COSE215: Theory of Computation

Lecture 0 — Introduction

Hakjoo Oh
2016 Spring

Basic Information

Instructor: Hakjoo Oh

- **Position:** Assistant professor in Computer Science and Engineering, Korea University
- **Expertise:** Programming Languages
- **Office:** 616c, Science Library
- **Email:** `hakjoo_oh@korea.ac.kr`
- **Office Hours:** 1:00pm–3:00pm Mondays and Wednesdays (by appointment)

TAs:

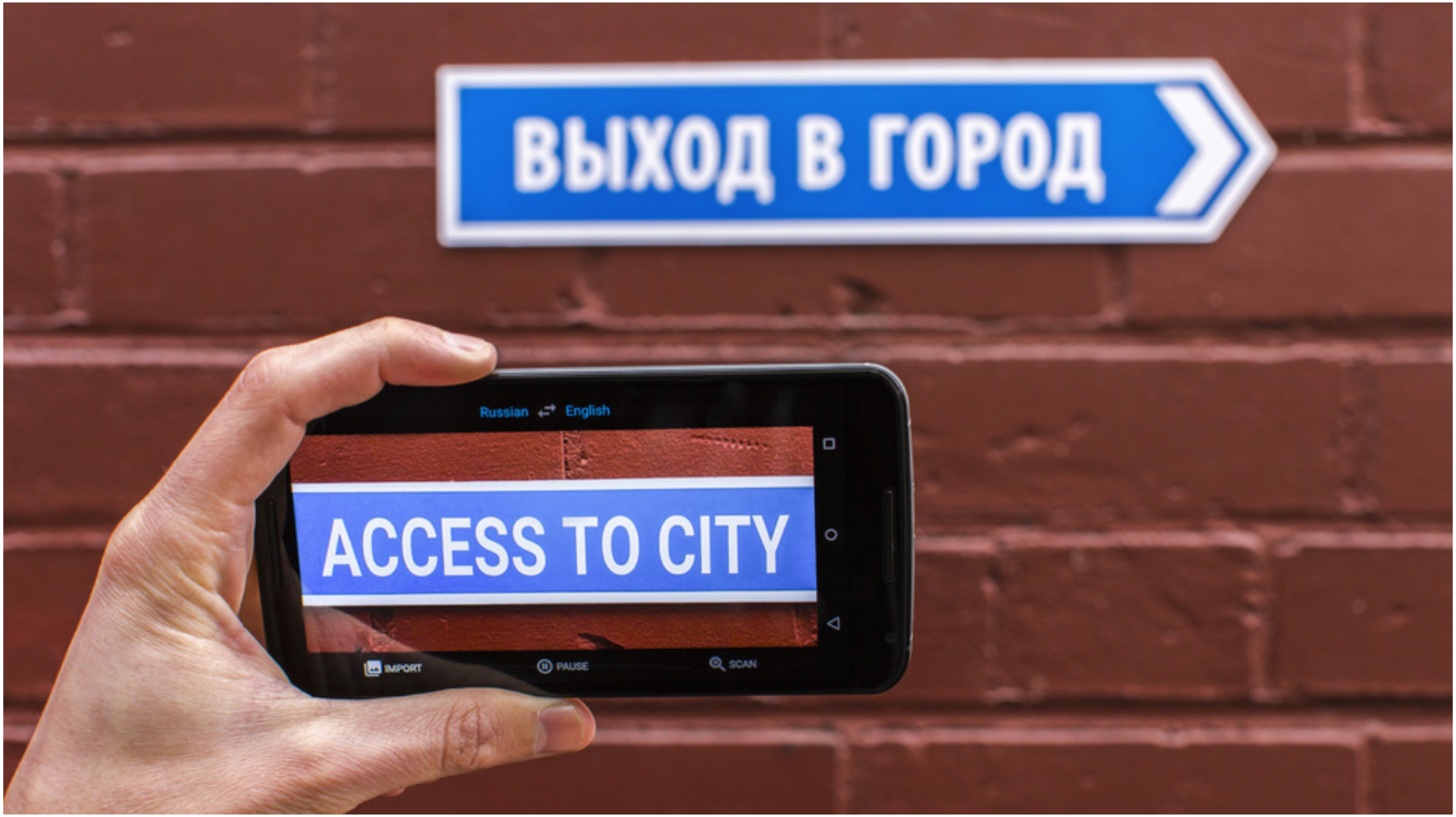
- Sungjoon Hong (`june3371@korea.ac.kr`)
- Mina Lee (`ok_lady@korea.ac.kr`)

Course Website:

- `http://prl.korea.ac.kr/~pronto/home/courses/cose215/2016/`
- Course materials will be available here.

