Homework 5

due Monday March 14, 2022

Directions: Do all problems.

A Public-Coin Protocol for Set-Lower Bound

Recall the "set lower bound" semi-function $f_{\mathsf{SLB}} : \{0,1\}^* \to \{0,1\} \cup \{\perp\}$ which takes as input a set $S \subset \{0,1\}^m$ and an integer $K \in \mathbb{Z}$ and outputs 1 if $|S| \geq K$, outputs 0 if $|S| \leq K/2$ and otherwise outputs $\perp$. We say that a set $S \subset \{0,1\}^m$ is *efficiently recognizable* if there exists a polynomial time TM M_S such that for all $s \in \{0,1\}^m$, $\mathsf{M}_S(s) = 1$ if $s \in S$ and $\mathsf{M}_S(s) = 0$ if $s \notin S$. In other words, S is efficiently recognizable if membership in S can be efficiently computed by a TM. We say that S is *efficiently certifiable* if there is a polynomial time TM M_S such that $s \in S$ if and only if there exists $w_s \in \{0,1\}^*$ such that $\mathsf{M}_S(s, w_s) = 1$; the string w_s is called the *certificate* of s. In other words, S is efficiently certifiable if membership in S can be efficiently computed by a NDTM. The exercises in this section will walk you through the analysis of the Goldwasser-Sipser protocol which is a public-coin protocol for computing set-lower-bound for sets which are efficiently certifiable.

The Connection to the Soda-Pop Protocol for Graph Non-Isomorphism. Recall in class we saw a 2-round interactive proof system for graph non-isomorphism. Given inputs (G_1, G_2), V chose $c \sim \{1,2\}$ and a random graph H which is isomorphic to G_c and sent H to P. Let $\mathcal{H}$ denote the set of possible messages that V could send. Note $\mathcal{H}$ is efficiently certifiable since given H, the certificate that $H \in \mathcal{H}$ would be the pair (π, c) where $\pi \in \mathrm{Perm}(V)$ is such that $H = \pi(G_c)$ (*i.e.*, the certificate is V's private randomness used to generate the message H). The key observation is that when $G_1 \not\simeq G_2$, $|\mathcal{H}| = 2n!$ (where $n = |V|$), whereas when $G_1 \simeq G_2$, $|\mathcal{H}| = n!$. Thus, the graph non-isomorphism problem can be reduced to the set-lower-bound problem via the reduction which maps (G_1, G_2) to $(\mathcal{H}, 2n!)$ (technically speaking, the reduction would not output the set $\mathcal{H}$ since this would be inefficient, instead the reduction outputs the description of the NDTM which computes membership in $\mathcal{H}$).

Problem 1. Suppose that a semi-function $f : \{0,1\}^* \to \{0,1\} \cup \{\perp\}$ can be computed by a two-round public-coin interactive proof system $\langle \mathsf{P}, \mathsf{V} \rangle$ which has completeness 2/3 and soundness 1/3. Let the transcript of this proof system by (a, b) so a is a random string and b is P's response. Consider the following two-round public-coin interactive proof system $\langle \mathsf{P}', \mathsf{V}' \rangle$.

Input: Both P' and V' get $\mathbf{x} \in \{0,1\}^*$ as input;

1. V' sends $a_1, \ldots, a_n$ to P' where each a_i is randomly chosen;

2. P' returns $b_1, \ldots, b_n$ to V';

Output: V' outputs $\mathsf{MAJORITY}_i\{\mathsf{V}(\mathbf{x}, a_i, b_i)\}$.

Prove that $\langle \mathsf{P}', \mathsf{V}' \rangle$ computes f with completeness $1 - 2^{-\Omega(n)}$ and soundness $2^{-\Omega(n)}$.

The amplification procedure of Problem 1 works just as well if you start with a proof system with completeness c and soundness s for any constants s, c as long as $s < c$. In Problems 2 and 3 we will analyze such a proof system for set-lower-bound. The proof system will make use of the following gadget.

Claim (Pairwise Independent Hash Functions). For any integers m, k there exists a family of functions $\mathcal{F} \subset \{\varphi : \{0,1\}^m \to \{0,1\}^k\}$ such that:

1. for all $\mathbf{x} \in \{0,1\}^m$ and $\mathbf{y} \in \{0,1\}^k$, $\Pr_{\varphi \sim \mathcal{F}}[\varphi(\mathbf{x}) = \mathbf{y}] = 2^{-k}$;
2. for all $\mathbf{x} \neq \mathbf{x}' \in \{0,1\}^m$ and $\mathbf{y}, \mathbf{y}' \in \{0,1\}^k$, $\Pr_{\varphi \sim \mathcal{F}}[\varphi(\mathbf{x}) = \mathbf{y} \ \& \ \varphi(\mathbf{x}') = \mathbf{y}'] = 2^{-2k}$.

