CMPUT 675: Computational Complexity Theory **Winter 2019**

Lecture 12 (Feb 14): Interactive Proofs with Public Coins

Lecturer: Zachary Friggstad *Scribe: Ramin Mousavi*

12.1 Introduction

Recall the definition of $\mathbf{IP}[k]$ from the last lecture, i.e., we say $L \in \mathbf{IP}[k]$ if there is a probabilistic verifier V whose running time is polynomial in the size of the input x alone and

$$\text{(Completeness)} \quad x \in L \Rightarrow \exists P,\ \Pr[k - \textit{round interactions of } V \textit{ and } P \textit{ on } x \textit{ make } V \textit{ accepts}] \geq \tfrac{2}{3},$$

$$\text{(Soundness)} \quad x \notin L \Rightarrow \forall P,\ \Pr[k - \textit{round interactions of } V \textit{ and } P \textit{ on } x \textit{ make } V \textit{ accepts}] \leq \tfrac{1}{3}.$$

The number of rounds k could up to a polynomial in $|x|$. We emphasize that the running time of the verifier in each round is be bounded by a polynomial in $|x|$ itself.

We also defined $\mathbf{IP}$ to be $\bigcup_{c \geq 0} \mathbf{IP}[n^c]$. One can show that $\mathbf{IP} \subseteq \mathbf{PSPACE}$ since for any verifier, we can compute the optimum prover (the one which maximizes the verifier's acceptance probability) using only polynomial space in the size of the input. The assignment asks you to fill out the details. Later we will prove that in fact $\mathbf{IP} = \mathbf{PSPACE}$.

One of the key property of $\mathbf{IP}$ is that the prover does not know the random bits of the verifier. In this lecture, we focus on a more restricted complexity class than $\mathbf{IP}$ known as $\mathbf{AM}$.

Definition 1 (Arthur-Merlin Intractive Proof) $\mathbf{AM}[k]$ *is a subset of* $\mathbf{IP}[k]$ *when we restrict the verifier to only send its random bits as messages to the prover, and not using any other random bits not contained in its messages. We also set* $\mathbf{AM} := \mathbf{AM}[2]$.

The above intractive proof also known as public coin proof simply because the prover is aware of all the random bits used by verifier.

Recall the Graph Non-Isomorphism problem (GNI), i.e., deciding whether two given graphs are *not* isomorphic. In the previous lecture, we showed that GNI $\in \mathbf{IP}$. In this lecture, we show that in fact by using a more sophisticated protocol, we can restrict the verifier to only use the public random bits. For the rest of this lecture, we will focus on proving the following result:

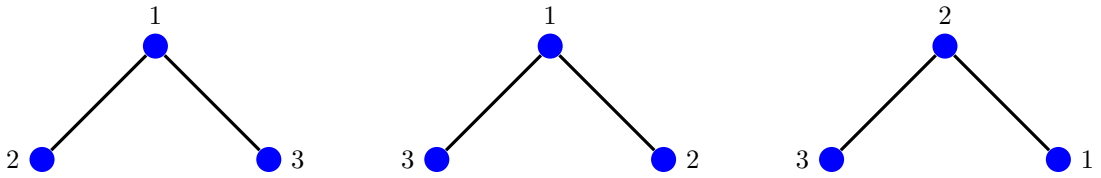
Theorem 1 GNI $\in \mathbf{AM}$.

More generally, we have the following theorem:

Theorem 2 (Goldwasser-Sipser '87) *For every* $k : \mathbb{N} \to \mathbb{N}$ *with* $k(n)$ *computable in* $\text{poly}(n)$,

$$\mathbf{IP}[k] \subseteq \mathbf{AM}[k+2].$$

The main idea of the proof of Theorem 2 is the same as the proof of Theorem 1 and it is omitted here.



(a) Graph G (b) Graph H_1 (c) Graph H_2

Figure 12.1: Graphs in Example 1.

12.2 Proof of Theorem 1 (First Attempt)

In this discussion, for any graph G we consider with, say, n nodes, we use numbers 1 through n to label the nodes of G.

Let $\text{aut}(G) := \{\pi \in S_n : G = \pi(G)\}$, where S_n is the symmetric group consists of all $n!$ permutations of the sequence $[1, 2, 3, \ldots, n]$ and $\text{aut}(G)$ is the set of all permutation π of the labels of G such that $ij \in E(G)$ if and only if $\pi(i)\pi(j) \in E(G)$. That is, permuting the labels of G according to some $\pi \in \text{aut}(G)$ gives us the same labelled graph.

