COMS 4236: Computational Complexity (Fall 2018)
Problem Set #3

Wenbo Gao - wg2313@columbia.edu

December 21, 2018

Problem 1

Problem. Problem 11.5.18 on page 275 of CC: Show that, if NP $\subseteq$ BPP, then RP = NP. (That is, if SAT can be solved by randomized machine, then it can be solved by randomized machines with no false positives, presumably by computing a satisfying truth as in Example 10.3.)

Goal. Prove SAT $\in$ NP-complete $\subseteq$ NP $\subseteq$ RP.

Proof.
Let A be a machine in BPP that decides SAT, $L(A) = SAT$,

Given.

$$\forall x \in SAT \Rightarrow [Pr]_{y \in \{0,1\}^{q(n)}}[D(x,y) = 1] \geq \frac{2}{3}$$

$$\forall x \notin SAT \Rightarrow [Pr]_{y \in \{0,1\}^{q(n)}}[D(x,y) = 1] \leq \frac{1}{3}$$

$\forall w \in SAT$, where w is binary encoding of a boolean formula Φ,
where $w =< \lambda x_1, x_2, \ldots, x_n.\Phi(x_1, x_2, \ldots, x_n) >$,
construct a DTM, D, as the following,

1. Initialize counter $i = 1$
2. Simulate both $\lambda x_{i+1}, \ldots, x_n.\Phi(x_1, \ldots, T, x_{i+1}, \ldots, x_n)$ and $\lambda x_{i+1}, \ldots, x_n.\Phi(x_1, \ldots, F, x_{i+1}, \ldots, x_n)$ on A
3. If $(A(\Phi(x_1, \ldots, T, x_{i+1}, \ldots, x_n)), A(\Phi(x_1, \ldots, F, x_{i+1}, \ldots, x_n))$
   - $= (0, 0)$, then reject
   - $= (1, 0)$, then keep $x_i = T$, increment counter $i = i + 1$, and repeat step 2
   - $= (0, 1)$, then keep $x_i = F$, increment counter $i = i + 1$, and repeat step 2
   - $= (1, 1)$, then keep $x_i = T$ (or $x_i = F$, the choice doesn't matter), increment counter $i = i + 1$, and repeat step 2.

4. Once we have an assignment of Φ, verify the assignment
    - If $\Phi(x_1, x_2, \ldots, x_n) = 1$, then accept;
    - otherwise, reject.

This way we guarantee that if $w \notin SAT$, our machine D always rejects.
If $w \in SAT$, there is $\geq (\frac{2}{3})^n$ chance that the machine A is correct in all n iterations.
We can amplify this probability to be $\geq \frac{1}{2}$ by repeating this experiment polynomial times.
Therefore, $L(D) = \text{SAT} \wedge L(D) \in \text{RP}$, and thus SAT $\in$ RP. $\square$

Problem 2

Problem. Let $0 < \epsilon_1 < \epsilon_2 < 1$ denote two constants. Let $D(\cdot,\cdot)$ be a deterministic polynomial-time computable Boolean function, and let L be a language (the setting so far is exactly the same as the definition of BPP.) D and L satisfy the following property: Given any $x \in \{0,1\}^n$, if y is sampled uniformly at random from $\{0,1\}^m$ for some m polynomial in n, then

$$x \in L \Rightarrow \Pr_{y\in\{0,1\}^m}[D(x,y)=1] \geq \epsilon_2 \text{ and } x \notin L \Rightarrow \Pr_{y\in\{0,1\}^m}[D(x,y)=1] \leq \epsilon_1.$$

Show that $L \in$ BPP. (Note that ϵ_2 can be smaller than 1/2. Use the Chernoff bound.)

Problem 3

Problem. Similar to P/poly, one can define P/$\log n$, where the advice string is of length only $O(\log n)$ for input size n. Show that, if SAT $\in$ P/$\log n$, then P = NP. (Hint: Self-reducibility.)

Problem 4

Problem. Show that, if PSPACE $\subseteq$ P/poly, then PSPACE $= \Sigma_2^P$. (Hint: Use self-reducibility to "implicitly" build a winning strategy for the existential player in the TQBF game.)