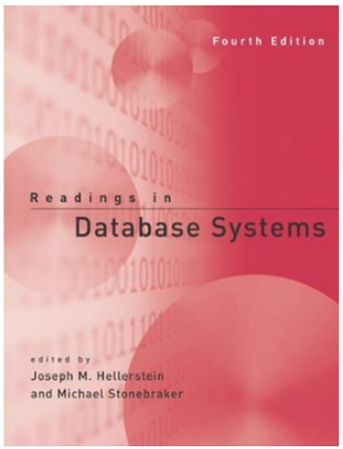
Computer Revolution

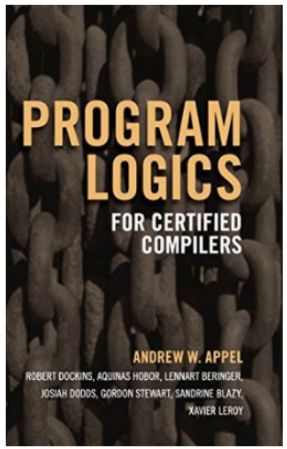
Computer Revolution

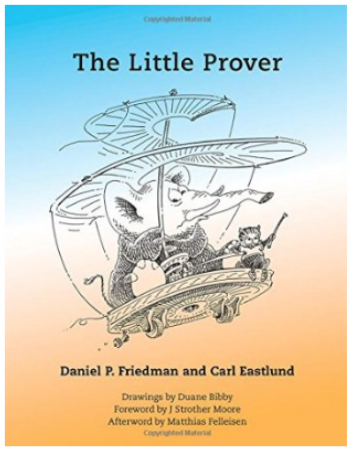


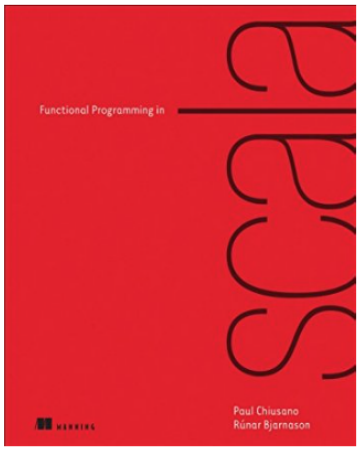
Computer Revolution

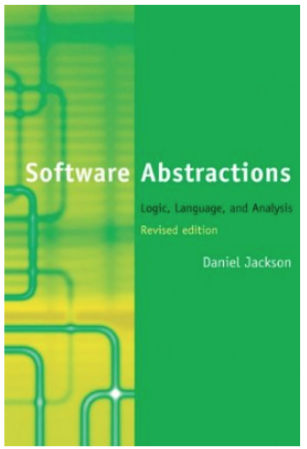
Inspired by Your Shopping Trends See more



