COS 451: Automata, Computability, and Languages

Sudarshan S. Chawathe

University of Maine

Fall 2019 preliminary version

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 Kleenes Theorem; context-free grammars, pushdown automata, the correspondence theorem and the pumping lemma; computability, Turing machines, and the halting problem.

Prerequisite: COS 250.

News and Reminders:

- Please read the newsgroup for timely announcements.
- Class newsgroup: Local group `umaine.cos451` on NNTP server `creak.um.umaine.edu`. Web interface to get started: `http://chaw.eip10.org/news/`.
- The most recent version of this document may be found at `http://chaw.eip10.org/cos451/`.
- 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.
- Please use the PDF version of this document for printing and reference: `cos451.pdf`

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; 9:00–9:50 a.m.
Location: Neville Hall, Room 116.

Instructor: Sudarshan S. Chawathe

Office: Boardman Hall, Room 329.
Office hours: (Please check for changes.)
- Mondays, Wednesdays, and Fridays; 10:50–11:40 a.m.

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 newsgroup. (See below.) Please use only this email address and put the string *COS451* near the beginning of the Subject header of the message. *All other messages may be ignored.*
Web: `http://chaw.eip10.org/`

Teaching Assistant: Mark Royer

Office: East Annex, Room 225 (but see below).
Office hours: (Please check for changes.) *In Boardman Hall, Room 138.*
- Tuesdays: 1:00–2:30 p.m.
- Fridays: 10:00–11:30 a.m.

Phone: +1-207-581-3946.
Email: `mark.royer@maine.edu`
Web: `https://markroyer.me/`

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 Newsgroup: We will use the local USENET newsgroup `umaine.cos451` on the NNTP (net news) server `creak.um.umaine.edu` for electronic discussions. The Web interface at `http://chaw.eip10.org/news/` provides convenient access. Some further, more general, information on USENET appears at `http://en.wikipedia.org/wiki/Usenet`. The newsgroup is the primary forum for electronic announcements and discussions, so please monitor it regularly, and post messages there as well. Unless there is a reason for not sharing a question or comment, please *use the newsgroup, not email*, for questions and comments related to this course.

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 newsgroup. 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****.*

component	% of grade
class participation	5
newsgroup activities	5
homeworks	25
three quizzes (short exams)	15
three midterm exams	30
final exam	20

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

Newsgroup activities Discussions on the class newsgroup are an important mode of learning. To encourage everyone to participate actively in these discussions, there is a portion of the grade assigned to how well, and how often, students participate on the newsgroup. Details of how newsgroup activity is scored will be described in class. Briefly, students' scores will reflect the number and regularity of their messages, the significance of the observations they make (including questions), and the initiative and self-directed learning they reflect.

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*. You are free to bring with you any resources that you find useful. However, no communications are permitted other than between students and me. The use of computers during exams is strongly discouraged, but brief use may be permitted provided it does not cause a disturbance, at the discretion of the proctor. You may use the Internet, but only as a library to look up material you may find useful. Ask for clarifications in case of any doubt. The exams are designed to require no equipment other than a pen and paper, along with the textbook and assigned readings.

Midterm exams will be held during regular class meetings, and will be roughly an hour long. Each quiz is a short exam, roughly half an hour long, held during part of a class meeting. The final exam follows the usual university schedule, and is thus held outside of regular class meeting times, and often at a different location.

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).

Attendance: Although I expect students to attend all class meetings, I will not be taking attendance. *If you miss a class meeting, you are responsible for catching up on the lost material, including any important announcements made in class, on your own.* 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 that of their classmates, and to devote 100% of their attention to these activities while in class. On a similar note, all electronic and other distractions (computers, phones, assorted gizmos, etc.) must be completely silenced and put away for the entire duration of the class. (Students who need any such devices for disability accommodations should follow the guidelines outlined below. 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 in a spill-proof container. 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 both the instructor's and TA's 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/`.]

- **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.
- **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 Java as the primary programming environment 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.*

Programming Environment and Tools: You are free to choose details such as operating system, development environment, and editor based on your preferences. However, no matter what you use, the submission should be a *source*-code package that works on the host *aturing* (see below). Further details on the packaging, submission, and testing procedure will be provided in class and on the newsgroup.

