CS 341 Foundations of Computer Science II
Spring 2023

Instructor: Dr. Ravi Varadarajan
Email: ravi.varadarajan@njit.edu
Office Hours: Will be posted in Canvas

Prerequisites
You must complete all of the following with grades of C or better:

1. A 100-series general undergraduate required course in CS

2. CS 241 (Foundations of Computer Science I)

3. CS 280 (Programming Language Concepts).

Course Goals
The course aims to steer the students towards understanding the fundamental computability results in Computer Science, namely

- Can a given problem be solved by computation?
- How efficiently can a given problem be solved by computation?

Towards that end, students will be introduced to various finite state automata models and their power and limitations in solving various computing problems.

Course learning outcomes

- Classify a particular language as regular, context-free, decidable, Turing-recognizable or non-Turing-recognizable.
- Provide a finite automaton and regular expression for a regular language.
- Prove that a specified language is not regular.
- Provide a context-free grammar and pushdown automaton for a context-free language.
- Prove that a specified language is not context-free.
- Provide a description of a Turing machine for a decidable language.
- Prove or disprove closure properties (under union, intersection, complementation, Kleene star) of classes of languages.
- Prove that certain languages are undecidable or non-Turing-recognizable.
- Understand nondeterminism and its role in computation and complexity theory.

- Understand the significance of complexity classes P, NP and NP-complete, and carry out some NP-completeness reductions.

Course Text
Michael Sipser, *Introduction to the Theory of Computation, Third Edition*. Course Technology, 2012, ISBN-10: 113318779X, ISBN-13: 978-1133187790. Chapters 0-5 and 7 of the text book will be used in this course.

Course topics and schedule
Week 1 (Jan 16) : Introduction, Mathematical preliminaries (Ch. 0)
Week 2 (Jan 23) : Math. Preliminaries, Proof techniques (Ch. 0)
Week 3 (Jan 30): Finite state automata (DFA, NFA) (Ch. 1)
Week 4 (Feb 6): Regular languages (Ch. 1)
Week 5 (Feb 13): Pumping lemma for regular languages (Ch.1), Myhill-Nerode theorem, DFA minimization, midterm I review
Week 6 (Feb 20): **Midterm exam I,** Context-free grammars (Ch. 2)
Week 7 (Feb 27): Pushdown automata (PDA), Context-free languages (CFLs) (Ch. 2)
Week 8 (Mar 6): Pumping lemma for CFLs, Deterministic CFLs, Deterministic PDAs (Ch. 2)
Mar 13-19: **Spring Recess**
Week 9 (Mar 20): Turing machines (Ch.3), Church-Turing thesis (Ch. 3)
Week 10 (Mar 27): Algorithms and decidability, halting problem (Ch. 4)
Week 11 (Apr 3): Midterm II review, **Midterm exam II**
Week 12 (Apr 10): Undecidable problems, reducibility (Ch. 5)
Week 13 (Apr 17): Complexity classes P and NP (Ch. 7)
Week 14 (Apr 24): NP-completeness, Cook's Theorem, NP-complete reductions
Week 15 (May 1) : Final exam review

There may be slight deviations of this schedule during the semester depending on the pace of lectures.

Grading Criteria
Homework assignments (12%) Pop quizzes (8%), Programming project (20%), Midterm Exams (30%), Final Exam (30%).

Programming project needs to be implemented in Java or Python and more details regarding the project will be revealed during the course.

Homework assignments will be assigned approximately once in 2 weeks. Late penalty of 10% per day will be assessed on a homework submission beyond the dead line up to 3 days after which no credit will be given to the assignments.

Pop quizzes will be given at the beginning of a lecture class.

All exams and quizzes will be closed- book and closed-notes. There are no make-up quizzes or exams.

Exceptions for missing any component of the final grade (e.g. quizzes, exams) will be considered on a case-by-case basis provided you submit documentation for the reason to the Dean of Students Office who in turn will contact me.

Grading Scale
Will be posted in Canvas.

Academic honesty policy

Read the following University Code on Academic Integrity:

"Academic Integrity is the cornerstone of higher education and is central to the ideals of this course and the university. Cheating is strictly prohibited and devalues the degree that you are working on. As a member of the NJIT community, it is your responsibility to protect your educational investment by knowing and following the academic code of integrity policy that is found at: http://www5.njit.edu/policies/sites/policies/files/academic-integrity-code.pdf.

All work that you submit as your own must, in fact be your own. Penalty for the most serious violations of academic integrity may be an F grade in the course or expulsion.

There should be no collaboration on home works or programming assignments. When you submit these, you will be asked to provide a declaration (exact wording to be provided later) that the work submitted is your own.