Theory of Computer Science

M. Helmert, G. Röger
F. Pommerening
Spring Term 2015

University of Basel
Computer Science

Exercise Sheet 12

Due: Wednesday, May 20, 2015

Note: Submissions that are exclusively created with LaTeX will receive a bonus mark. Please submit only the resulting PDF file (or a printout of this file).

Note: This is the last exercise sheet of the lecture. This means there is a total of 116 points (without bonus points). As discussed, 50% of those points (58 points) are required to take the exam.

Exercise 12.1 (Algorithms for SAT, 1.5+1.5 Points + 1 Bonus Point)

The *Satisfiability problem in propositional logic* (SAT) is defined as follows: given a propositional formula φ, is φ satisfiable?

(a) Specify a non-deterministic algorithm for SAT, whose runtime is limited by a polynomial in the length of φ. Explain why the algorithm's runtime is polynomial.

(b) Specify a deterministic algorithm for SAT.

(c) *Bonus exercise:* Estimate the runtime of your algorithm from part (b) in O-Notation.

You can use any common programming concepts in your answer. You do *not* have to use the restricted syntax of WHILE-programs or similar languages. High-level pseudo code is sufficient as long as it can be easily seen that each step runs in polynomial time. Use the SAT statements from the lecture for non-deterministic statements.

Exercise 12.2 (P vs. NP, 1+1+1.5+1.5 Points)

Prove or refute the following statements. In all cases, specify a short proof (2–3 sentences are sufficient).

(a) Let X be an NP-hard problem and Y a problem with $X \leq_{\mathrm{p}} Y$. Then Y is NP-hard.

(b) Let X be an NP-hard problem. If there is a polynomial algorithm for X, then there is a deterministic polynomial algorithm for DirHamiltonCycle.

(c) There are NP-complete problems X and Y where there is a deterministic polynomial algorithm for X but not for Y.

(d) Let $Y \subseteq \Sigma^*$ be any problem with $Y \neq \emptyset$ and $Y \neq \Sigma^*$. Then $X \leq_{\mathrm{p}} Y$ holds for all $X \in \mathrm{P}$.

Exercise 12.3 (Polynomial Reduction, 2 Points + 1 Bonus Point)

A *Hamilton path* is defined analogously to a Hamilton circle (see chapter 19.3) with the only difference that we look for a simple path instead of a circle. More formally: a Hamilton path in a directed graph $\langle V, E\rangle$ is a sequence of vertices $\pi = \langle v_1, \dots, v_n\rangle$ that defines a path ($\langle v_i, v_{i+1}\rangle \in E$ for all $1 \leq i < n$) and contains every vertex in the graph exactly once.
Consider the decision problem DirHamiltonPathWithEndPoints:

- *Given:* directed graph $G = \langle V, E\rangle$, start vertex $v_\mathrm{s} \in V$, end vertex $v_\mathrm{e} \in V$
- *Decide:* Is there a Hamilton path from v_s to v_e in G, i.e., a Hamilton path $\pi = \langle v_1, \dots, v_n\rangle$ with $v_1 = v_\mathrm{s}$ and $v_n = v_\mathrm{e}$?

(a) Prove that DirHamiltonPathWithEndPoints is NP-hard. You can use without proof that DirHamiltonCycle is NP-complete.

(b) *Bonus exercise:* Is DirHamiltonPathWithEndPoints NP-complete? Justify your answer.