Aalto University
School of Science

CS-E4530 Computational Complexity Theory

Midterm 1 Recap: Lectures 1–6

Aalto University
School of Science
Department of Computer Science

Spring 2019

Lecture 0: Introduction and Overview
Lecture 1: Problems, Algorithms, Complexity, Reductions

- Topics:
  - Problems vs. algorithms
  - Efficiency of algorithms, complexity of problems
  - Rates of growth
  - Example problems
  - Reductions between problems
- Notes:
  - These lectures mainly aim to provide some "big picture" discussion and a few examples of how computational complexity questions arise in common settings.
  - However, on the technical side you should be familiar and comfortable with the concept of function growth rates, and the different varieties of the O-notation.

Lecture 2: Turing Machines, Decision Problems, Languages

- Topics:
  - Modelling computation
  - Turing machines
  - Formal languages and decision problems
  - Time and space complexity, complexity class P
- Notes:
  - All the technical material in this lecture is essential, except the detailed constants in the alphabet- and tape-reduction theorems. (Because the specific constants don't matter anyway. However you should know the overhead *rates* of the different simulations.)
  - You should also be able to comprehend and design simple Turing machines, at least in the single-tape model and the difficulty level of the successor-computing example in the lecture. You don't need to remember all the details of the formalism, but should be comfortable with the transition table and the diagram notations.
  - Also make sure that you know and understand the concepts of decidability, undecidability and semidecidability.

Lecture 3: Representations, Universality, Undecidability

- Topics:
  - Decision problems
  - Instance (= input) representations
  - Turing machine representations
  - The universal Turing machine
  - Undecidability
- Notes:
  - Again all the technical material in this section is essential.
  - For present purposes you do not need to remember the details of the design of a universal Turing machine, although a general understanding of the implementation idea is useful. However you must know what a universal machine *is*. (I.e. the statement of the Universal Turing Machine theorem.)
  - You must also be familiar, understand, and able to reproduce the proof of the undecidability of the Halting Problem.

Lecture 4: Reductions

- Topics:
  - Many-to-one reductions
  - Example: Graph colouring
  - Turing reductions
  - Closure and completeness
- Notes:
  - You must be familiar and comfortable with the concepts of many-to-one and Turing reductions and the associated ideas (transitivity, hardness, completeness, closure of language classes, using reductions in combination with solution algorithms).
  - You should understand the Graph Colouring reduction examples, but you do not need to memorise them. (At this point the ideas are important, more examples are forthcoming later.)

Lecture 5: NP and Nondeterminism

- Topics:
  - Polynomial-time verifiers
  - Examples of polynomial-time verifiers
  - The language class NP
  - Nondeterministic Turing Machines
  - NP-completeness
- Notes:
  - You must be familiar and comfortable with the definition of an NP-type problem, both in the polynomial-time verifier and the nondeterministic-machine framework. For an "obviously" NP-type problem you should be able to justify the existence of an efficient verifier.
  - You should be familiar with and able to reproduce the definition of each of the example problems presented in the lecture.
  - You should also be familiar with and able to reproduce the definitions of all the complexity classes introduced in the lecture: P, EXP, NTIME(T(n)), NP.
  - You should be familiar and comfortable with the notions of NP-completeness and NP-hardness, and how to establish these properties via reductions.
  - You should be familiar with the "artificial" NP-complete language TMSAT, and how to prove its NP-completeness.

Lecture 6: The Cook–Levin Theorem

- Topics:
  - Boolean satisfiability
  - CNF formulas and Boolean functions
  - The Cook–Levin theorem
- Notes:
  - You should be familiar with the concepts of Boolean formulas, their satisfiability, conjunctive normal forms, and the problems CNF-SAT and k-SAT.
  - You should be familiar with the statement of the Cook-Levin Theorem, and the general idea of representing the computation of a Turing machine M on an input x in terms of a formula φ_x^M. However you do not need to be able to reproduce the proof of the theorem. (Even though a general understanding of the essential ideas involved is highly commendable.)