**Computer Language Theory**

MET CS 662 A2

**Madani Naidjate**

[madani@bu.edu](mailto:madani@bu.edu)

**Course Description**

The goal of this course is to provide the student with a solid knowledge of the fundamental concepts and methods of the theory of computation as well as to outline modern research directions. Three different approaches for capturing the idea of computing in a formal mathematical way will be discussed finite state machines, grammars and recursive functions. At the end of the course students are expected to be able to interpret relate and apply the basic concepts of the theory of computation to problems from different areas of computer science.

**Course Objectives**

* Apply the algebra of theoretical machines to computing problems
* Possess knowledge of non-determinism in computational models.
* Remove nondeterminism from simple models when it is possible.
* Understand the basic problems of computability, decidability and the halting problem as well as the relationships among them.
* Apply the concept of a Turing machine to the decidability problem where possible.
* Understand the Church-Turing Thesis and its significance in computer science.
* Have a working knowledge of the Chomsky hierarchy of languages.
* Relate theoretical computer science topics to programming languages and other recursively enumerable sets.

**Prerequisites**

* Discrete Mathematics course (MET CS248 or equivalent.)
* Introductory computer programming class

**Textbook**

* P. Linz "An Introduction to Formal Languages and Automata" *any* edition, D.C. Heath and Co. (2001 - 2016). Available at Barnes & Noble or on-line.

**References**

* 1- H. R. Lewis, C. H. Papadimitriou "Elements of the Theory of Computation" Prentice Hall, 1981.
* *2-* J.E. Hopcroft, "Introduction to Automata Theory, Languages and Computations" Addison Wesley, 1979.

**Courseware**

All assignments course materials and announcements pertinent to the course can be found on Blackboard.

**BU Community COVID-19 Public Health Policies**

All students returning to campus will be required to be [vaccinated against COVID-19](https://www.bu.edu/back2bu/campus-life-graduates/bu-community-health-safety/covid-19-vaccination-information/), and upload information about their status (including applications for a medical or religious exemption or an extension) to the [Patient Connect](https://patientconnect.bu.edu/) portal. In addition to the vaccine requirement, students must follow all other safety protocols, including the [face covering policy](https://www.bu.edu/back2bu/campus-life-graduates/bu-community-health-safety/face-coverings/), and [screening](https://www.bu.edu/back2bu/campus-life-graduates/bu-community-health-safety/covid-19-screening-testing-contact-tracing/screening/), [contact tracing](https://www.bu.edu/back2bu/campus-life-graduates/bu-community-health-safety/covid-19-screening-testing-contact-tracing/contact-tracing/), and [testing](https://www.bu.edu/back2bu/campus-life-graduates/bu-community-health-safety/covid-19-screening-testing-contact-tracing/testing/) requirements. At the beginning of each class you will be asked to show a green [Healthway](https://www.bu.edu/healthway/) compliance badge on your mobile device to the instructor, and wear your face mask over your mouth and nose at all times.

**Class Policies**

1. **Attendance & Absences –** Students are expected to attend and sit through the entire class meetings. In case of an absence, the student is responsible to arrange for notes and missed announcements.
2. **Assignment Completion & Late Work** – Late assignment submission is not allowed, unless a permission is granted by the instructor prior to the deadline. Students should submit their work by the due date.
3. **Academic Conduct Code** – Cheating and plagiarism will not be tolerated in any Metropolitan College course. They will result in no credit for the assignment or examination and may lead to disciplinary actions. Please take the time to review the Student Academic Conduct Code:

[**http://www.bu.edu/met/metropolitan\_college\_people/student/resources/conduct/code.html**](http://www.bu.edu/met/metropolitan_college_people/student/resources/conduct/code.html)**.**

**NOTE:** [This should not be understood as a discouragement for discussing the material or your particular approach to a problem with other students in the class. On the contrary – you should share your thoughts, questions and solutions. Naturally, if you choose to work in a group, you will be expected to come up with more than one and highly original solutions rather than the same mistakes.]

**Grading Criteria**

* Midterm 35% - Section A1 - Tuesday, March 29th // Section A2: Wednesday March 30th (tentative)
* Final Exam 35% - Section A1 - Tuesday, May 10th // Section A2: Wednesday, May 11th
* Assignments 30% - Check Blackboard for due dates

**Class Meetings, Lectures & Assignments**

*Lectures, Readings, and Assignments subject to change, and will be announced in class as applicable within a reasonable time frame.*

|  |  |  |
| --- | --- | --- |
| **Lecture** | **Topic** | **Description** |
| 1 | 1.1 | Mathematical preliminaries |
| 2 | 1.2, 2.1, 2.2, 2.3 | Basic concepts of languages, grammars and automata - DFAs - NDFAs and equivalence |
| 3 | 3.1, 3.2, 3.3 | Regulars expressions, regular grammars |
| 4 | 4.1, 4.2, 4.3 | Basic properties of regular languages |
| 5 | 5.1, 5.2, 5.3 | Context-Free (CTF) languages. Parsing and Ambiguity. Programming languages |
| 6 | 6.1, 6.2, 7.1 | Simplification of CTF grammars. Normal forms. |
| 7 | 7.2, 7.3, 7.4 | Pushdown automata (PA). Nondeterministic & deterministic PA. PA and CTF languages. |
|  | Midterm Examination | |
| 8 | 8.1, 8.2 | Discuss exam. Properties of CTF languages. Pumping lemmas. Properties. |
| 9 | 9.1, 9.2, 9.3 | Turing Machines (TM). Standard TMs, Turing Thesis. |
| 10 | 10.1, 10.2, 10.4, 10.5 | Models of TMs (option stay, Semi-infinite tape, off-line, Multitape, Multidimensional, Nondeterministic, universal). Linear bounded Automata. |
| 11 | 11.1, 11.2, 11.3, 11.4 | Hierarchy of formal languages and Automata. Recursive and Recursively Enumerable Languages. |
| 12 | 12.1, 12.2, 13.1 | Limits of Algorithmic Computation. Problems that cannot be solved by TMs. Undecidable Problems for Recursively Enumerable Languages. Other Models of computation |
|  | Final Examination | |