Example 1 *Consider the graphs G, H_1, and H_2 in Figure 12.1. Note that $G = H_1$, i.e., $V(G) = V(H_1)$ and $E(G) = E(H_1)$. Define $\pi \in S_3$ to be $\pi(1) = 1$, $\pi(2) = 3$, and $\pi(3) = 2$. Then, $\pi(G) = H_1 = G$; hence, $\pi \in \text{aut}(G)$. Define $\pi' \in S_3$ to be $\pi(1) = 2$, $\pi(2) = 3$, and $\pi(3) = 1$. So $\pi(G) = H_2$ and H_2 is isomorphic to G but $\pi' \notin \text{aut}(G)$.*

The following lemma reduces GNI problem to a counting problem.

Lemma 1 *Let G_1, G_2 be two graphs on n nodes. Let $S := \{(H, \pi) : (H \cong G_1 \text{ or } H \cong G_2) \text{ and } \pi \in \text{aut}(H)\}$. Then,*

1. *If $G_1 \cong G_2$, then $|S| = n!$, and*
2. *If $G_1 \ncong G_2$, then $|S| = 2 \cdot n!$.*

Proof. Let G be a graph on n nodes. Define set S_G as follows: $S_G := \{(H, \pi) : H \cong G \text{ and } \pi \in \text{aut}(H)\}$. We will show that $|S_G| = n!$ and this proves the lemma since $S = S_{G_1} \cup S_{G_2}$ and in Case 1 we have $S_{G_1} = S_{G_2}$ and in Case 2 we have $S_{G_1} \cap S_{G_2} = \emptyset$.

Suppose $\pi_1, ..., \pi_k$ are all the permutations such that $\pi_i(G) = H$ for all $1 \le i \le k$. Let $\text{aut}(H) = \{\sigma_1, ..., \sigma_t\}$. By considering the permutations $\pi_i \circ \pi_1^{-1} : H \to H$ for all $1 \le i \le k$, we see they are distinct automorphisms of H so $t \ge k$. Next, by considering the permutations $\sigma_i \circ \pi_1 : G \to H$ for all $1 \le i \le t$ we also see $k \ge t$. So we conclude $t = k$. Now it is easy to see there is a bijection between S_n and S_G. ■

By Lemma 1, in order for the prover to convince the verifier that $G_1 \ncong G_2$ with good probability, it is enough to convince the verifier that $|S| = 2 \cdot n!$ with good probability.

Also note that given a pair (H, π), to decide whether it is in S or not can be certified by the prover easily. The prover can send a permutation σ and $i \in \{1, 2\}$ and verifier can check $\sigma(G_i) = H$ or not. Also verifier can check $\pi(H) = H$ holds or not without any help from the prover.

Lemma 1 and the above discussion suggest that to prove Theorem 1, it suffices to show that the *set lower bound* problem defined below, is in **AM**.

Definition 2 (Set Lower Bound Problem) *Given $m \in \mathbb{N}$, an implicit set $S \subseteq \{0,1\}^m$ such that membership in S can be certified in* $\text{poly}(m)$*, and $\gamma \in \mathbb{N}$. The Yes-instances of the problem are when we have $|S| \geq \gamma$ and No-instances are when $|S| \leq \frac{\gamma}{2}$. The problem is to decide between Yes/No-instances.*

Now, suppose in the set lower bound problem, Yes-instances are $|S| = \gamma = \alpha \cdot 2^m$ and No-instances are $|S| = \frac{\gamma}{2} = \frac{\alpha}{2} \cdot 2^m$ for some constant α. Then, the following protocol works correctly:

Protocol For Set Lower Bound Problem When $\frac{|S|}{2^m}$ Is Constant

Reapeat $O(\alpha) = O(1)$ times in parallel:

Verifier: pick $x \sim \{0,1\}^m$ and ask prover is $x \in S$?

Prover: Send a certificate that $x \in S$ (if possible).

Verifier: Check that the certificate is valid.

Verifier accepts if the fraction of valid certificates is at least $\frac{3}{4}\alpha$.

