6.541/18.405 Problem Set 3

due on **Thursday, May 2, 11:59pm**

Rules: You may discuss homework problems with other students and you may work in groups, but we require that you *try to solve the problems by yourself before discussing them with others*. Think about all the problems on your own and do your best to solve them, before starting a collaboration. If you work in a group, include the names of the other people in the group in your written solution. **Write up your *own* solution to every problem**; don't copy answers from another student or any other source. Cite **all** references that you use in a solution (books, papers, people, websites, etc) at the end of each solution.

We encourage you to use LaTeX, to compose your solutions. The source of this file is also available on Piazza, to get you started!

How to submit: Use Gradescope entry code **2P3PEN**.

Please use a separate page for each problem.

Problem 1: Circuit Lower Bounds from Pseudorandom Generators (3 Points)

Question

We showed that circuit lower bound implied "good" PRGs. Here is a sort-of converse:

Show that if there is a $O(\log(n))$-seed 1/10-pseudorandom generator computable in $2^{O(m)}$ time on m-bit seeds, then there is an $\varepsilon > 0$ such that $\mathbf{DTIME}[2^{O(n)}] \not\subset \mathbf{SIZE}[2^{\varepsilon n}]$.

Answer

TODO: clean this up! Currently messy enough there is a small chance my approach doesn't work.

Say there exists such a pseudorandom generator $g : \{0,1\}* \to \{0,1\}*$ computed by Turing machine G running in 2^{km+l} time. WLOG, say that for all n, $|x| = c\log(n) \implies |g(x)| = n$. For any circuit C of size n,

$$|Pr_{x\in\{0,1\}^{c\log(n)}}[C(g(x)) = 1] - Pr_{x\in\{0,1\}^n}[C(x) = 1]| < 1/10$$

The idea is going to be that in time $2^m \times 2^{km} = 2^{(k+1)m}$, a Turing machine can run g on every length m string, and compute $Pr_{x\in\{0,1\}^{c\log(n)}}[C(g(x)) = 1]$ exactly, but no circuit of size 2^m can do this.

Consider the decision problem $f : \{0,1\}^* \to \{0,1\}$ where

$$f(x) = 1 \iff \exists y \in \{0,1\}^{|x|-1} \text{ s.t. } g(y) = x \qquad (*)$$

Suppose for contradiction that $f \in \mathbf{SIZE}[2^n]$. Let C be the circuit of size 2^n that decides this for a given n. Say $m = cn$ (so $m = c\log(2^n)$). Observe

$$Pr_{y \in \{0,1\}^m}[C(g(y)_{1:m+1}) = 1] = 1$$

Here, $g(y)_{1:m+1}$ is the first $m+1$ bits of $g(y)$.

However,

$$Pr_{x \in \{0,1\}^{2^n}}[C(x_{1:m+1}) = 1] \leq 1/2$$

because only half of the $m+1$ bit strings can be in the range of a function with domain $\{0,1\}^m$. This contradicts $(*)$. Thus $f \notin \mathbf{SIZE}[2^n]$. But certainly $f \in \mathbf{DTIME}[2^{O(n)}]$, because it is possible to loop over all 2^n y values of length n, and for each one check if $x = g(y)$ in time $2^{O(n)}$. This yields an algorithm for deciding f with total runtime $2^n 2^{O(n)} = 2^{O(n)+n} = 2^{O(n)}$.

Problem 2: Randomized Approximate Counting with an NP Oracle (12 pts, 3 for each sub-problem)

Question

We will develop a real-life application of SAT solvers. **Assume $\mathbf{P} = \mathbf{NP}$ in this question.** Let $H_{n,k}$ be a pairwise independent hash family of functions from $\{0,1\}^n$ to $\{0,1\}^k$.

(a) Prove that there is a constant $p \in (0,1)$ and a constant $\varepsilon > 0$ such that for every k and $S \subseteq \{0,1\}^n$,

- if $|S| \leq 2^{k-2}$, then

$$\Pr_{h \in H_{n,k}}[\text{there is an } x \in S \text{ such that } h(x) = 0^k] < p - \varepsilon,$$

  and

- if $|S| \geq 2^{k-1}$, then

$$\Pr_{h \in H_{n,k}}[\text{there is an } x \in S \text{ such that } h(x) = 0^k] > p + \varepsilon.$$

(b) Use part (a) to show that that there is a randomized polynomial-time algorithm that approximates $\#\mathsf{SAT}$ within a factor of 4. More precisely, there is a randomized polynomial-time algorithm that given any Boolean formula F outputs a number K such that $\#\mathsf{SAT}(F)/2 \leq K \leq 2 \cdot (\#\mathsf{SAT}(F))$.

