Martin Ziegler
Carsten Rösnick

Distributed on 1.12.2011
Solutions to be submitted by 8.12.2011 14h25

Complexity Theory

WS 2011/2012, Exercise Sheet #6

EXERCISE 15:
Prove:

a) Every PSPACE-hard problem is also $\mathcal{NP}$-hard.

b) If some PSPACE-complete problem belongs to $\mathcal{NP}$, it follows $\mathcal{NP} = \mathsf{PSPACE}$.

c) If every $\mathcal{NP}$-hard problem is PSPACE-hard, it follows $\mathcal{NP} = \mathsf{PSPACE}$.

Let co$\mathcal{NP}$ denote the class of all problems whose complement belongs to $\mathcal{NP}$; cmp. Exercise 11e).

d) Define formally "co$\mathcal{NP}$–hard" and "co$\mathcal{NP}$–complete", then prove:

e) The two classes of problems $\mathcal{NP}$–complete and co$\mathcal{NP}$–complete, respectively, are either disjoint or coincide.

A problem A is *polynomial-space reducible* to B if there exists a total function $f : \{0,1\}^* \to \{0,1\}^*$ computable in polynomial space such that $\vec{x} \in A \Leftrightarrow f(\vec{x}) \in B$.

f) Prove: $A \in \mathsf{PSPACE}$ is polynomial-space reducible to every $\emptyset \subsetneq B \subsetneq \Sigma^*$.

EXERCISE 16:
Recall the two player game GraphGame on directed graphs from the lecture and the question for a winning strategy.

a) Prove: Either the first or the second player has a winning strategy.
Which properties of the game have you used?

b) Describe and analyze a recursive algorithm deciding within polynomial space the truth of a given quantified Boolean formula.

c) Prove GraphGame$\in$PSPACE.

EXERCISE 17:
Let #SAT denote the function problem of calculating, given a CNF formula, how many satisfying it has. For a nondeterministic Turing Maschine M and input $\bar{x} \in \Sigma^*$, write $\#M(\bar{x})$ for the number of accepting computations of M on $\bar{x}$. Furthermore

$$\#\mathcal{P} \;:=\; \big\{ f_M \,\big|\, M \text{ polynomial-time NTM},\; f_M : \Sigma^* \to \mathbb{N}_0,\, \bar{x} \mapsto \#M(\bar{x}) \big\}$$

a) Prove that every $f \in \#\mathcal{P}$ can be computed within polynomial space.

b) Define "#$\mathcal{P}$–hard" and "#$\mathcal{P}$–complete".
Which notion of reduction is appropriate for such *counting* problems?

c) Prove: #SAT is #$\mathcal{P}$–complete. (Hint: Proof of Cook-Levin.) How about #3SAT?