Note that repeating in parallel in the above protocol means verifier picks several choices of x, $O(1)$ many, and send them all to the prover at once. So this protocol is a 2-round interactions between the verifier and the prover. Furthermore, if $|S| = \alpha \cdot 2^m$, then the probability that $x \in S$ is exactly α, otherwise the probability is exactly $\frac{\alpha}{2}$. Applying Chernoff bounds, we can prove that if the instance is a Yes-instance, then the verifier's acceptance probability is at least $\frac{2}{3}$ and if it is a No-instance, then the acceptance probability is at most $\frac{1}{3}$. See the proof of Theorem 1 in the next section for an explicit example of applying Chernoff bounds in this cases.

The parallel repetition is to restrict the number of rounds of communication to 2. A great question to ask is whether the prover can coordinate its answers to all of the questions/random bits from the verifier at once in order to improve the probability that the verifier accepts. In this case, it is not possible because the prover's best strategy is to clearly answer each query by providing a certificate, if possible, and the various queries were sampled independently.

Note that if α is not constant, then even with Chernoff bounds we need more than constant repetition of the protocol we discussed above.

Unfortunately, in GNI problem, $|S|$ is much smaller than 2^m. Since the verifier needs to pick a random graph with n nodes, and the number of all graphs with n nodes is almost 2^{n^2} but the size of S is $O(n!) = O(2^{n \cdot \log n})$.

To overcome the issue that $|S|$ is much smaller than the size of the universe $\{0,1\}^m$, we use a collection of hash functions to map the universe to a smaller universe where the value of the "new" α is a constant. The verifier can then sample from this smaller universe. We define this protocol formally in the next section.

Definition 3 (Pairwise Independent Hash Functions) *$\mathcal{H}_{m,k}$, a collection of hash functions $h : \{0,1\}^m \to \{0,1\}^k$, is pairwise independent if $\forall x, x' \in \{0,1\}^m$ where $x \neq x'$ and $\forall y, y' \in \{0,1\}^k$, we have*

$$\Pr_{h \sim \mathcal{H}_{m,k}}[h(x) = y \text{ and } h(x') = y'] = \frac{1}{2^{2k}}.$$

From Definition 3, we have the following immediate observation:

Claim 1 *Let $\mathcal{H}_{m,k}$ be the family of hash functions defined in Definition 3. Then, for given $x \in \{0,1\}^m$ and $y \in \{0,1\}^k$ we have*

$$\Pr_{h\sim\mathcal{H}_{m,k}}[h(x) = y] = \frac{1}{2^k}.$$

Proof. Pick an element $x' \neq x$, then we have

$$\Pr_{h\sim\mathcal{H}_{m,k}}[h(x) = y] = \Pr_{h\sim\mathcal{H}_{m,k}}[(h(x) = y) \cap (\bigcup_{y'\in\{0,1\}^k} h(x') = y')] \tag{12.1}$$

$$= \sum_{y'\in\{0,1\}^k} \Pr_{h\sim\mathcal{H}_{m,k}}[(h(x) = y) \cap (h(x') = y')] \tag{12.2}$$

$$= 2^k \cdot \frac{1}{2^{2k}} \tag{12.3}$$

$$= \frac{1}{2^k}, \tag{12.4}$$

where (12.1) and (12.2) follow from the fact that events $h(x') = y$ for different ys are disjoint and their union is $\{0,1\}^k$. And (12.3) follows from pairwise independence property of $\mathcal{H}_{m,k}$. ■

We have the following result for a collection of hash functions. Say that a collection $\mathcal{H}_{m,k}$ is **implicitly constructible** if we can represent each $h \in \mathcal{H}_{m,k}$ using $\text{poly}(m)$ bits, the value $h(x)$ can be computed in $\text{poly}(m)$ time using the representation for h, and if we can sample $h \sim \mathcal{H}_{m,k}$ uniformly in polynomial time.

Lemma 2 *For $k \leq m$, there is an implicitly-constructible pairwise-independent family $\mathcal{H}_{m,k}$.*

See the Appendix A for the proof of Lemma 2.

12.3 Proof of Theorem 1 (Second Attempt)

In this section, we give a new protocol for set lower bound problem that even works when $|S|$ is very small compare to the size of the universe.

Set Lower Bound Protocol (Goldwasser-Sipser '87)

Repeat α times in parallel where α is a constant (fixed later):

Verifier: Let $k \in \mathbb{N}$ be such that $2^{k-2} < \gamma \leq 2^{k-1}$. Sample $y \sim \{0,1\}^k$, $h \sim \mathcal{H}_{m,k}$, and send these to prover.