(c) Show that for any constant $\varepsilon > 0$, there is a randomized polynomial-time algorithm that approximates $\#\mathsf{SAT}$ within a factor of $1+\varepsilon$. (Hint: Try to modify the given formula F in some natural way that changes the number of SAT assignments, then feed the modification to your algorithm from part (b).)

(d) Show that you can derandomize the algorithm. That is, prove that if $\mathbf{P} = \mathbf{NP}$ then for every function $f \in \#\mathbf{P}$, and any constant $\varepsilon > 0$, there is a deterministic polynomial-time algorithm that approximates f within a factor of $1 + \varepsilon$. (Warning: the approximate counting problem is *not* a decision problem, so you cannot just "plug in" $\mathbf{P} = \mathbf{NP}$ here...)

Answer

Answer to (a)

Fix S s.t. $|S| \le 2^{k-2}$.

For $x \in S$, let A_x denote the event $A_x = \{h : h(x) = 0^k\}$. We wish to show $\Pr[\cup_{x\in S} A_x] < p - \varepsilon$.

By the inclusion-exclusion principle, letting x_i denote the ith largest value in S,

$$\Pr[\cup_{x\in S} A_x] \le \sum_{i=1}^{|S|} \Pr[A_{x_i}] - \sum_{i=1}^{|S|}\sum_{j=1}^{i-1} \Pr[A_{x_i} \cap B_{x_j}] + \sum_{i=1}^{|S|}\sum_{j=1}^{i-1}\sum_{k=1}^{j-1} \Pr[A_{x_i} \cap A_{x_j} \cap A_{x_k}]$$

$$= \frac{1}{2^k}|S| - \frac{1}{2^{2k}}\binom{|S|}{2} + \sum_{k=1}^{j-1} \Pr[A_{x_i} \cap A_{x_j} \cap A_{x_k}]$$

$$\le \frac{1}{2^k}|S| - \frac{1}{2^{2k}}\binom{|S|}{2} + \frac{1}{2^{2k}}\binom{|S|}{3}$$

$$\le \frac{1}{4} - \frac{1}{2^{2k}}\binom{|S|}{2} + \frac{1}{2^{2k}}\binom{|S|}{3} = \frac{1}{4} - \frac{|S|^2}{2^{2k+1}} + \frac{|S|}{2^{2k+1}} + \frac{1}{2^{2k}}\binom{|S|}{3}$$

$$\le \frac{1}{4} - \frac{1}{32} + \frac{1}{2^{k+3}} + \frac{1}{2^{2k}}\binom{|S|}{3}$$

$$\le \frac{1}{4} - \frac{1}{32} + \frac{1}{2^{k+3}} + \frac{|S|(|S|-1)(|S|-2)}{6}\frac{1}{2^{2k}}$$

$$\le \frac{1}{4} - \frac{1}{32} + \frac{1}{2^{k+3}} + \frac{|S|^3 - 3|S|^2 + 2|S|}{6}\frac{1}{2^{2k}}$$

$$\le \frac{1}{4} - \frac{1}{32} + \frac{1}{2^{k+3}} +$$

[TODO]

[TODO TODO]

Answer to (b)

Let $S = \{x : F(x) = 1\}$. Say $|S| \in [2^{k-2}, 2^{k-1}]$.

Algorithm: For each $k = 0, \ldots, |x|$, Do N times: Check if F and [h(x) = 0] is satisfiable.

If $|S| \leq 2^{k-2}$, with high probability, it will be satisfied $< N(p - \epsilon)$ times. If $|S| \geq 2^{k-1}$, with high probability, it will be satisfied $> N(p - \epsilon)$ times.

So for very large N, there will be some K where for $k < K$, $< N(p + \frac{\varepsilon}{2})$ are satisfied and for $k > K$, $> N(p + \frac{\varepsilon}{2})$ are satisfied.

So our algorithm can be to return 2^{R-1} (or is it $2^{R-\frac{1}{2}}$) where R is the first value of k where the count was $> N(p + \frac{\varepsilon}{2})$. We just have to show that this succeeds with high probability.

[TODO: fill in the details]

Answer to (c)

Let F be a formula on variables $x_1, \ldots, x_n$. Let F_2 be a formula, identical to F, but on a distinct set of variables $y_1, \ldots, y_n$. And in factor for each $i \in \mathbb{N}$, let F_i be a formula identical to F, but on a unique set of variables.

