6.841: Advanced Complexity Theory Spring 2016

Problem Set 1

Due Date: February 23rd, 2016

Turn in your solution to each problem on a separate piece of paper. Mark the top of each sheet with the following: (1) your name, (2) the question number, (3) the names of any people you worked with on the problem, or "Collaborators: none" if you solved the problem individually. We encourage you to spend time on each problem individually before collaborating!

Problem 1 – Equivalent Definitions of the Polynomial Heirarchy

In class, we defined the polynomial heirarchy with quantifiers. We said language L is in Σ_2^{P} if there exists a polynomial time TM M and a polynomial q such that:

$$x \in L \iff \exists u_1 \in \{0,1\}^{q(|x|)} \forall u_2 \in \{0,1\}^{q(|x|)} M(x, u_1, u_2) = 1$$

Show that $\Sigma_2^{\mathsf{P}} = \mathsf{NP}^{\mathsf{NP}}$.

Problem 2 – $\mathsf{SPACE(n)}$ vs. NP

Show that $\mathsf{SPACE(n)} \neq \mathsf{NP}$. Hint: Use the padding argument from Lecture 1.

Problem 3 – The Polynomial Hierarchy, and Time-Space Tradeoffs

In lecture, we saw that the proof for the time-space tradeoff consisted of two steps: first, simulate Σ_2 in NTIME, and then simulate TISP in Σ_2. This leads to a contradiction with a hierarchy theorem. In this problem we'll see how we can push the second technique a bit more. For the definitions of the complexity classes TISP, $\Sigma_{\mathsf{k}}\mathsf{TIME}$, $\Pi_{\mathsf{k}}\mathsf{TIME}$, etc., consult chapter 5 of Arora-Barak.

(a) Show that, for all $k \geq 1$, $\mathsf{TISP}[t, s] \subseteq \Sigma_{\mathsf{2k}}\mathsf{TIME}[(ts^k)^{1/(k+1)}]$.

(b) (Improved simulation of $\mathsf{TISP}[t, s]$). Show that for all $k \geq 1$, $\mathsf{TISP}[t, s] \subseteq \Pi_{\mathsf{k+1}}\mathsf{TIME}[(ts^k)^{1/(k+1)}]$.

Problem 4 – Circuits and the Polynomial Hierarchy

Show that for every k, there exists a language in Σ_2^{P} that does not have circuits of size n^k. [Note: this does not show that PH does not have polynomial sized circuits! Indeed, showing that $\mathsf{PH} \not\subseteq \mathsf{P/poly}$ (or $\mathsf{PSPACE} \not\subseteq \mathsf{P/poly}$, or even $\mathsf{NEXP} \not\subseteq \mathsf{P/poly}$) seems to be quite beyond the reach of current circuit lower bound techniques.]

Problem 5 – An Implication of P=NP for Circuit Lower Bounds

Here, we will show that upper bounds can sometimes be used to show lower bounds. Suppose that $\mathsf{P} = \mathsf{NP}$[1]. First, show that $\mathsf{P} = \mathsf{NP}$ implies that $\mathsf{EXP} = \mathsf{NEXP}$, where NEXP is the exponential-time version of NP (i.e. the proof size can be $2^{O(n^c)}$ for some constant c, and the proof verifier can also run in exponential time). Then, consider an exponential-time version of the polynomial hierarchy to deduce our lower bound: there exists a language in EXP that requires circuits of size $2^n/n$.

Problem 6 – Bonus Problem

A new tech firm is offering a new revolutionary service: customers can now send any 3-SAT formula to the firm's servers, and in a matter of seconds, they are told whether their formula is satisfiable or not for a flat fee of 100,000$. A finance company has two 3-SAT formulas ϕ_1, ϕ_2 and wants to use the above service to find out about their satisfiability. But of course it would like to do so while spending as little money as possible. This gives raise to the following problem:

Prove that if there exist a polynomial time Turing Machine that given two 3-SAT formulas ϕ_1, ϕ_2 can determine the satisfiability of both formulas using a single query to an NP oracle, then $P = NP$.

[1] Some believe this not to be true.