CS 710: Complexity Theory 11/8/2011

Homework 2

Instructor: Dieter van Melkebeek

This homework is due at the beginning of class on 12/1/2011. Good luck!

1. We don't know whether EXP $\not\subseteq$ P/poly or even whether NEXP $\not\subseteq$ P/poly. However, prove the following for every positive constant c:

   (a) EXP $\not\subseteq$ P/n^c, and

   (b) NEXP $\not\subseteq$ $\mathrm{P}^{\mathrm{NP}[n^c]}/n^c$,

   where $\mathrm{P}^{\mathrm{NP}[f(n)]}$ denotes all languages that can be decided in polynomial time with access to an oracle for a language in NP such that no more than $f(n)$ oracle queries are made on an input of length n.

2. Show that the problem of deciding whether an arithmetic circuit over the integers is identically zero $\leq_m^p$-reduces to the problem of deciding whether an arithmetic circuit over the integers evaluates to zero on a given input.

3. We showed in class that randomized decision procedures with bounded error can be simulated by Σ_2-machines with roughly a quadratic overhead in time. Is is open whether these simulations can be made subquadratic. This problem shows that we can get subquadratic simulations on Σ_3-machines.

   (a) Consider the explicit expander construction we mentioned in class and label the incident edges at each of the N vertices by the corresponding affine transformation.

   Given a start vertex ρ_0 and a sequence of s labels, show how to compute the vertex ρ_s at the end of the walk in quasilinear time, i.e., in time $n \cdot (\log n)^{O(1)}$, where $n = s + \log N$. You can use the fact that integer multiplication can be computed in quasilinear time.

   (b) Show that $\mathrm{BPTIME}(t) \subseteq \Sigma_3\mathrm{TIME}(t \cdot (\log t)^{O(1)})$.

4. You are given oracle access to a Boolean function on n variables and want to get an estimate of the fraction μ of inputs that map to true.

   For any positive constants δ and ϵ, give a randomized algorithm that outputs an estimate which, with probability at least $1 - \delta$, differs from μ by no more than ϵ. Your algorithm should use no more than $n + O(\log \frac{1}{\delta})$ random bits, make $O(\frac{1}{\epsilon^2} \log \frac{1}{\delta})$ oracle calls, and run in time polynomial in n, $\frac{1}{\epsilon}$ and $\log \frac{1}{\delta}$.

5. A universal traversal sequence (UTS) for size n is a sequence σ of labels from $\{1, 2, \ldots, n-1\}$ such that for any undirected connected graph G with n vertices in which the incident edges at every vertex have been assigned distinct labels from $\{1, 2, \ldots, n-1\}$, the following process always visits every vertex of G: Pick an arbitrary start vertex, and in subsequent steps, go along the edge with the label matching the next symbol of σ; in case of no match, stay put during that step and continue with the next symbol of σ.

   (a) Show that there exists a UTS for size n that is of length $n^{O(1)}$.

(b) Show that we can find a UTS with the parameters as in part (a) in polynomial time if there exists a language L in E such that $\mathrm{BP}_L(n) = 2^{\Omega(n)}$.

(c) Show how to construct a UTS for size n of size $n^{O(\log n)}$ in time polynomial in the output.