Let $G_i = G_1 \wedge G_2 \wedge \cdots \wedge G_i$, a formula on ni variables. Observe that if F has s satisfying assignments, G_i has s^i satisfying assignments.

Choose i large enough that $(1+\epsilon)^i > 4$. Let s_i denote the number of satisfying assignments to G_i. Run our algorithm from part (b) on G_i to find a value K such that $K \in [s_i/2, 2s_i]$. Then $s_i \in [K/2, 2K]$. Thus $s^i \in [K/2, 2K]$. Let $k = K/2$, so $s^i \in [k, 4k]$. Then

$$s \in [(k)^{1/i}, 4^{(1/i)}(k^{1/i})] \subseteq [(k)^{1/i}, (1+\epsilon)(k^{1/i})]$$

Answer to (d)

Part (c) showed that if $\mathbf{P} = \mathbf{NP}$, then $\#\mathbf{P} \in \mathbf{BPP}$.

Let $A(F, r)$ be a randomized algorithm with randomness r that, with probability $1 - \delta$, outputs a $1 + \varepsilon/2$ approximation to $\#\mathsf{SAT}(F)$.

Here is a deterministic algorithm to check if $\#\mathsf{SAT}(F) \leq \rho(1 + \varepsilon)$ for any ρ. Using the fact that

Problem 3: Constant Round Arthur-Merlin Collapses (3 points)

Question

Prove that for every fixed positive integer k, $\mathbf{AM}[k] \subseteq \mathbf{AM}[2]$.

Hint: Try error-reduction, to make the probability of error very small.

Answer

WLOG assume k is even. Consider an AM[k] protocol to decide $f(x)$. Say on any iteration, a sequence of messages $r_1, m_1, r_2, m_2, \ldots, r_{k/2}, m_{k/2}$ are sent, where r_i are the random messages from the verifier, and m_i are the messages from the prover. Fix a given prover P. Let A_{acc} denote the event that the r_i result in V accepting, and let A_{rej} denote the event that the r_i result in V rejecting. By the definition of an AM protocol, if $f(x) = 1$ there is a prover so $\Pr[A_{\text{acc}}] > 2/3$ and if $f(x) = 0$ then for every prover, $\Pr[A_{\text{rej}} < 1/3]$.

By the error reduction lemma, by running this protocol in parallel $O(k)$ times (with a minor modification to be described in a moment), we obtain an AM[k] protocol such that if $f(x) = 1$, there is a prover so $\Pr[A_{\text{acc}}] > 1 - 2^{-k}$, and if $f(x) = 0$, for every prover, $\Pr[A_{\text{rej}}] < 2^{-k}$.

[TODO: do we need to prove this lemma?]

Now, consider the following $\mathbf{AM}[2]$ protocol. On the first round, the verifier sends a sequence of random bits $r_1, r_2, \ldots, r_{k/2}$, where each sequence of bits is as long as the longest sequence that r_i could have been in the $\mathbf{AM}[k]$ protocol with exponential error bounds. The prover will send a message which is a concatenation $m_1, m_2, \ldots, m_{k/2}$, and the verifier will accept if $r_1, m_1, r_2, m_2, \ldots, m_{k/2}$ would have been an acceptable transcript in the $\mathbf{AM}[k]$ protocol with exponential error bounds.

Analysis. If $f(x) = 1$, then there is a prover which would have almost certainly been accepted in the $\mathbf{AM}[k]$ version of the protocol, and running it in this $\mathbf{AM}[2]$ will succeed with equally high probability. Now say $f(x) = 0$. Let r denote the full string of randomness sent by the verifier and let m denote the prover's full response. It is sufficient to upper-bound $\Pr_r[\exists m.V(x, m, r) = 1]$. By the union bound,

$$\Pr_r[\exists m.V(x, m, r) = 1] \leq \sum_m \Pr_r[V(x, m, r) = 1] \leq \sum_m 2^{-k} \leq 2^M/2^k$$

where M is an upper bound on the length of m, and k is the value we chose in the error bound for the reduced-error $\mathbf{AM}[k]$ protocol. If we choose to set $k > M + 2$, we get

$$\Pr_r[\exists m.V(x, m, r) = 1] < 1/4$$

which is certainly sufficient.

