CS 710: Complexity Theory 10/11/2011

Homework 1

Instructor: Dieter van Melkebeek

This homework is due at the beginning of class on 11/8/2011. Good luck!

1. Recall that the exact vertex cover problem consists of deciding whether the minimum size of a vertex cover of a given graph G is exactly a given integer k.

   Show that this problem is $\leq_m^p$-complete for the class $\{L_1 \setminus L_2 \mid L_1, L_2 \in \mathrm{NP}\}$, where "$\setminus$" denotes set difference.

2. We mentioned that there is no good hierarchy known for BPTIME. The goal of this problem is to establish a hierarchy for the corresponding notion for promise problems, namely promise-BPTIME.

   Promises are conditions on the inputs to a computational problem. They allow us to express that we only care about the behavior of the algorithm on certain inputs – those that satisfy the promise. A promise decision problem over an alphabet Σ can be formalized as two disjoint sets, namely a set $I_p \subseteq \Sigma^*$ of positive instances and a set $I_n \subseteq \Sigma^*$ of negative instances. The corresponding promise is that the inputs belong to $I_p \cup I_n$; we do not care about inputs outside of that union (if any). We say that a randomized algorithm solves the promise decision problem if the algorithm accepts every input from I_p with probability at least 2/3 and rejects every input from I_n with probability at least 2/3. On inputs outside of the promise, the randomized algorithm may have a probability of acceptance in the range $(1/3, 2/3)$. The class promise-BPP consists of all promise decision problems that can be solved by randomized algorithms that run in polynomial time.

   Show that if t' and t are two time bounds with t time constructible and "somewhat larger" than t', then there exists a promise decision problem that can be solved by a randomized algorithm that runs in time t but not by a randomized algorithm that runs in time t'. Quantify "somewhat larger" as tightly as you can.

3. Construct oracles A, B, C, D such that

   (a) $\mathrm{BPP}^A \neq \mathrm{NEXP}^A$ but $\mathrm{BPP}^B = \mathrm{NEXP}^B$, and

   (b) $\mathrm{PH}^C \neq \mathrm{PSPACE}^C$ but $\mathrm{PH}^D = \mathrm{PSPACE}^D$.

   Assume that the oracle tape does count towards the space bound.

4. Show that NEXP/poly = coNEXP/poly. What about NP/poly = coNP/poly?

5. Show that for any space constructible $s(n) \geq \log n$, $\mathrm{ASPACE}(s(n)) = \cup_{c>0}\mathrm{DTIME}(2^{cs(n)})$.

6. (a) Show that if NP is in BPP then PH is in BPP.

   (b) What is wrong with the following purported proof of (a)? If NP is in BPP then

   $$\Sigma_2^p = \mathrm{NP}^{\mathrm{NP}} \subseteq \mathrm{NP}^{\mathrm{BPP}} \subseteq \mathrm{BPP}^{\mathrm{BPP}} = \mathrm{BPP} \subseteq \Pi_2^p,$$

   so PH collapses to the second level and lies in BPP.