listed below can offer support but may have to report the incident to others who can help:

For support services on campus: Office of Sexual Assault & Violence Prevention: 207-581-1406, Office of Community Standards: 207-581-1409, University of Maine Police: 207-581-4040 or 911. Or see the OSAVP website for a complete list of services at http://www.umaine.edu/osavp/

Programming

This course focuses on high-level concepts that are mostly oblivious to choices of programming languages and environments. However, in order to provide concrete realizations of these concepts, we will use Python as the primary programming language and a POSIX environment as the primary operating system.

Submissions will be in the form of packaged, well documented *source* files. *Proper documentation and packaging of source code and other material is a crucial component of assigned work and submissions failing in this regard will receive no credit.*

Other Languages: If you prefer to use other programming languages or systems, please contact me by the third class meeting. I am quite open to the idea, and encourage interested students to explore it further. However, please check with me very early in the semester so that we can determine the specifics to make sure your submissions can be tested and graded fairly. You should avail of this option only if you are confident enough of your programming skills to not require any programming help, and are prepared to take on additional work. *This option is designed for students who are proficient in Python and wish to use this opportunity to master another language, not for students weak in Python and who wish to avoid it.* Anyone granted this option will still be responsible for all the material related to the default languages and systems used in the course.

Literate Programming: All submitted work must use a *literate programming style*: Your programs must be designed with *a human as the intended reader*, although they must also direct a computer correctly. *Programs that do not meet this requirement are likely to receive a zero score with no further consideration.* Details will be discussed in class.

Schedule

A rigid schedule is not conducive to effective learning, since it would limit our flexibility in exploring ideas as they arise in class. A partial and *approximate* schedule, to serve as a baseline, appears in Figure 1; it will be updated as we progress. Please use it only as a rough guide to plan your studies. *Do not use it to schedule travel or other events.* If you need a definite answer on when something will or will not occur, you should check with me.

At the beginning and end of each class, I typically announce the topics and textbook sections covered in that class and those due at the next class. It is important that students read the material *before* the class in which it is discussed and, in general, keep up with readings and studies.

Textbook and Readings

Required textbook: Michael Sipser. *Introduction to the Theory of Computation.* Cengage Learning, 3rd edition, 2013.

Readings: This list will change as we progress through the semester, based on student interests and classroom discussions.

1. Ken Thompson. Reflections on trusting trust. *Communications of the ACM*, 27(8):761–763, August 1984.

Monday	Wednesday	Friday
September 2nd ✕ *No class.* Labor Day.	4th C1 Introduction; Fun and Games. § 6.1.	6th C2 § 0.*.
9th C3 Finite-state automata (FSAs). § 1.1.	11th C4 Non-determinism (FSAs). § 1.2.	13th C5 Regular expressions (regexes). § 1.3.
16th C6 Equivalence of regexes and FSAs. § 1.3.	18th ★ **Quiz 1**	20th C7 Nonregular languages. § 1.4.
23rd C8 Context-free grammars (CFGs). § 2.{0, 1}.	25th C9 Catch-up; review.	27th C10 Pushdown automata (PDAs). § 2.2.
30th ★ **Midterm Exam 1**	**October** 2nd C11 CFGs and PDAs. § 2.{2, 3}.	4th C12 Non-context-free languages. § 2.3.
7th C13 Catch-up; review.	9th C14 Special topic; catch-up; review.	11th C15 Turing Machines. § 3.1.
14th ✕ *No class.* Fall break Oct. 14th–15th.	16th C16 Turing Machine variants. § 3.2.	18th C17 Church-Turing Thesis. § 3.3.
21st C18 Catch-up; review.	23rd C19 Decidability. § 4.{0, 1}.	25th C20 Undecidability. § 4.2.
28th C21 Reducibility. § 5.1.	30th C22 Post Correspondence Problem (PCP). § 5.2.	**November** 1st C23 Catch-up; review.
4th C24 Mapping reducibility. § 5.3.	6th ★ **Quiz 2**	8th C25 Time complexity basics and the class P. §§ 7.{0, 1, 2}.
11th ✕ *No class.* Veterans Day.	13th C26 The class P; CYK algorithm. § 7.2.	15th C27 Catch-up and review.
18th C28 The class NP. § 7.3.	20th ★ **Midterm Exam 2**	22nd C29 NP-completeness. § 7.4.
25th C30 NP-complete problems. § 7.5.	27th ✕ *No class.* Thanksgiving break Nov. 27th–Dec. 1st.	29th ✕ *No class.* Thanksgiving break Nov. 27th–Dec. 1st.
December 2nd C31 Space complexity; Savitch's Thm.; PSPACE completeness. §§8.1–8.3.	4th C32 Classes L and NL. §§ 8.4–8.5.	6th C33 Synthesis and review.
9th C34 Synthesis and review.	11th C35	13th C36
16th ✕ *No class.* ★ **Finals week** Dec. 16th–20th.	18th ✕ *No class.* ★ **Final exam: Dec. 18th 12:15–2:15 p.m.**	20th ✕ *No class.* ★ **Check Univ. schedule for final exams.**

Figure 1: **Approximate** schedule, likely to change. Notation: **§§ *x.y*** ⇒ textbook chapter x, section y.

2. Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 212–219, Philadelphia, PA, May 1996.

Exercises, Homeworks, Tests, and Notes

Material will appear here as we progress through the semester.

It may be useful to refer to the homeworks and tests from a previous session (recursively): `http://chaw.eip10.org/202201/cos451/`.

- Class exercises:
  - Class Exercise 1: `hwq/ce01.pdf`.
  - Class Exercise 2: `hwq/ce02.pdf`.
  - Class Exercise 3: `hwq/ce03.pdf`.
  - Class Exercise 4: `hwq/ce04.pdf`.
  - Class Exercise 5: `hwq/ce05.pdf`.
  - Class Exercise 6: `hwq/ce06.pdf`.
- Homework assignments:
  - Homework 1: `hwq/hw01.pdf`.
  - Homework 3: `hwq/hw03.pdf`.
- Exams
  - Quiz 1: `hwq/q01.pdf`. Sample solutions: `p/q01s.pdf`.
  - Midterm Exam 1: `hwq/mt01.pdf`. Sample solutions: `p/mt01s.pdf`.
  - Quiz 2: `hwq/q02.pdf`. Sample solutions: `p/q02s.pdf`.

Homework and Project Submissions

All electronic submissions must be made using the procedure that will be outlined in class and here later. Electronic submissions in **all other forms**, such as email or physical media, will be **discarded and receive no credit.**

Illegible, hard to read, or otherwise messy submissions, whether handwritten on typed, are likely to be returned without grading, for zero credit.

Fallback procedure If (and only if) there are unexpected problems and you are unable to submit your work as above, then you should save your file on your own computer (with some backups), compute its MD5 checksum using the md5sum utility on Unix-like systems (or other similar tools), and submit the file name, time stamp, and MD5 checksum (only, not the file itself) by email with a suitable Subject header.

Keys to Success

1. Don't lose sight of the fun parts. If you cannot find them then ask for help.
2. Take responsibility for your own learning. At the first sign of trouble, no matter how minor, seek help. I am here to help but you must take the first step.

3. Read *every day*! The textbook, code, your own notes, discussion forum, and more.

4. Write *every day*! The textbook (mark it up!), notes (in and out of the classroom), code, discussion forum, more.