Hash functions are very useful in computer science at large and they come in many different varieties offering different security guarantees, performance specifications, and so on. Pairwise-independent hash functions are the particular variety we will use here; the second condition above is the important one. It is the reason for the name 'pairwise independent', and it also implies the first condition, we have listed both for convenience. We will construct a family of pairwise independent hash functions later in the problem set. Now, consider the following protocol for set-lower-bound.

Input: Both P and V get (S, K) as input where $S \subset \{0,1\}^m$ is an efficiently certifiable set $K \in \mathbb{N}$. Let $k \in \mathbb{N}$ be such that $2^{k-2} < K \leq 2^{k-1}$, and let $\mathcal{F} \subset \{\varphi : \{0,1\}^m \to \{0,1\}^k\}$ be a family of pairwise independent hash functions.

1. $\mathsf{V} \to \mathsf{P}$: V draws $\varphi \sim \mathcal{F}$ and $\mathbf{y} \sim \{0,1\}^k$ and sends $(\varphi, \mathbf{y})$ to P.

2. $\mathsf{P} \to \mathsf{V}$: Upon receiving $(\varphi, \mathbf{y})$ from V, P returns $(\mathbf{x}, w_{\mathbf{x}})$.

Output: V checks that $\varphi(\mathbf{x}) = \mathbf{y}$ and $\mathbf{x} \in S$ (*i.e.*, V checks that $\mathbf{x} \in S$ using the certificate $w_{\mathbf{x}}$); if both checks pass V outputs 1, otherwise 0.

Problem 2. Start by proving soundness (it's easier). Specifically, prove that if $|S| \leq K/2$, then for any φ, $\Pr_{\mathbf{y} \sim \{0,1\}^k}[\mathbf{y} \in \varphi(S)] \leq \frac{K}{2} \cdot 2^{-k}$. Deduce that if $f(\mathbf{x}) = 0$ then for all P^*, $\Pr[\langle \mathsf{P}^*, \mathsf{V} \rangle (S, K) = 1] \leq \frac{K}{2} \cdot 2^{-k}$.

Problem 3. Prove completeness. Specifically, prove that if $|S| \geq K$, then for all $\mathbf{y} \in \{0,1\}^k$, $\Pr_{\varphi \sim \mathcal{F}}[\mathbf{y} \in \varphi(S)] \geq \frac{3K}{4} \cdot 2^{-k}$ (**Hint:** use the inclusion-exclusion formula). Deduce that when $f(\mathbf{x}) = 1$, $\Pr[\langle \mathsf{P}, \mathsf{V} \rangle (S, K) = 1] \geq \frac{3K}{4} \cdot 2^{-k}$.

A MIP System for 3COL

In class we considered the following MIP system for $\mathsf{3COL}$:

- **Input:** $\mathsf{P}_1, \mathsf{P}_2, \mathsf{V}$ all get a regular graph G as input; P_1 and P_2 additionally get a coloring $\Gamma : V \to \{0,1,2\}$ as input.

1. V flips a coin $c \sim \{0, 1\}$;

- if $c = 0$, V chooses a random edge $e = (v, w) \sim E$ and sends v to P_1 and w to P_2;
- if $c = 1$, V chooses a random $v \sim V$ and sends v to both P_1 and P_2;

2. both provers P_1 and P_2 play the same way: upon receiving a vertex v, they return $\Gamma(v)$;

- **Output:** V prepares its output bit as follows:
  - if $c = 0$, V outputs 1 if $\Gamma(v) \neq \Gamma(w)$, 0 if $\Gamma(v) = \Gamma(w)$;
  - if $c = 1$, V outputs 1 if $\Gamma(v) = \Gamma(v)$ (*i.e.*, if V receives the same color from both provers), and outputs 0 otherwise.

Problem 4. Prove that the above MIP system computes f_{3COL} and has completeness 1 and soundness $1 - \frac{1}{2|E|}$.

Approximation Algorithms

Problem 5. In class we mentioned that a random assignment will satisfy a $7/8-$fraction of the clauses of a $\mathsf{3CNF}$ formula Φ in expectation. Describe a deterministic, polynomial time algorithm which, given a $\mathsf{3CNF}$ formula with m clauses, outputs an assignment which satisfies at least $7m/8$ clauses.

Problem 6. The algorithm below is a simple greedy algorithm which, given a graph $G = (V, E)$, outputs a vertex cover $C \subset V$. Prove that the output of the algorithm satisfies $|C| \leq 2|C_{\mathsf{opt}}|$ where $C_{\mathsf{opt}} \subset V$ is an *optimal* vertex cover (*i.e.*, a vertex cover of minimal size). Thus, the algorithm is a $2-$approximation algorithm of the $\mathsf{NP}-$complete problem vertex cover.

- **Input:** The algorithm takes a graph $G = (V, E)$ as input;

1. Initialize $C = \emptyset$ and $E_{\mathsf{cur}} = E$.

2. While $E_{\mathsf{cur}} \neq \emptyset$, do the following:

- pick any $e = (v, w) \in E_{\mathsf{cur}}$ and set $C = C \sup\{v, w\}$;
- remove from E_{cur} any edge which contains either v or w (or both) as an endpoint.

- **Output:** Output C.