Technische Universität München (I7)
Dr. J. Kreiker / Dr. M. Luttenberger

Summer term 2010
1. Juli 2010, 12:30

Complexity Theory – Homework 10

Discussed on 06.07.2010.

Exercise 10.1

Show that, if SAT $\in \mathbf{PCP}(r(n), 1)$ for some $r(n) = o(\log n)$, then $\mathbf{P} = \mathbf{NP}$.

Exercise 10.2

Prove that QUADEQ is **NP**-complete.

Exercise 10.3

Consider the following problem:

Input: A matrix $A \in \mathbb{Q}^{m \times n}$, a vector $b \in \mathbb{Q}^m$.
Target: Determine the maximal number of equations in $Ax = b$ which can simultaneously be satisfied by some $x \in Q^n$.

Show that there is a constant $\rho < 1$ such that approximating the maximal size is **NP**-hard.

Exercise 10.4

We consider the optimization variant of the KNAPSACKPROBLEM:

Input: Values $v_1, \ldots, v_n$, weights $w_1, \ldots, w_n$ and a weight bound W, all natural numbers representable by n bits.
Target: Compute the maximal total value attainable by any selection S of total weight at most W, i.e.,

$$\max\{\sum_{i \in S} v_i \mid S \subseteq \{1, 2, \ldots, n\} \wedge \sum_{i \in S} w_i \leq W\}.$$

(a) In Exercise 3.2(c) we have discussed a pseudo-polynomial algorithm which solves this problem in time $\mathcal{O}(nW)$. Similarly, design an algorithm which finds the maximal total value by computing an array A with

$$A[j, v] = \min\{W + 1, \sum_{i \in S} w_i \mid S \subseteq \{1, 2, \ldots, j\} \wedge \sum_{i \in S} v_i = v\}.$$

Your algorithm should be polynomial in n and $V := \sum_{i=1}^{n} v_i$.

(b) Assume you replace all values v_i by $v_i' := \lfloor v_i / 2^k \rfloor$ for some fixed $k \geq 0$, i.e., you remove the k least significant bits. The weights w_i and the weight limit W stay unchanged. Let v_{opt}, resp. v'_{opt} be the optimal value for the original resp. reduced instance.

We take $v'_{\text{opt}} \cdot 2^k$ as an approximation for v_{opt}.

- Show that $v_{\text{opt}} \geq v'_{\text{opt}} 2^k$. What is the approximation error in the worst case?
- Choose k s.t. the approximation error is at most $\epsilon > 0$. Show that for this k the algorithm runs in time polynomial in n and $1/\epsilon$.