15-855: An Introduction to Computational Complexity Theory — Fall 2015

**Problem Set #2** — V. Guruswami & M. Wootters

Due on **October 8, 2015** (in class, or by email to yuzhao1@cs.cmu.edu).

---

**Instructions:** Same as for problem set 1.

---

**Pick any 6 problems to solve out of the 8 problems.** *If you turn in solutions to more than 6 problems, we will take your top 6 scoring problems.*

1. **"Karp-Lipton variant."** If $\mathsf{EXP} \subseteq \mathsf{P}/\text{poly}$, then $\mathsf{EXP} = \Sigma_2$.

   Hint: For any $L \in \mathsf{EXP}$ with an exponential time TM $M$ deciding $L$, using the hypothesis, guess a circuit that computes any cell of the computation tableau of $M$ on input $x$, and use it to decide whether $x \in L$ in $\Sigma_2$.

2. **"Circuit lower bounds."**

   (a) Define $\mathsf{EXPSPACE} = \bigcup_{c \geq 1} \mathsf{SPACE}(2^{n^c})$. Prove that $\mathsf{EXPSPACE} \not\subseteq \mathsf{P}/\text{poly}$, i.e., exponential space does not admit polynomial-sized circuits.

   (b) Prove that for every fixed integer $k \geq 1$, $\mathsf{PH} \not\subseteq \mathsf{SIZE}(n^k)$.

   (c) Strengthen the above result to $\Sigma_2^P \cap \Pi_2^P \not\subseteq \mathsf{SIZE}(n^k)$ for any $k \geq 1$. (Hint: Karp-Lipton Theorem.)

3. **"PH big $\Rightarrow$ PH small."**: Show that $\mathsf{PH} = \mathsf{PSPACE} \Rightarrow \mathsf{PH} = \Sigma_k$ for some finite $k \in \mathbb{N}$.

4. **"Easy decision, hard counting."** A Boolean formula is "monotone" if it uses only ANDs and ORs: no negations. Show that #MONOTONE-SAT (counting the number of satisfying assignments to a given monotone formula) is $\#\mathsf{P}$-complete.

5. **"Toda's theorem for NP via linear codes"** Let $K = 2^n$ and $N = 2^{n^2}$, and let us say that a $K \times N$ $G^{(n)}$ with 0-1 entries is nice if

   - Given $i, j$, the entry $G^{(n)}_{i,j}$ can be computed in $n^{O(1)}$ time.
   - For every $i$, $1 \leq i \leq K$, the $i$'th row of $G^{(n)}$ has at least $N/8$ 1's. Moreover, so does every $N$-vector obtained by XOR-ing any subset $S$ of the rows of $G^{(n)}$.

   Take for granted the existence of a family $G^{(n)}$ of such nice matrices for all large enough $n$. Use this to show that $\mathsf{NP} \subseteq \mathsf{RP}^{\oplus\mathsf{P}[1]}$ where the [1] indicates that the $\mathsf{RP}$ machine makes only one query to the $\oplus\mathsf{P}$ oracle.

6. **"Deciding by majority":** Let us say a language $A$ is in the class $\mathsf{PP}$ if there exists a polynomial time Turing Machine $M$ and a positive integer $c$ such that

$$x \in A \quad \Leftrightarrow \quad \mathbf{Pr}_y[M(x,y) \text{ accepts}] > 1/2 \tag{1}$$

   where the probability is over a random choice of $y$ from $\{0,1\}^{c|x|^c}$. In other words, $x$ is in $L$ iff *more than half* the witnesses $y$ cause $M$ to accept.

(a) Argue that THRESHOLD SAT $= \{\langle \varphi, K\rangle \mid \varphi$ is a Boolean formula on $n$ variables with at least $K$ satisfying assignments$\}$ is $\mathsf{PP}$-complete.

(b) Show that $\mathsf{P}^{\mathsf{PP}} = \mathsf{P}^{\#\mathsf{P}}$.

(c) Show that $\mathsf{PP}$ is closed under complement and symmetric difference.

7. **"Counting with a margin for error"**

(a) Suppose there is a polynomial time algorithm $A_2$ to approximate the number of satisfying assignments to a CNF formula within a factor of 2, i.e., on input $\varphi$ the algorithm outputs a number $A_2(\varphi)$ such that $\#\varphi/2 \leq A_2(\varphi) \leq 2\#\varphi$ where $\#\varphi$ is the number of satisfying assignments to $\varphi$. Prove that for every constant $\epsilon > 0$, there is a polynomial time algorithm $A_{1+\epsilon}$ that approximates the number of satisfying assignments of a CNF formula within a factor $(1+\epsilon)$, i.e., on input $\varphi$, outputs a number $A_{1+\epsilon}(\varphi)$ such that $\frac{\#\varphi}{1+\epsilon} \leq A_{1+\epsilon}(\varphi) \leq (1+\epsilon)\#\varphi$.

(b) Prove that the following problem can be solved in $\mathsf{BPP}^{\mathsf{NP}}$: Given as input a CNF formula $\varphi$ on $n$ variables and an integer $k$, output Yes with probaility at least $1-\frac{1}{n^2}$ if $\#\varphi \geq 2^{k+1}$, and No with probability at least $1-1/n^2$ if $\#\varphi < 2^k$. (There is no requirement on the algorithm if $2^k \leq \#\varphi < 2^{k+1}$.)

<u>Hint</u>: Use pairwise independent hashing.

(c) Using the above, prove that one can approximate the number of satisfying assignments of a CNF formula within a factor of 2 in $\mathsf{BPP}^{\mathsf{NP}}$.

8. **"On $\mathsf{P}^{\mathsf{NP}}$."** Define the language

$$L = \{n\text{-variable formulas } \varphi : \ \varphi\text{'s lexicographically last satisfying assignment has } x_n = 1\}.$$

(Let's also say that $\varphi$'s with no satisfying assignments are not in $L$.)

Show that $L$ is complete for the class $\mathsf{P}^{\mathsf{NP}}$. (Hint: think of an $\mathsf{NP}$-machine trying to simulate an $\mathsf{P}^{\mathsf{NP}}$ machine; it can convince itself of *some* of the oracle answers...)