COSE215: Theory of Computation

Hakjoo Oh

Spring, 2016

Meeting Time	09:00AM–10:15AM, Tuesdays and Thursdays
Classroom	202

Instructor: Hakjoo Oh

- **Position:** Assistant professor in Computer Science and Engineering, Korea University
- **Expertise:** Programming languages, static program analysis, compilers
- **Office:** 616c, Science Library
- **Email:** `hakjoo.oh@gmail.com`
- **Office Hours:** 1:00pm–3:00pm Mondays and Wednesdays (by appointment)

Course motivation To be a well-founded computer scientist, you should be ready to answer the following fundamental questions:

- What can a digital computer do at all?
- What is the limits of computation?
- What can a computer do efficiently?

If you cannot answer the questions precisely, you should take this class.

Course overview This course introduces fundamental principles on computing. The central goal is to learn a hierarchy of abstract models ("automata") of computation and investigate what can be done and what cannot be done by digital computers. Not only do they form a theoretical foundation of computer science, they also have practical significance in many branches of computer science, e.g. compilers, software engineering, concurrent systems, etc. Main topics:

- **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:

- John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman. Introduction to automata theory, languages, and computation. Third edition.
- Self-contained slides will be provided.

Grading:

- 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.

To be successful in this course, you are expected to review course materials (textbook/slides) before and after each lecture. If any student needs some helps, feel free to contact the instructor or TAs.

Academic honesty: All homework assignments must be your own work.

Tentative Schedule:

Weeks	Topics
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