Problem 4: AM Protocol for Set Lower Bound (6 Points, 2 for each sub-problem)

Question

In this problem, we will develop an **AM** protocol for proving a set lower bound, which is used as a subroutine in the **AM** protocol for graph non-isomorphism. In a set lower bound protocol, the prover needs to prove to the verifier that given a (large) set $S \subseteq \{0,1\}^m$ (where membership in S is efficiently verifiable), S has cardinality at least K, up to a factor of 2. More precisely, given any K,

- if $|S| \geq K$ then the prover can make the verifier accept with high probability;
- if $|S| < K/2$ then the verifier rejects with high probability regardless of what the prover does.

(a) Let $H_{m,k}$ be a pairwise independent hash family of functions from $\{0,1\}^m$ to $\{0,1\}^k$. Use the pairwise independent hash family $H_{m,k}$ to give a 2-round **AM** protocol for the set lower bound problem described above.

(b) Show that there exists an **AM** protocol for set lower bound with perfect completeness.

Hint: Consider the case where the prover uses multiple hash functions $h_1, \ldots, h_n$ *so that* $\bigcup_{i=1}^n h_i(S) = \{0,1\}^k$.

(c) Generalize the idea from part (b) to show that every problem in **MA** has a protocol with perfect completeness. Namely, show that for every language $L \in$ **MA**, there exists a probabilistic polynomial time verifier V such that

- If $x \in L$, then there exists m such that $\Pr_r[V(x,r,m) = 1] = 1$.
- If $x \notin L$, then for all m, $\Pr_r[V(x,r,m)] \leq 1/3$.

Answer to (a)

The protocol. The verifier will send l random pairwise independent hash functions $h_1, \ldots, h_l$. (Each one can be sent in $O(mk)$ bits.)

The prover will send back K, and then, for each y_i, either a message saying there is no $x \in S$ s.t. $h_i(x) = 0^k$, or a string x_i. The verifier will reject if any x_i does not satisfy $h_i(x_i) = 0^k$ or if any $x_i \notin S$. If the prover sends back $\geq \frac{3}{4}\frac{K}{2^k}l$ valid x_i, the verifier will accept. The verifier will reject otherwise.

Analysis. Say $|S| \geq K$. Then for any h_i,

$$\Pr[\exists x \in S.h_i(x) = 0^k] \geq \sum_{x \in S} \frac{1}{2^k} - \sum_{x \neq y \in S} \frac{1}{2^{2k}}$$
$$> \frac{K}{2^k} - \frac{K^2}{2^{2k+1}} = \frac{K}{2^k} - \frac{1}{2}(\frac{K}{2^k})^2$$

Problem 5: The Limits of PCPs (4 Points, 2 for each sub-problem)

Recall that in class we defined $\mathbf{PCP}_s[r(n), q(n)]$ to be the set of functions with probabilistically checkable proofs having "soundness" s. In general, we can parametrize the "completeness" as well.

Specifically, define $f : \{0,1\}^* \to \{0,1\}$ to be in $\mathbf{PCP}_{c,s}[r(n), q(n)]$ if there is a probabilistic polynomial time algorithm V such that for all x, V uses $O(r(|x|))$ random bits, asks $q(|x|)$ oracle queries to a proof string P non-adaptively, must decide whether accept or reject, and

- $f(x) = 1 \Longrightarrow$ there is a P such that $\Pr[V^P(x) \text{ accepts}] \geq c$.
- $f(x) = 0 \Longrightarrow$ for all P, $\Pr[V^P(x) \text{ accepts}] < s$.

Note that in this generalized version, when $f(x) = 1$, we do not require the verifier to accept with probability 1 on some proof P.

In the PCP lectures, it was proved that $\mathbf{PCP}_{1,1}(\log n, 3) = \mathbf{NP}$. The number 3 here is actually the smallest possible. In this problem, you are asked to show that if we reduce the number of queries to two or one, the classes become $\mathbf{P}$. Prove that:

(a) for every $0 < s \leq c \leq 1$, $\mathbf{PCP}_{c,s}(\log n, 1) = \mathbf{P}$.

(b) for every $0 < s \leq 1$, $\mathbf{PCP}_{1,s}(\log n, 2) = \mathbf{P}$.

Hint: Think about these 1-query and 2-query PCPs from the CSP/inapproximability perspective: what you want to show is that the resulting CSPs are in fact easy to solve.

Extra credit: Prove that for every $0 < s \leq 1$, $\bigcup_{k \geq 1} \mathbf{PCP}_{1,s}(n^k, 2) \subseteq$ **PSPACE**

Hint: Use the fact that 2SAT is in NL.