Problem Set 5

- Due Date: **16th June 2021**
- Turn in your problem sets electronically (pdf or text file) on Acadly.
- Collaboration is encouraged, but all writeups must be done individually and must include names of all collaborators.
- Referring to sources other than the text book and class notes is strongly discouraged. But if you do use an external source (eg., other text books, lecture notes, or any material available online), ACKNOWLEDGE all your sources (including collaborators) in your writeup. This will not affect your grades. However, not acknowledging will be treated as a serious case of academic dishonesty.
- The points for each problem are indicated on the side. There are **5 questions** with a total of **75 points** in this problem set (there are totally 95 points, but may choose to skip either Question 1 or Question 2)
- Some of the questions are broken-down in to multiple subdivisions to guide you towards a solution. Feel free to use the earlier subdivisions for later ones, even if you haven't solved it (yet).
- Be clear in your writing.

You have a choice to answer just one of Question 1 and Question 2.

Question 1 (Perfect completeness in MA and AM). (2 + 4 + 4 + 2 + 4 + 4)

Note: You can either answer this question or Question 2.

In this problem, you will be showing that you may assume, without loss of generality, that AM and MA protocols have perfect completeness.

Recall the definition of MA that says that a language $L \in \mathsf{MA}$ if there is a language R in P with y, z having lengths polynomial in $|x|$ such that

$$x \in L \Rightarrow \exists y \;:\; \Pr_z[(x,y,z) \in R] \geq 2/3,$$
$$x \notin L \Rightarrow \forall y \;:\; \Pr_z[(x,y,z) \in R] \leq 1/3.$$

(a) Show that the definition below is equivalent to the above definition of MA — there is a polynomial time language R, with y, z having lengths polynomial in $|x|$ such that

$$x \in L \Rightarrow \exists y \;:\; \Pr_{z \in \{0,1\}^r}[(x,y,z) \in R] \geq 1 - \frac{1}{4r},$$
$$x \notin L \Rightarrow \forall y \;:\; \Pr_{z \in \{0,1\}^r}[(x,y,z) \in R] \leq \frac{1}{4r}.$$

That is, the error can be made to be roughly $1/|z|$.

(b) Suppose S is an arbitrary subset of $\{0,1\}^r$. Prove the following two facts:

- If $|S| \le \frac{2^r}{4r}$, then for any choice of vectors $a_1, \ldots, a_r \in \{0,1\}^r$ we have
$$\left| \bigcup_{i=1}^{r} (S \oplus a_i) \right| \le \frac{2^r}{4}.$$
- If $|S| \ge 2^r \cdot \left(1 - \frac{1}{4r}\right)$, then there exists choice of vectors $a_1, \ldots, a_r \in \{0,1\}^r$ such that
$$\bigcup_{i=1}^{r} (S \oplus a_i) = \{0,1\}^r.$$

(c) Using the above two parts, show that the following definition is equivalent to the definition of MA — there is a polynomial time language R, with y, z having lengths polynomial in $|x|$ such that
$$x \in L \Rightarrow \exists y \ : \ \Pr_z[(x,y,z) \in R] = 1,$$
$$x \notin L \Rightarrow \forall y \ : \ \Pr_z[(x,y,z) \in R] \le \frac{1}{3}.$$

(d) Recall the definition of AM — a language $L \in \mathsf{AM}$ if there is a polynomial time language R such that
$$x \in L \Rightarrow \Pr_y[\exists z \ : \ (x,y,z) \in R] \ge \frac{2}{3},$$
$$x \notin L \Rightarrow \Pr_y[\exists z \ : \ (x,y,z) \in R] \le \frac{1}{3}.$$

Show that the following is an equivalent definition, without loss of generality:
$$x \in L \Rightarrow \Pr_{y \in \{0,1\}^r}[\exists z \ : \ (x,y,z) \in R'] \ge 1 - \frac{1}{4r},$$
$$x \notin L \Rightarrow \Pr_{y \in \{0,1\}^r}[\exists z \ : \ (x,y,z) \in R'] \le \frac{1}{4r}.$$

(e) Consider the following MAM protocol for $L \in \mathsf{AM}$, similar to the earlier case.

> Merlin sends $a_1, \ldots, a_\ell \in \{0,1\}^r$ to Arthur. Arthur picks a random $y \in \{0,1\}^r$ and sends that to Merlin.
> Merlin now sends an $i \in [\ell]$ and a z to Arthur.
> Arthur accepts if $(x, y \oplus a_i, z) \in R'$.

Show that this is a legitimate MAM protocol with perfect completeness for an arbitrary AM protocol.