Prover: Send $x \in \{0,1\}^m$ such that $h(x) = y$ and a certificate for $x \in S$, if possible.

Verifier: Check if $h(x) = y$ and check if the certificate for $x \in S$ is correct. If both tests are passed, verifier marks this certificate as a valid certificate.

Verifier accepts if the fraction of valid certificates is at least $\frac{5p}{8}$, where $p := \frac{\gamma}{2^k}$.

We comment that the verifier can in fact expose all of their random bits to the prover when sampling y, h, so this is a "public coins" protocol.

The following is the main lemma for proving Theorem 1.

Lemma 3 *Consider one iteration of the set lower bound protocol. Recall $p = \frac{\gamma}{2^k}$. Then, the prover can find a valid certificate with probability $\Pr_{y,h}[\exists x \in S : h(x) = y]$, furthermore:*

1. *If $|S| = \gamma$, then $\forall y \in \{0,1\}^k$, $\Pr_{h \sim \mathcal{H}_{m,k}}[\exists x \in S : h(x) = y] \geq \frac{3}{4}p$, and*
2. *If $|S| \leq \frac{\gamma}{2}$, then $\forall y \in \{0,1\}^k$, $\Pr_{h \sim \mathcal{H}_{m,k}}[\exists x \in S : h(x) = y] \leq \frac{p}{2}$.*

Proof. Proof of part (1): Given $y \in \{0,1\}^k$, for $x \in \{0,1\}^m$ let E_x be the event that $h(x) = y$. Then,

$$\Pr_{h \sim \mathcal{H}_{m,k}}[\exists x \in S : h(x) = y] = \Pr[\bigcup_{x \in S} E_x] \tag{12.5}$$

$$\geq \sum_{x \in S} \Pr[E_x] - \frac{1}{2} \sum_{\substack{x,x' \in S \\ x \neq x}} \Pr[E_x \cap E_{x'}] \tag{12.6}$$

$$= \sum_{x \in S} 2^{-k} - \frac{1}{2} \sum_{\substack{x,x' \in S \\ x \neq x'}} 2^{-2k} \tag{12.7}$$

$$\geq \frac{|S|}{2^k} - \frac{1}{2} \cdot \frac{|S|^2}{2^{2k}} \tag{12.8}$$

$$= \frac{\gamma}{2^k}(1 - \frac{\gamma}{2^{k+1}}) \tag{12.9}$$

$$\geq \frac{\gamma}{2^k}(1 - \frac{2^{k-1}}{2^{k+1}}) \tag{12.10}$$

$$= \frac{3p}{4}, \tag{12.11}$$

where (12.5) follows from inclusion-exclusion principle truncated to the second term, (12.7) follows from definition of pairwise independence and Claim 1, (12.9) holds because of our assumption that $|S| = \gamma$, and finally (12.10) holds since $\gamma \leq 2^{k-1}$. Also note that if $|S| \geq \gamma$, then of course the probability that y has a preimage in S increases; hence, we get a better lower bound than (12.11).

Proof of part (2): This is easy. Since $h(S)$, the set of range of h on set S, has size at most $|S|$, for any $h \in \mathcal{H}_{m,k}$ the probability that y has a preimage in S is at most $\frac{|h(S)|}{2^k} \leq \frac{|S|}{2^k} \leq \frac{\gamma}{2^{k+1}} = \frac{p}{2}$, where the last inequality comes from the assumption that $|S| \leq \frac{\gamma}{2}$ and the equality follows from the value of p defined in the lemma. ■

Note that the set lower bound protocol is a two round interactions, and the verifier sends all the random bits that are used to the prover. Finally, let us apply Chernoff bounds to get the desired bounds on the probabilities of success. Suppose $|S| \geq \gamma$. By Lemma 3, the probability that in one iteration verifier marks a certificate as a valid certificate is at least $\frac{3p}{4}$; thus $\mu := \mathbb{E}[\#\ of\ valid\ certificates] \geq \frac{3p}{4}$. Also note that the probability that verifier rejects is the probability of the event that the number of valid certificate is less than $\frac{5p \cdot \alpha}{8}$. Let $\epsilon := 1 - \frac{5p \cdot \alpha}{8\mu}$. Note that $\frac{1}{6} \leq \epsilon < 1$. So we can apply Chernoff bounds and get:

