Lehr- und Forschungsgebiet
Mathematische Grundlagen der Informatik
RWTH Aachen
Prof. Dr. E. Grädel, D. Fischer, T. Ganzow,
B. Puchala

WS 09/10

Complexity Theory and Quantum Computing — Assignment 1

Due: Monday, November 02, 12:00

Exercise 1

Prove the following facts.

(a) $p(n) = o(n^{\log n})$ for any polynomial p.

(b) $n^{(\log n)^k} = o(2^n)$ for any natural number k.

(c) If $f \sim g$ and $h = o(g)$, then $f + h \sim g$.

(d) If $f = \Theta(g)$ and $h = o(g)$, then $f + h = \Theta(g)$.

Exercise 2

A constant challenge within algorithmics is the *traveling salesman problem* (TSP): Given a graph with n vertices (cities) with distances $d_{ij} \in \mathbb{N}$ for $i, j \in \{1, \dots, n\}$, we want to construct a shortest tour. Where a *tour* is a path which contains all cities. The corresponding decision problem TSP(D) asks, given a graph and a natural number k, whether there is a tour of length at most k. For none of these problems a polynomial time algorithm is known.

Prove that the existence of a polynomial time algorithm for the decision problem TSP(D) implies that constructing an optimal tour can be done in polynomial time as well.

Hint: First, use the given algorithm for TSP(D) to determine the length of an optimal tour in polynomial time. Then, for each edge, test whether there is an optimal tour that does not use this edge.

Exercise 3

A 2D-Turing-machine uses a two-dimensional infinite work memory $\mathbb{N} \times \mathbb{N}$ instead of just a single linear tape. On that memory, the reading and writing head can be moved left, right, up and down.

(a) Give a formal definition for this machine model and the corresponding notion of computation.

(b) Give a rough proof for the following fact. A decision problem can be solved by a 2D-Turing-machine in polynomial time if and only if, the problem can be solved by a usual Turing-machine with a bounded number of tapes in polynomial time.