(f) Show that an MAM protocol with perfect completeness can be simulated by an AM protocol with perfect completeness. (This can be done by imitating the proof done in lecture that $\mathsf{MA} \subseteq \mathsf{AM}$ and observing it preserves completeness).

Question 2 (Round reduction for AM)**.** (3 + 9 + 5 + 3)

Note: You can either answer this question or Question 1.

We mentioned in our lectures that $\mathsf{AM}[k] = \mathsf{AM}$ for constant k; we will prove this in this problem using some *quantifier jugglery*. For the purpose of this problem, we will use a definition different from that in lecture, the k in $\mathsf{AM}[k]$ and $\mathsf{MA}[k]$ will *not* refer to the number of rounds, but to the number of players (w/ alternation). In other words, $\mathsf{AM}[3] = \mathsf{AMA}$ and *not* AMAMAM.

Let $\mathsf{prAM}[k]$ be the promise problem version of $\mathsf{AM}[k]$ (i.e, it has the same completeness and soundness properties for the YES and NO instances as $\mathsf{AM}[k]$, but the YES and NO instances do not partition the universe (there could be "don't care" instances)).

For a class C of promise problems, we define $\mathsf{pr\Sigma} \cdot \mathsf{C}$ to be the class of promise problems Π such that there exists a promise problem $\Pi' \in \mathsf{C}$ and a polynomial p for which

$$\begin{aligned} x \in \Pi_Y &\quad\Rightarrow\quad \exists y \in \{0,1\}^{p(n)} \;:\; (x,y) \in \Pi'_Y \\ x \in \Pi_N &\quad\Rightarrow\quad \forall y \in \{0,1\}^{p(n)} \;:\; (x,y) \in \Pi'_N \end{aligned}$$

Similarly, we define $\mathsf{prBP}\cdot\mathsf{C}$ to be the class of promise problems Π such that there exists a promise problem $\Pi' \in \mathsf{C}$ and a polynomial p for which

$$\begin{aligned} x \in \Pi_Y &\quad\Rightarrow\quad \Pr_{y \in \{0,1\}^{p(n)}}[(x,y) \in \Pi'_Y] \geq 2/3 \\ x \in \Pi_N &\quad\Rightarrow\quad \Pr_{y \in \{0,1\}^{p(n)}}[(x,y) \in \Pi'_N] \geq 2/3 \end{aligned}$$

(a) Show that for every integer $k \geq 1$, we have $\mathsf{prMA}[k] = \mathsf{pr\Sigma} \cdot \mathsf{prAM}[k-1]$ and $\mathsf{prAM}[k] = \mathsf{prBP} \cdot \mathsf{prMA}[k-1]$, (where $\mathsf{prMA}[0] = \mathsf{prAM}[0] = \mathsf{prP}$ by definition).

(b) Similar to the proof you saw in class for $\mathsf{MA} \subseteq \mathsf{AM}$, show that we have $\mathsf{pr\Sigma} \cdot \mathsf{prBP} \cdot \mathsf{C} \subseteq \mathsf{prBP} \cdot \mathsf{pr\Sigma} \cdot \mathsf{C}$ for any class C of promise problems.

[Hint: You may want to change that 2/3 to something exponentially close to 1 first (with appropriate justification of course!).]

(c) Prove that for every constant $k \geq 2$, we have $\mathsf{prAM}[k] = \mathsf{prAM}$. Conclude that $\mathsf{AM}[k] = \mathsf{AM}$.

(d) Where in the above parts was it important to work with promise problems instead of languages?

Question 3 (Two ways in which the PCP theorem is optimal). (5+10)

We mentioned in class that there are NP-complete languages L (such as CircuitSat) that we now know to be in $\mathrm{PCP}_{1,0.51}(O(\log n), 3)$ (that is, there is a 3-query PCP with perfect completeness where the verifier only tosses $O(\log n)$ random coins).

(a) Show that, for any constant $s < 1$, we have $\mathrm{PCP}_{1,s}(O(\log n), 2) \subseteq P$. (Hence, if an NP-complete language has a 2-query PCP with perfect completeness, then $\mathsf{P} = \mathsf{NP}$).

(b) Let $s < 1$ be any constant. If L is an NP-complete language and $L \in \mathrm{PCP}_{1,s}(o(\log n), O(1))$, then $\mathsf{P} = \mathsf{NP}$. (That is, if there is a PCP for an NP-complete language L that uses constant number of queries and *sub-logarithmic* randomness, then $\mathsf{P} = \mathsf{NP}$).

[Hint: Try and use the PCP to build a *length-reducing* reduction from L to L.]

