CS 535: Complexity Theory, Fall 2023

Homework 8

Due: 11:59PM, Tuesday, November 15, 2023.

Reminder. Homework must be typeset with LaTeX preferred. Make sure you understand the course collaboration and honesty policy before beginning this assignment. Collaboration is permitted, but you must write the solutions *by yourself without assistance*. You must also identify your collaborators. Assignments missing a collaboration statement will not be accepted. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

Problem 0 (Term Paper)**.** Reminder that a draft of your term paper is due on Tuesday, Nov. 21. Of course it's "just" a draft, but the more fleshed out it is, the more useful feedback we (and your classmates) will be able to give you.

Problem 1 (Consequences of Valiant-Vazirani)**.** Recall the Valiant-Vazirani Theorem, which says that there is a probabilistic polynomial-time algorithm A that takes as input an n-variable Boolean formula ϕ and outputs another n-variable formula such that

$$\begin{aligned} \varphi \in \mathsf{SAT} &\implies \Pr[A(\varphi) \in \mathsf{USAT}_Y] \geq \frac{1}{8n} \\ \varphi \notin \mathsf{SAT} &\implies \Pr[A(\varphi) \in \mathsf{USAT}_N] = 1. \end{aligned}$$

(a) Adapt Definition 7 (**PromiseBPP**) from Lecture Notes 17 to the class **RP** to define an analogous class **PromiseRP**. (1 point)

(b) Use the VV Theorem to show that $\mathbf{NP} = \mathbf{RP}$ if and only if $\mathsf{USAT} \in$ **PromiseRP**. (3 points)

(c) (*Individual Review: No collaboration allowed for this part only*) A *non-adaptive* oracle Turing machine $M^{\mathcal{O}}$ may write a sequence of strings $q_1, \ldots, q_k$ to its oracle tape and in at most one point in its computation obtain the sequence of answers $\mathcal{O}(q_1), \ldots, \mathcal{O}(q_k)$ (where $\mathcal{O}(q_i)$ stands for the answer to the question "Is $q_i \in \mathcal{O}$?"). That is, all the queries to the oracle have to be issued in a single batch, so the selection of a query cannot depend on the answers to previous queries.

Describe a deterministic poly-time non-adaptive oracle TM M^{SAT} that, given a Boolean formula $\varphi \in \mathsf{USAT}_Y$, outputs a satisfying assignment to φ after making $O(n)$ queries to its SAT oracle. (2 points)

(No need to explain correctness or runtime for this problem, as long as they're reasonably clear from the description of your algorithm.)

(d) Use part (c) and the VV Theorem to describe a *randomized* poly-time non-adaptive oracle TM R^{SAT} that, given a Boolean formula $\varphi \in \mathsf{SAT}$, outputs a satisfying assignment to φ with probability at least $\Omega(1/n)$ after making $O(n)$ queries. (2 points)

(No need to explain correctness or runtime for this problem, as long as they're reasonably clear from the description of your algorithm.)

Problem 2 (Approximate Counting with an **NP** Oracle)**.** As we'll see on Thursday, an important technical tool in the proof of the VV Theorem is *pairwise independent hash functions.* In the following problem, you may assume that for every $k \le n$, there is a family of functions $\mathcal{H}_{n,k}$ such that for every pair of inputs $x, x' \in \{0,1\}^n$ and every pair of possible outputs $y, y' \in \{0,1\}^k$, we have

$$\Pr_{h \leftarrow \mathcal{H}_{n,k}}[h(x) = y \wedge h(x') = y'] = \Pr_{h \leftarrow \mathcal{H}_{n,k}}[h(x) = y] \cdot \Pr_{h \leftarrow \mathcal{H}_{n,k}}[h(x') = y'] = 2^{-k} \cdot 2^{-k} = 2^{-2k}.$$

Moreover, each function h can be described by a poly(n)-size circuit (and hence, be evaluated efficiently), and can be sampled from $\mathcal{H}_{n,k}$ in poly(n) time.

(a) Show that for every subset $S \subseteq \{0,1\}^n$, and every parameter $t > 0$, we have

$$\Pr_{h \leftarrow \mathcal{H}_{n,k}}\left[\left||\{x \in S \mid h(x) = 0^k\}| - 2^{-k} \cdot |S|\right| \ge t\right] \le \frac{1}{t^2} \cdot 2^{-k} \cdot |S|.$$

Hint: The expression $|\{x \in S \mid h(x) = 0^k\}|$ counts the number of elements in S that hash to the all-0 string. The all-0 string is one out of 2^k possible hash images, so the expression $2^{-k} \cdot |S|$ is the expected value of this quantity. So this statement just says that the number of elements that hash to 0 concentrates around its expectation. Use Chebyshev's inequality to control this. Another helpful fact here is that that if $X_1, \dots, X_m$ are pairwise independent random variables, i.e., $\Pr[X_i = a \wedge X_j = b] = \Pr[X_i = a] \cdot \Pr[X_j = b]$ for all i, j, a, b, then $\mathrm{V}ar[\sum_{i=1}^m X_i] = \sum_{i=1}^m \mathrm{V}ar[X_i]$. (6 points)

(b) For a Boolean circuit $C : \{0,1\}^n \to \{0,1\}$, define $F(C) = |\{x \in \{0,1\}^n \mid C(x) = 1\}|$ to count the number of inputs that cause C to evaluate to 1. Use part (a) to show that the following problem ApxC (short for approximate counting) is in $\mathbf{PromiseBPP}^{\mathbf{NP}}$:

$$\mathsf{ApxC}_Y = \{\langle C, k\rangle \mid F(C) \ge 8 \cdot 2^k\}$$
$$\mathsf{ApxC}_N = \left\{\langle C, k\rangle \mid F(C) \le \frac{1}{8} \cdot 2^k\right\}.$$

Hint: Use the fact that the satisfiability problem for Boolean circuits is in **NP**. (6 points)

(c) (Bonus Problem) Use part (b) to show that there exists a constant $\alpha > 0$ and a poly-time randomized **NP**-oracle algorithm that, given as input a Boolean circuit C, estimates $F(C)$ to within a multiplicative factor of α with probability at least 3/4. That is,

$$\Pr\left[\frac{1}{\alpha} \cdot F(C) \leq M^A(C) \leq \alpha \cdot F(C)\right] \geq \frac{3}{4}.$$

(d) (Bonus Problem) Show that there is a poly-time randomized **NP**-oracle algorithm that, given as input a Boolean circuit C, outputs $\perp$ with probability at most 1/2 and, conditioned on not outputting $\perp$, outputs a uniformly random x for which $C(x) = 1$.