CS 520 An Introduction to the Theory of Computation (3 credit hours)

Professor: Jin-Yi Cai
Office: 4393 Computer Science
http://www.cs.wisc.edu/~jyc/
Phone: 608-262-3158
e-mail: `jyc (jyc@cs.wisc.edu)`
Class room: COMP SCI 1325
Time: TuTh 11:00AM - 12:15PM
Office hours: TuTh 3:30PM - 5:00PM (subject to change) or by appointment.
Textbook: *Introduction to the Theory of Computation,*
by Michael Sipser, PWS Publishing Company, ISBN 113318779X **third** edition.
TA: Ben Young
TA e-mail: `bmyoung4@wisc.edu`.
Office room: 3219 Computer Science (unit 2)
TA office hours: Wednesday 10am-11am and Friday 11am-12:30pm.

Midterm exam: 3/11/2025 (Tuesday), in class.
Final Exam: 5/7/2025 (Wednesday), 10:05AM–12:05PM, Room Pending

This course is an introduction to the theory of computation. We will study automata theory, formal languages, the theory of computability and undecidability, and some basic computational complexity, especially P and NP, including NP-completeness. This material is the theoretical foundation of computer science; it underlies the basic principles of compiler design, programming language design and processing, the efficiency of algorithms, and the limitations of computing. Complexity theory is also what makes public-key encryption possible. This will be perhaps the first course in which a computer science student is exposed to the rigorous treatment of computation as a formal process, and to explore the fundamental theory of computability and complexity.

The course will focus on:

1. Various computational models such as Finite Automata, Pushdown Automata and the Turing machines. The hierarchy and interplay between the levels of computational devices are the core material of our course.

2. Formal language counter parts of these computational devices, such as regular languages, context-free languages and recursive languages. Mathematical techniques useful in the theory will be emphasized.

Topics include:

Deterministic finite automata; Nondeterministic finite automata; Nondeterministic finite automata with ϵ-moves; Regular expressions; Regular languages; Equivalence theorem; Myhill-Nerode Theorem; Finite automata minimization; Pumping Lemma for regular languages;

Context-free languages; Push-down automata; Equivalence theorem; Non-context-free languages;

Turing Machines; Countable and uncountable sets; Diagonalization; Decidability Theory; Undecidability; Halting Problem; Reducibility; Complete sets;

P & NP; Satisfiability; 3-SAT; More NP-completeness such as Vertex Cover, Independent Set, Hamiltonian Circuit, 3-Dimensional Matching, Set Cover, Partition, Knapsack, Subset Sum, etc.

Some problems are NP-complete, but solvable in pseudo-polynomial time. We will discuss the distinction here.

The core material of this course constitutes the most fundamental concepts of computer science; one can say the stuff of this course is what makes computer science a hard science. Students are expected to spend a great deal of time thinking about the material and homeworks. Computer programming skills and experience may be helpful in guiding intuition, but primarily students are expected to exercise mathematical reasoning and logical deduction. Mathematical rigor in presentation is expected. Essays or philosophical discussions are generally not acceptable as answers to precise questions. ChatGPT and similar apps are **Not** allowed to complete your course work. There will be weekly homeworks, which serves as regular checks on your understanding of the material of the course. It will be graded and count for approximately 20% of the grades. **Each student should find a fellow student to pair up, and jointly discuss the homeworks and submit one solution to be graded.** This is a good learning opportunity; it is often that when you can explain your ideas to another person is the time that you fully understand. (See more below.)

