Theory of Computation

General Course Information

- *Term:* Spring 2025
- *Department*: COMP
- *Course Number:* 475/575
- *Time:* Tue & Thu, 12:30 pm - 1:45 pm
- *Location*: SERC 2036
- *Course Webpage:* bineet.cs.ua.edu/teaching/S2025/CS575/ToC.html

Instructor

Dr. Bineet Ghosh

- *Email:* bineet@ua.edu
  - Please add the following in subject line if you are emailing regarding the course: `[s25-CS575] <your-subject>`
    - Note: Emails that do not adhere to this guideline may experience significant delays in response.
- *Office:* Cyber Hall 3053
  - *Office Hours:* Thu, 2:20 pm - 4:20 pm
- *Instructor Webpage:* bineet.cs.ua.edu

Course Description

This course will introduce students to the mathematical models of computations. Following are some of the broad topics that will be covered in this course:

1. Finite automata
2. Regular languages
3. Pushdown automata
4. Context-free languages
5. Turing machines
6. Undecidable problems

Prerequisites

1. Algorithms (CS 201/470)
2. Data Structures and Analysis (eg., CS 201)
3. Discrete Structures (Or their equivalents).
4. Programming in Python (CS 223/323)
5. Mathematical Maturity
   - Have you formally proved correctness of algorithms?

- Are you familiar with proofs using induction?
    - *Test your knowledge of inductive proofs.* One can refer to basic inductive proofs (available online) and see how comfortable they are in understanding the proofs.
- Familiarity with theorems in graph theory would be useful
- Assignment 0 will be posted on the first day of classes (Jan 9) to test your pre-requisites.

Speak to instructor, if unsure.

Textbooks

Recommended textbooks:

1. *"Introduction to the Theory of Computation"*, by Michael Sipser **(Primary)**
2. *"Introduction to Automata Theory, Languages, and Computation"*, by Hopcroft, Ullman, and Motwani

Following additional resources could also be useful:

- **JFLAP:** www2.cs.duke.edu/csed/jflap/
- **Automata Tutor:** automatatutor.com

Course Objectives

By the end of this course students are expected to be able to:

1. Identify and classify languages based on its computation model.
2. Understand formal proofs done using induction (recreate proofs, finding bugs in proofs)
3. Clearly identify the differences between several computation models, *viz.*, the power it gains from going from one computation model to another.
4. Some basic understanding Undecidability is further desired.

Student Learning Outcomes

The students can test the above given objectives in the following way:

1. Identify and classify languages based on its computation model.
    1. Given a language (say by your friend), one should identify the model of computation that can efficiently solve the problem. Further, they should be able to comment on its complexity.
2. Understanding proofs.
    1. Students should be able to recreate inductive proofs by themselves.
    2. They should be able to find bugs in the proofs as well.
3. If a computation model is enhanced with a certain "operation", they should be able to comment if this enhancement results in increased computation power.
4. Given a problem, they should be able to comment if the problem can be solved by a machine. If so, they should be able to identify the complexity class.

Outline of Topics

1. Regular Languages

    1. Deterministic Finite Automata (DFA)
    2. Non-Deterministic Finite Automata (NFA)
    3. Regular Expressions
    4. Properties, operations, and equivalences (with rigorous proofs)
    5. Non-Regular Languages & Pumping Lemma
2. Context-Free Grammars
    1. Derivations
    2. Parse Trees
    3. Ambiguity
    4. Deterministic and Non-Deterministic Pushdown Automata (PDA)
    5. Properties and equivalences (with rigorous proofs)
3. Decidability
    1. Turing Machines (TM)
    2. Multi-Tape TM
    3. Recursively Enumerable Languages
    4. Undecidability
    5. Halting Problem

Grading

1. Assignments (Graded + Ungraded): 30%
    1. Assignment 0 will be posted on the first day of classes (Jan 9) to test your pre-requisites.
    2. Must be typed (preferably in `LaTex`), and not handwritten. However hand-drawn figures (such as automata) are permitted.
    3. Assignments will be clearly marked as graded/ungraded.
    4. Submission of ungraded assignments is mandatory. Failure to comply will substantially affect the class participation score.
    5. All submissions must follow this format: `<firstname-lastname-CWID>`.
    6. Almost every week.
2. Quiz: 20%
3. Presentation (of topics/papers), and leading class discussion: 20%
4. Final project (with presentation): 20%
5. Class participation & student’s growth: 10%

Student’s growth. Identifies the effort being made by a student to understand the concepts being taught, despite their prior backgrounds. In other words, this can be used identify the efforts being put in by a student even if it is not being reflected in their scores. Completing the ungraded assignments is an excellent way to showcase this.

Late work policy

- One late assignment will have 0 (zero) penalty, if submitted within a week of the posted deadline.
- Any further late assignments will not be graded.

- In case of emergency (or unforeseen cases), please speak to instructor. The instructor will try his best to accommodate for the situation (subject to proof) in accordance with the university policy.

Final project

- **Types**
    - **Survey.** Extensive survey about the research on a topic of your interest. Not a reading and copy-paste of existing papers. Expected to identify the core of the problem and provide deep insights.
    - **Applications.** Pick an application of your interest and verifying it. Might require deploying several techniques and tools. Might lead to interesting theoretical results. The application should be nontrivial and "interesting" enough.
    - **Theory.** Identify a theoretical problem (algorithm/property of models) and develop new fundamental solutions. Fairly challenging task. Identifying a "good" candidate solution or proving an "interesting" property suffices.
- **Logistics**
    - Each group member is responsible to *clearly* identify their contribution.
    - 1-2 reports maybe due before the final report and presentation to share progress with the class.
    - *Presentation:* 15 mins, including discussion, debate, comparative analysis.
        - Following elements should be a part of your presentation:
            - Problem statement
            - Related work
            - Methodology
            - Main novelty
            - Results. Identify the interesting parts (your opinion)
            - Have at least two questions for the class. For each question, students will form groups to discuss, and then one member of each group will volunteer to answer the question.
    - *Report:* 8 page document in IEEE conference format (similar to conference paper).
    - *Guaranteed way to score A+:* Project report of conference paper quality.

Attendance and Participation

- Though attendance will not be formally recorded, it is recommended the lectures are attended. Please note that lectures are the best way to demonstrate the student growth (which has 10%).
- Further, if a student has missed noticeably many lectures (for instance, shows up only during their presentation/Quiz), there will be a minimum of 50% deduction from their obtained score in the Student's Growth section.
- Students are responsible to make up for missed courseworks.

Disclaimer Notification of Changes

The instructor will make every effort to follow the guidelines of this syllabus as listed; however, the instructor reserves the right to amend this document as the need arises (including project due dates and test dates). In such instances, the instructor will notify students in class and/or via email and will endeavor to provide reasonable time for students to adjust to any changes.

Standard Course Policies

Please refer to the university's standard course policies (Honor code, misconduct, behavior etc.) for details.