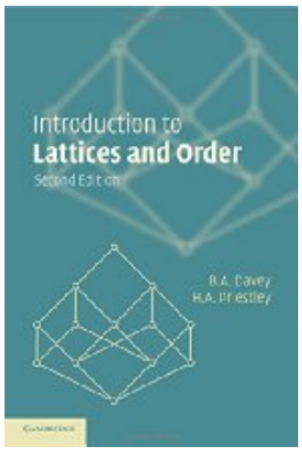


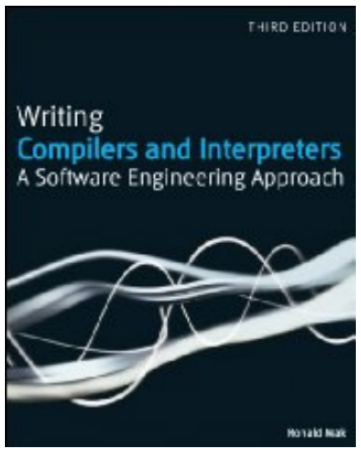


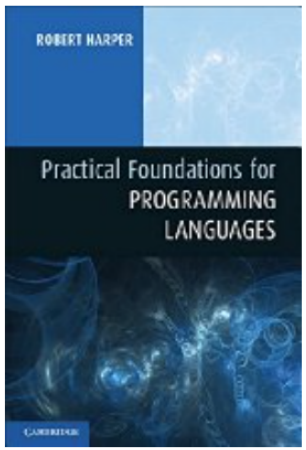


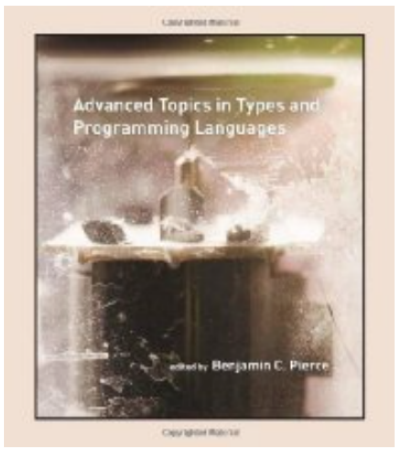


Inspired by Your Browsing History See more



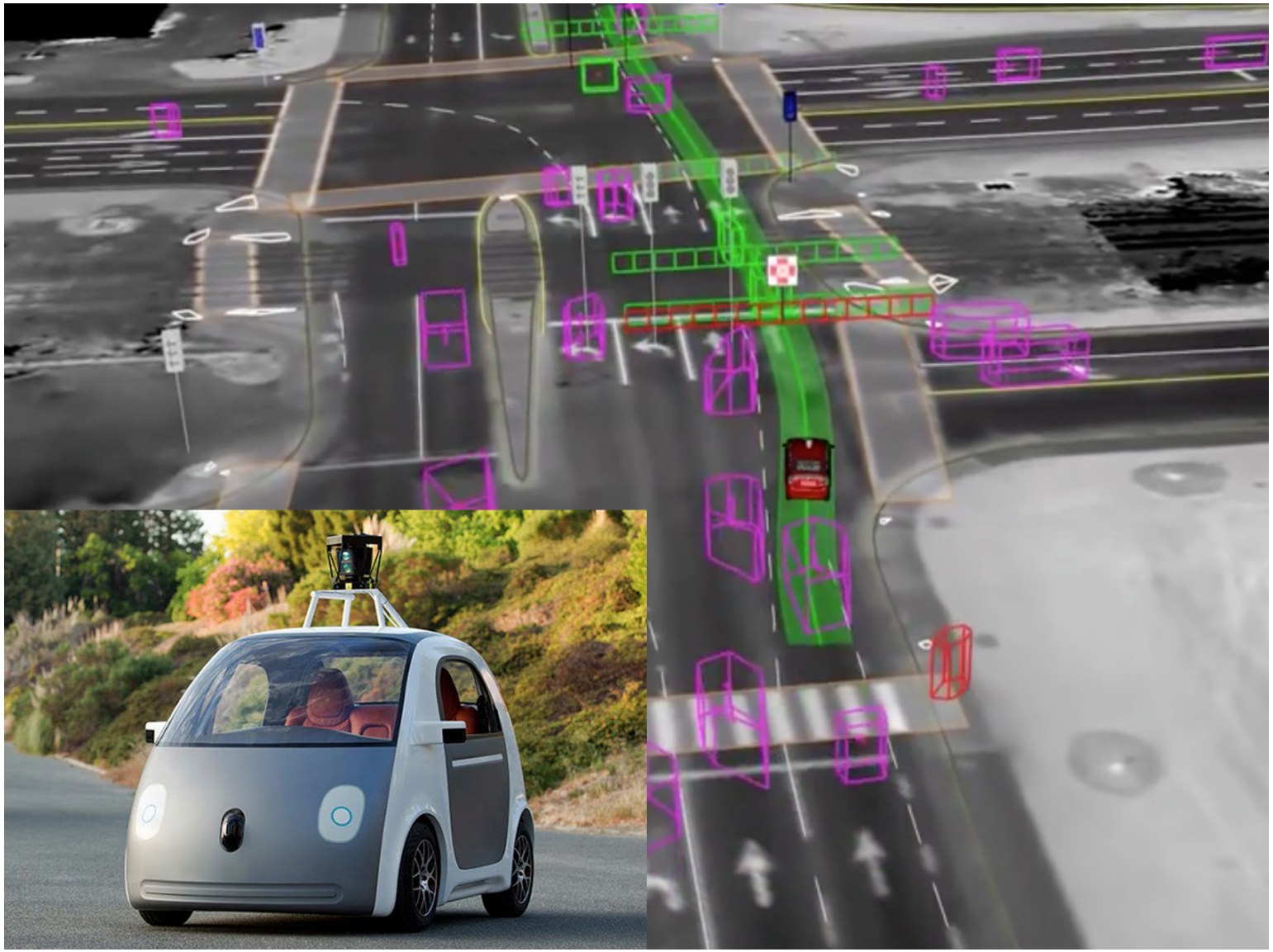






Computer Revolution

Computer Revolution



Computer Revolution

Stanford

Automata

This course covers finite automata, context-free grammars, Turing machines, undecidable problems, and intractable problems (NP-completeness).

Preview Lectures

About the Course

I am pleased to be able to offer free over the Internet a course on Automata Theory, based on the material I have taught periodically at Stanford in the course CS154. Participants have access to screencast lecture videos, are given quiz questions, assignments and exams, receive regular feedback on progress, and can participate in a discussion forum. Those who successfully complete the course will receive a statement of accomplishment. You will need a decent Internet connection for accessing course materials, but should be able to watch the videos on your smartphone.

The course covers four broad areas: (1) Finite automata and regular expressions, (2) Context-free grammars, (3) Turing machines and decidability, and (4) the theory of intractability, or NP-complete problems.

