CS-E4530

Computational Complexity Theory

Janne H. Korhonen
Aalto University
Spring 2018

Lecture A1:

Midterm 1 Recap

Turing Machines

- **Turing machines**
  - No need to remember all the details of the definition, but you should understand the general idea
  - You can do high-level mathematical proofs involving Turing machines and their existence

- **Concepts related to Turing machines**
  - Running time and space usage
  - Functions computed by Turing machines
  - Executions, halting and not halting
  - Transforming Turing machines to obtain new machines
  - Universal Turing machine

Decisions Problems

- **Decision problems**
  - You understand what is a decision problem
  - A set of instances and a question
  - Encoding objects as binary strings and decision problems over mathematical objects
  - Decidable problems
  - Undecidable problems, the halting problem

Reductions

- **Many-to-one reductions**
  - Polynomial-time many-to-one reductions
  - You understand the basic ideas of using reductions to prove upper and lower bounds
  - Composing reductions with other reductions and algorithms
  - Using reductions from the halting problem to prove undecidability

Complexity Classes

- **Class P**
  - Polynomial-time Turing machines
  - You can prove that a problem is in P
- **Class NP**
  - Polynomial-time verifiers
  - Polynomial-size certificates for yes-instance
  - You can prove that a problem is in NP
  - NP-hardness and NP-completeness

NP-complete Problems

- **Existence of NP-complete problems**
  - You know that there are NP-complete problems
  - CNF-SAT is NP-complete
  - No need to remember any specific NP-complete problem, but it's good have some idea what some NP-complete problems look like

- **Proving NP-completeness**
  - You can construct polynomial-time reductions
  - You can prove correctness of a reduction
  - **Spoiler:** this will be in the exam!