Welcome to CS2200: Languages, Machines and Computation

Why Study LMC?

What the Course is About

Administrivia

Why study LMC?

Foundational Course for..

- Compiler Design, Paradigms of Programming
  - Research in Programming Language Design, Automated Program Verification, Program Analysis…
- Complexity Theory

Practical Applications

- Models many important pieces of software and hardware
- Software for scanning large bodies of text: Search utility in editors, grep in Linux, Web Crawling by search engines.
  - Modelled by *Regular Expressions*
- Hardware circuits, communication protocols.
  - Modelled by *Finite Automata*.

Practical Applications

- Syntax of every major programming language is modelled by *Context-Free Grammars.*
  - Theory behind design and construction of major parts of compiler such as Lexical Analyzers and Parsers.

Foundations of Computation

- When developing solutions to real problems, we often confront the limitations of what software can do.
  - *Undecidable* things – no program can do it.
  - *Intractable* things – there are programs, but no fast programs.
- Turing Machines, to understand the fundamental limits of computation.

Abstraction, and a new way of thinking

- Abstraction: Building a mathematical model that captures the essence of the system in the simplest possible way
- In this course, we will use formal definitions for abstract models, and reason about properties of the model purely using the definitions.
  - We will also learn how to formally prove these properties.

What is the course about?

Languages, Machines and Computation

- Languages and Machines are two parallel and independent ways to understand the foundations of Computation.

Machines

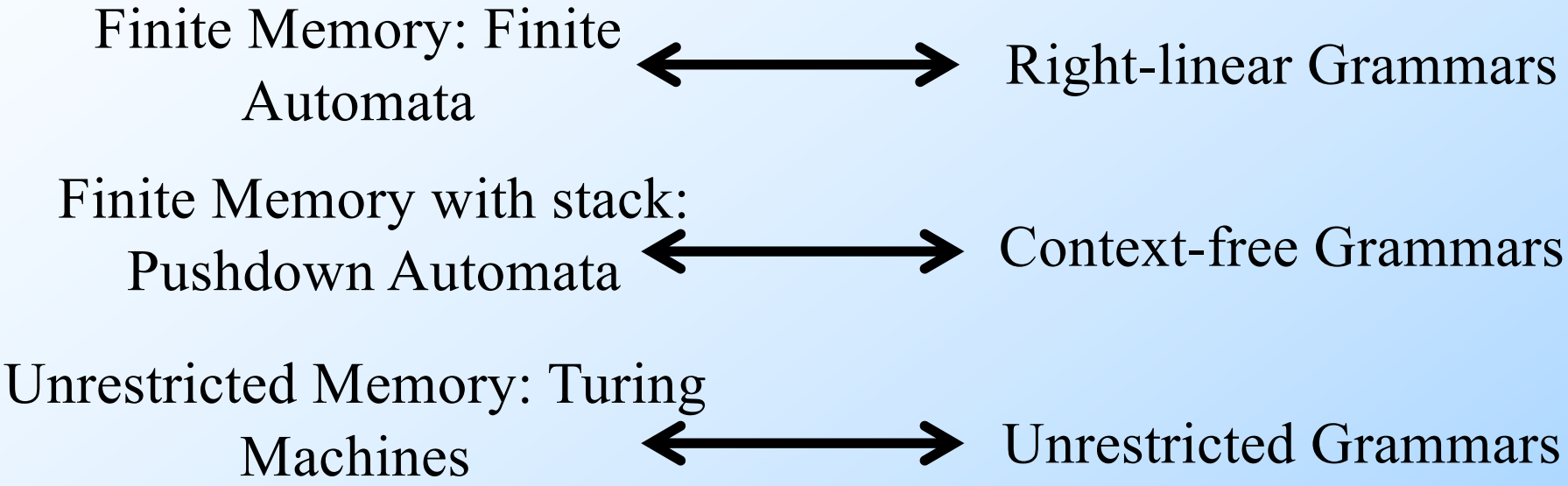
- Abstract Models which capture some fundamental aspect of computation.
- We will study the following three classes of Machines:
  - Finite Memory: Finite Automata
  - Finite Memory with stack: Pushdown Automata
  - Unrestricted Memory: Turing Machines

Languages

- Sets of strings, formalized by grammars
- Chomsky hierarchy: A hierarchy of language classes with increasing complexity
  1. Right-linear Grammars
  2. Context-free Grammars
  3. Unrestricted Grammars

Languages, Machines and Computation

- Problem of deciding whether a sentence belongs to a language has a strong correspondence with computation



Administrivia

Course Details

- Lecture Timings
  - Slot D: Mon 11 AM, Tue 10 AM, Wed 9 AM, Thu 1 PM
  - Online on Google Meet.
- Course webpage: https://kartiknagar.github.io/courses/lmc/
  - Links to Lecture slides and video lectures will be uploaded on the course webpage (after the lecture, typically within couple of hours).
- Course Moodle Page:
  - Tests, Practice Problems, etc. will be uploaded on the moodle page.
- Course Google Group: CS2200-Feb-May-2021
  - I encourage everyone to post any questions, doubts, clarifications, etc.

Course Outline

- Finite Automata and Regular Languages
  - Finite State Automata, Regular Languages, Notion of non-determinism, Subset construction
  - Pattern matching and Regular Expressions, Closure properties of regular languages
  - Limitations of Regular Languages, Pumping Lemma, Myhill-Nerode relations, Quotient Construction, Minimization Algorithm

Course Outline – (2)

- Pushdown Automata and Context-free Languages :
  - Pushdown Automata(PDA), PDA vs CFLs. Deterministic CFLs
  - Grammars and Chomsky Hierarchy, CFLs, Regular Grammars, Chomsky Normal Form, Pumping Lemma for CFLs, Inherent Ambiguity of Context-Free Languages
  - Cock-Younger-Kasami Algorithm, Applications to Parsing

Course Outline – (3)

- Turing Machines and Computability.
  - Introduction to Turing Machines, Configurations, Halting vs Looping.
  - Multi-tape Turing machines, Recursive and Recursively enumerable languages, Undecidability of Halting Problem, Reductions.
  - Introduction to Theory of NP-completeness.

Course Evaluation (tentative)

- Bi-weekly tests : 30%
  - Beginning from the second week.
  - Total 6 tests, 5% per test.
- Endsem : 70%
- Attendance : I won't take attendance, but I encourage everyone to attend all the classes.

Textbooks

- *Automata and Computability*. Dexter C. Kozen. Springer Publishers.
- *Introduction to Automata Theory, Languages, and Computation.* Hopcroft, Motwani, Ullman. Pearson Publishers 3rd Edition.
- *Introduction to the Theory of Computation*. Michael Sipser. 3rd Edition.

Acknowledgments

- The lecture slides are based on Jeffrey Ullman's slides used for the course on Automata and Complexity Theory at Stanford University.