Sessions

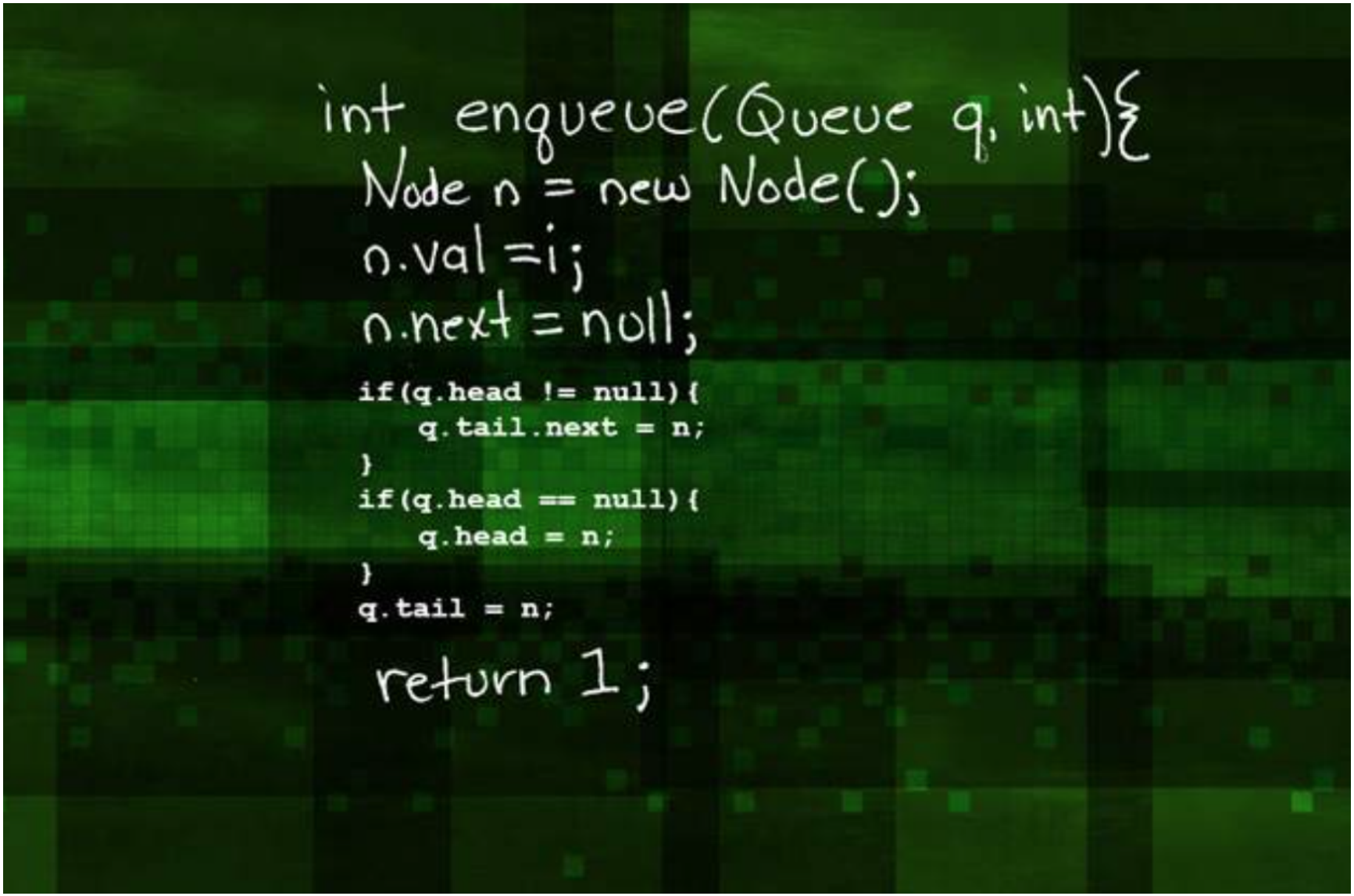
September 1, 2014 - October 26, 2014

Enroll

Course at a Glance

- 6 weeks of study
- 8-10 hours/week
- English

Computer Revolution



Goal: Understand the Rationale behind Computer Revolution

- How is it possible?
- What are computers?
- What is the unique characteristic of computers?
- What can be done by a digital computer?
- What cannot be done by a digital computer?

RoadMap: Towards Understanding Turing Machines

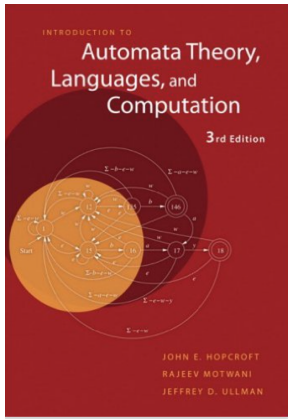
- Turing Machines
  - Decidability, universal Turing machine
- Pushdown Automata
  - Context-free languages and grammars
  - Applications: e.g., compilers, programming languages, natural language processing, webs, etc.
- Finite Automata
  - Regular expressions and languages
  - Applications: text search, pattern matching, etc.

Overview

- **Part 0:** basic concepts, mathematical backgrounds
- **Part 1:** finite automata, deterministic finite automata, nondeterministic finite automata, equivalence, regular languages, regular expressions, regular grammars, connections between regular languages and expressions/between languages and grammars, closure properties, pumping lemma, etc
- **Part 2:** context-free grammars/languages, parsing and ambiguity, normal forms, nondeterministic pushdown automata, relation with context-free languages, deterministic pushdown automata, pumping lemmas, closure properties, decision algorithms
- **Part 3:** turing machines, standard turing machine, Turing's thesis, variations of Turing machines, nondeterministic Turing machines, universal Turing machine, recursively enumerable languages, computability, decidability, halting problem, reduction, recursive functions, complexity, P/NP

