Theory of Computer Science

M. Helmert
F. Pommerening
Spring Term 2016

University of Basel
Computer Science

Exercise Sheet 13

Due: Wednesday, June 1, 2016

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 regular exercise sheet of the lecture. Next week there will be another exercise sheet that consists only of bonus exercises and covers chapters E3–E5. This means there is a total of 120 points (without bonus points). As discussed, 50% of those points (60 points) are required to take the exam.

Exercise 13.1 (Algorithms for CLIQUE, 1.5+1.5 Points + 1 Bonus Point)

Consider the decision problem CLIQUE:

- *Given:* undirected graph $G = \langle V, E \rangle$, number $K \in \mathbb{N}_0$
- *Question:* Does G contain a clique of size K or more, i.e., a set of nodes $C \subseteq V$ with $|C| \geq K$ and $\{u, v\} \in E$ for all $u, v \in C$ with $u \neq v$?

(a) Specify a non-deterministic algorithm for CLIQUE, whose runtime is limited by a polynomial in $|V| + |E|$. Explain why the algorithm's runtime is polynomial.

(b) Specify a deterministic algorithm for CLIQUE.

(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 GUESS statements from the lecture for non-deterministic statements.

Exercise 13.2 (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 deterministic polynomial algorithm for X, then there also 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 13.3 (Polynomial Reduction, 2 Points + 1 Bonus Point)

A *Hamilton path* is defined analogously to a Hamilton cycle (see chapter E1) with the only difference that we look for a simple path instead of a cycle. 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 DirHamiltonPath:

- *Given:* directed graph $G = \langle V, E \rangle$
- *Question:* Does G contain a Hamilton path?

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

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