Other Languages: If you prefer to use other programming languages or systems, please contact me by the second 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 Java and wish to use this opportunity to master another language, not for students weak in Java and who wish to avoid them.* 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 compile and run 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. The use of any specific literate-programming or documentation tool is neither necessary nor sufficient for this requirement.

Class accounts: Shell accounts will be generated on the host `aturing.umcs.maine.edu` based on registration records. These accounts are required for successful completion of homeworks and other assignments. You should be able to access your accounts from anywhere on the Internet by using *ssh*. On most Unix-like hosts (GNU/Linux, Mac OS), the command `ssh -l` *`username`* `aturing.umcs.maine.edu` should suffice. For Windows hosts, the freely available *Putty* program works well.

Schedule

A rigid schedule is not conducive to effective learning, since it would limit our flexibility in exploring ideas as they arise in class. The actual schedule (both the timing and the selection of topics) will be determined by in-class interactions. Nevertheless, a partial and *approximate* schedule, to serve as a baseline, appears in Figure **??**; 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

Textbook: Michael Sipser. *Introduction to the Theory of Computation*. Cengage Learning, 3rd edition, 2013. The university bookstore carries this book, which is a required textbook for this course.

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.
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.
3. Julia Chuzhoy, David H. K. Kim, and Rachit Nimavat. New hardness results for routing on disjoint paths. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 86–99, New York, NY, USA, 2017. ACM.

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 the previous session (recursively): `http://chaw.eip10.org/201801/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`.
  - Class Exercise 7: `hwq/ce07.pdf`.
  - Class Exercise 8: `hwq/ce08.pdf`.
  - Class Exercise 9: `hwq/ce09.pdf`.
  - Class Exercise 10: `hwq/ce10.pdf`.
  - Class Exercise 11: `hwq/ce11.pdf`.
  - Class Exercise 12: `hwq/ce12.pdf`.
  - Class Exercise 13: `hwq/ce13.pdf`.
  - Class Exercise 14: `hwq/ce14.pdf`.
  - Class Exercise 15: `hwq/ce15.pdf`.
  - Class Exercise 16: `hwq/ce16.pdf`.

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

  - Class Exercise 17: `hwq/ce17.pdf`.
  - Class Exercise 18: `hwq/ce18.pdf`.
  - Class Exercise 19: `hwq/ce19.pdf`.
  - Class Exercise 20: `hwq/ce20.pdf`.
  - Class Exercise 21: `hwq/ce21.pdf`.
  - Class Exercise 22: `hwq/ce22.pdf`.
- Homework assignments:
  - Homework 1: `hwq/hw01.pdf`.
  - Homework 2: `hwq/hw02.pdf`.
  - Homework 3: `hwq/hw03.pdf`.
  - Homework 4: `hwq/hw04.pdf`.
  - Homework 5: `hwq/hw05.pdf`.
- Quizzes and Exams:
  - Quiz 1: `hwq/q01.pdf`.
  - Midterm Exam 1: `hwq/mt01.pdf`.
  - Quiz 2: `hwq/q02.pdf`.
  - Midterm Exam 2: `hwq/mt02.pdf`.

Homework Submissions

Handwritten answers to non-programming problems should be submitted in class on the due date, at the beginning of class (within the first five minutes), unless prior alternate arrangements are made. If you prefer to type your answers, please make sure the result uses the proper symbolic notation for mathematical constructs. *Illegible, hard to read, or otherwise messy submissions, whether handwritten on typed, are likely to be returned without grading, for zero credit.* Answers to programming problems should be submitted electronically, using the packaging and submission procedure that will be described in class and on the class newsgroup.

All electronic submissions must be made using the upload interface at `http://chaw.eip10.org/u/`. Electronic submissions in **all other forms**, such as email or physical media, will be **discarded and receive no credit.**

If your upload is successful, you will be presented with a confirmation Web page similar to the following sample. You should record the reported MD5 checksum and timestamp (important in case there are undetected problems).

```
SUCCESS: Please note the following for your records.

Successfully saved cos451-hw01-aardvark-alice-4323.tgz.
MD5 checksum: 09ee098b83d94c7c046d6b55ebe84ae2
Timestamp: 2019-09-03 17:24:03
```

If you do not see something very similar then your submission is unsuccessful.

Contingency 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.

Do not submit your work by email; it will be discarded; really.