Course Materials

- Self-contained slides will be provided.
- Reference:



John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman. Introduction to automata theory, languages, and computation. Third edition.

Grading

- 6 homework assignments – 30%
- Midterm exam: (in class, 75 minutes) – 30%
- Final exam: (in class, 75 minutes) – 30%
- Attendance and participation – 10%

Assignments policy:

- You have roughly one and half weeks for each homework assignment.
- All the assignments must be as a stapled printout, in class, on the due date *before* lecture begins. **No late submissions will be accepted.**
- The writing must be clear and legible. What cannot be read/understood will not be graded.
- All homework assignments must be your own work.

Schedule (tentative)

Week 1	Introduction
Week 2	Finite Automata
Week 3	Regular Languages
Week 4	Properties of Regular Languages
Week 5	Context-free Languages
Week 6	Simplifications and Normal Forms
Week 7	Pushdown Automata
Week 8	Mid-term exam
Week 9	Properties of Context-free Languages
Week 10	Turing Machines
Week 11	Other Models of Turing Machines
Week 12	A Hierarchy of Formal Languages
Week 13	Limits of Algorithmic Computation
Week 14	(optional) Other Models of Computation
Week 15	(optional) Computational Complexity
Week 16	Final exam

Next Class

- Icebreaking: Introduce yourself
- Mathematical backgrounds and notation

Icebreaking

Be prepared to introduce yourself ($< \mathbf{1}$min):

- Free format. Say anything.
- Nothing to talk about? major, grade, interests, hobbies, specialty, goal, motivation for this course, what you expect from this course, etc