Homeworks should be written legibly. (It is your responsibility that I and the TA and the grader can understand your answers.) There will also be an in-class written midterm, which counts for roughly 30%, and an in-class written final exam, which counts for roughly 45%. (Roughly 5% will be given for class participation, extra credit, particularly innovative ideas etc.) While you and your partner in your pair-team work jointly on the homeworks, each should actively participate for every homework problem. No one should be a free loader to your partner. It is expected that the two are equal contributors. If not, your performance on the midterm and final exams will suffer. Students are encouraged to discuss among themselves on the course material. But the homeworks should be done by you and your partner. You cannot consult any source for ready answers to homework problems. I encourage you to think independently. Your midterm and final exams will be all by yourself. The midterm and final exam are closed book, closed notes. Homeworks are due on the due date **in class**. **No late homeworks will be accepted,** except by explicit permission of the TA or the Professor, under exceptional circumstances. All students are expected to observe faithfully the code of academic integrity. Cheating of any sort is subject to an F grade for the entire course and possible further consequences. All students are expected to attend *all classes*. Attendance is vitally important to your understanding of the course material. I intend to take occasional attendances. (For exceptions which are not unforeseeable, such as for religious observances, students should notify the Professor within the first two weeks of class.) Class participation and/or extra credits on homeworks will be taken into account for the final grade.

You are encouraged to ask questions, in class or after class, and during TA's or my office hours. (You should ask questions that pertain to the subject matter of the class, and not the "How do I do this problem of the homework" variety.) See your TA first regarding any grading questions, and only if such a problem is not resolved you should come to see me.

For a more philosophical take on the materials of this course, I highly recommend the following book by Douglas R. Hofstadter: "Godel, Escher, Bach: an Eternal Golden Braid"
http://www.forum2.org/tal/books/geb.html

• Course learning outcomes: Students are expected to learn the fundamental concepts in the theory of computing. They are expected to become familiar with formalization of computation, models of computation including finite automata and regular languages, pushdown automata and context free languages, Turing machines and computability theory, reasoning about process and correctness. They are also expected to become familiar with the basic ideas of the P and NP theory.

• Number of credits associated with the course: 3 credits.

• How credit hours are met by the course: This class meets for two 75-minute class periods each week over the Spring Semester and students work on course learning activities (reading, writing, problem sets, studying, etc) for about 3 hours out of classroom for every class period.

Tentative Syllabus for CS 520

Week 1: (1/21, 1/23) A general introduction. Mathematical induction. Formal definition of DFA. NFA. Simulation of NFA by DFA.

Week 2: (1/28, 1/30) Subset construction and simulation of NFA by DFA. NFA with ϵ-moves. Regular expressions. Simulation of REG EXP by NFA with ϵ-moves.

Week 3: (2/4, 2/6) Dynamic Programming. Simulation of DFA by REG EXP. Equivalence Theorem. Pumping Lemma for regular sets. Non-regular languages. PRIMES is not regular.

Week 4: (2/11, 2/13) Public-key Cryptography. RSA protocol. Myhill-Nerode Theorem. DFA minimization.

Week 5: (2 /18, 2/20) Context-free grammars. Derivations and Derivation Trees. Context-free languages. Normal forms.

Week 6: (2/25, 2/27) Ambiguity. Pumping Lemma for CFL. Non-context-free languages. Push-down automata. Equivalence theorem.

Week 7: (3/4, 3/6) Countable and uncountable sets. Diagonalization. Computability.

Week 8: (3/11 **Midterm**, 3/13) Turing Machines. Church-Turing thesis.

Week 9: (3/18, 3/20) Halting Problem. Decidability and Undecidability. Reducibility. Recursive sets and Recursively enumerable sets. Complete sets.

Spring Break (3/20–3/30)

Week 10: (4/1, 4/3) More on Undecidable sets. Non-computable sets. Rice's Theorem. Post Correspondence Problem.

Week 11: (4/8, 4/10) Introduction to P and NP. NP-completeness.

Week 12: (4/15, 4/17) Satisfiability problem. SAT and 3SAT.

Week 13: (4/22, 4/24) Vertex cover, Independent Sets, CLiques. Hamiltonian circuits.

Week 14: (4/29, 5/1) More NP-completeness. 3DM. Set covers. Subset Sum, Pseudopolynomial time. Knapsack, Partition. Review.

Final Exam. 5/7, **Wed.** 10:05 AM – 12:05 PM