Question 4 (Proof of Knowledge, with one of two witnesses). (4 + 4 + 6 + 6)

In class, we saw a *perfect zero-knowledge proof of knowledge* for Graph Isomorphism. Consider the following language

$$\mathrm{GI}_{\mathrm{OR}} = \{((G_0, G_1), (H_0, H_1)) \;:\; G_0 \equiv G_1 \text{ or } H_0 \equiv H_1\}.$$

Clearly the above language is in NP, where an accepting witness is an isomorphism between one of the pairs.

Suppose a Prover is given an isomorphism for one of the pairs, we want to build a PZK-PoK for the above language (in particular, the Verifier shouldn't even learn *which* of the two pairs the prover has the witness for, or even which of the two graphs are isomorphic). Below is a description of the protocol between the honest prover and the honest verifier.

> Prover sends two graphs G, H to the verifier (that are supposed to be a shuffling of one of the G_i's and one of the H_j's).
>
> The verifier sends a bit $s \in \{0,1\}$.
>
> The prover now sends (σ_1, b_1) and (σ_2, b_2).
>
> The verifier accepts if $\sigma_1(G_{b_1}) = G$ and $\sigma_2(H_{b_2}) = H$ and $b_1 \oplus b_2 = s$.

(a) Show that the above protocol for $\mathrm{GI}_{\mathrm{OR}}$ has completeness 1 and soundness error at most 1/2.

(b) Show that the above protocol is a perfect zero-knowledge protocol by constructing an appropriate simulator (for possibly cheating verifiers as well).

(c) Show that even a randomized polynomial time machine that has been provided a valid isomorphism for just *one* of the pairs can execute the above protocol faithfully (that is, the "prover" needn't be all-powerful).

(d) Show that the above protocol has knowledge-soundness 1/2. That is, any Prover that manages to convince the honest verifier with probability substantially bigger than 1/2 (say probability at least 3/4) must "know" an isomorphism for one of the two pairs of graphs.

Question 5 (Perfectly binding and hiding commitment schemes?). (4 + 4 + 6 + 6)

In class, we introduced the concept of commitment schemes, which were defined by a function

$$\text{Commit} : \Sigma^n \times \{0,1\} \to \Sigma^m$$

such that

- (computationally hiding) $\{\text{Commit}(k,0)\}_k \approx_{c.i} \{\text{Commit}(k,1)\}_k$, and
- (perfectly binding) $\text{Commit}(k_1,b_1) = \text{Commit}(k_2,b_2) \implies b_1 = b_2$.

(a) Formally define the notion of a "perfectly hiding" commitment scheme, and show that there are *no* commitment schemes that are both perfectly hiding and perfectly binding

When we have multiple provers at our disposal, we can have these commitments as interactive protocols. We will now see a *commitment protocol* that will allow us to convert any multi-prover interactive protocol into a perfect zero-knowledge interactive protocol! Below is the description of the *commitment protocol*.

> **Before protocol begins:** Let $f_0, f_1 : \{0,1\} \to \{0,1\}$ given by $f_0(x) = x$ and $f_1(x) = 1 - x$. Prior to the protocol beginning, the provers P_1 and P_2 are supposed to share random bits $r_1, \ldots, r_k$.
>
> **P_1 commits to bit b:** The commitment protocol, when P_1 wishes to commit to a bit b, begins with the verifier sending m uniformly random bit $s_1, \ldots, s_m \in \{0,1\}$ to P_1 (these are NOT sent to P_2)

P_1 now sends their *commitment* $c_1, \ldots, c_m$ where $c_i = f_{s_i}(r_i) \oplus b$.

Reveal: In order to reveal the committed bit, the Verifier P_1 for the bit b and asks P_2 for the bits $r_1, \ldots, r_m$ and checks if $c_i \oplus f_{s_i}(r_i) = b$.

(b) (Perfectly hiding) Show that the distributions of the commitment $c_1, \ldots, c_m$ for both $b = 0$ and $b = 1$ are identical.

(c) (Almost-perfectly binding) For any commitment $c_1, \ldots, c_m$ for a bit b, show that the probability that the two provers convince the verifier that the committed bit is $(1 - b)$ is at most $\frac{1}{2^m}$.

(That is, there is an at most $\frac{1}{2^m}$ for the provers to cheat on their original bit after the commitment).

(d) Show that the above commitment protocol is a zero-knowledge protocol by describing a suitable simulator.

You should now be able to (vaguely) convince yourself that every multi-prover interactive proof can now be made into a perfect zero-knowledge multi-prover interactive proof!