$$\Pr[\#\ valid\ certificate \leq (1-\epsilon)\mu] \leq e^{-\frac{\mu\epsilon^2}{2}} \leq e^{-\frac{3p \cdot \alpha}{4} \cdot \frac{1}{36} \cdot \frac{1}{2}} \leq e^{-\frac{\alpha}{384}},$$

where the second inequality comes from the fact that $\epsilon \geq \frac{1}{6}$, and the last inequality follows from the fact that $\frac{1}{4} < p$. So if we set $\alpha = 2 \times 384$, the above probability becomes $\leq \frac{1}{4}$, i.e., if $|S| \geq \gamma$, then with probability at least $\frac{3}{4} \geq \frac{2}{3}$, the verifier accepts, as desired. The calculations for the No-instances are almost the same and omitted here.

References

AB09 S. Arora and B. Barak, Computational Complexity: A Modern Approach, *Cambridge University Press*, 2009.

Appendix A: Proof of Lemma 2

Let $\mathrm{GF}(2^m)$ be the field of size 2^m. It is known that $\mathrm{GF}(2^m)$ has a "natural" binary representation in $\{0,1\}^m$ (see below for further low-level comments along this line) in which both addition and multiplication can be done in $\mathrm{poly}(m)$ time. So in the following discussion, $\mathrm{GF}(2^m)$ can be thought of as $\{0,1\}^m$ with additional arithmetic structure.

For two elements $a, b \in \mathrm{GF}(2^m)$, let $h_{a,b} : \mathrm{GF}(2^m) \to \mathrm{GF}(2^m)$ be a hash function such that $h_{a,b}(x) = a \cdot x + b$, where the addiction and multiplication are in $\mathrm{GF}(2^m)$. Then, we define the family of hash functions $\mathcal{H}_{m,m}$ to be $\mathcal{H}_{m,m} := \{h_{a,b} : \forall a, b \in \mathrm{GF}(2^m)\}$. Observe it is trivial to sample from this family: we sample $a, b \sim \mathrm{GF}(2^m)$ simply by flipping m random coins for each and appealing to the fact that elements of $\mathrm{GF}(2^m)$ correspond to $\{0,1\}^m$.

Claim 2 *$\mathcal{H}_{m,n}$ is pairwise independent.*

Proof. Given $x, x' \in \mathrm{GF}(2^m)$ that $x \neq x'$ and $y, y' \in \mathrm{GF}(2^m)$, consider the following system of equations:

$$\begin{cases} a \cdot x + b = y \\ a \cdot x' + b = y' \end{cases}$$

An elegant proof would use the fact that this is a full-rank system for fixed $x \neq x', y, y'$ so there is a unique a, b pair satisfying the system. Here is a more hands-on proof.

From the above system, we have $a \cdot x - y = a \cdot x' - y' \Rightarrow a = (y-y') \cdot (x-x')^{-1}$ (note that $x - x' \neq 0$), and $b = a \cdot x - y$. So a, b are determined uniquely based on x, x', y, y', i.e., there is exactly one pair $(a, b) \in \mathrm{GF}(2^m) \times \mathrm{GF}(2^m)$ such that $h_{a,b}(x) = y$ *and* $h_{a,b}(x') = y'$. Thus, we have

$$\begin{aligned} \Pr_{h_{a,b} \sim \mathcal{H}_{m,m}}[h_{a,b}(x) = y \text{ and } h_{a,b}(x') = y'] &= \Pr_{a,b \sim \mathrm{GF}(2^m)}[a \cdot x + b = y \text{ and } a \cdot x' + b = y'] && (12.12) \\ &= \Pr_{a,b \sim \mathrm{GF}(2^m)}[a = (y-y') \cdot (x-x')^{-1} \text{ and } b = a \cdot x - y] && (12.13) \\ &= \frac{1}{2^{2m}}. && (12.14) \end{aligned}$$

■

To finish the proof of the lemma, we need to show that for $k \leq m$ we can construct $\mathcal{H}_{m,k}$ in $\mathrm{poly}(m)$. But this is easy, we start by constructing $\mathcal{H}_{m,m}$ and then for each $h \in \mathcal{H}_{m,m}$ and $x \in \mathrm{GF}(2^m)$, we truncate the last $m - k$ bits from $h(x)$. Let $\mathcal{H}_{m,k}$ be the set of all such functions. Given a two vectors (not necessarily same size) y, w, let $y|w$ be the concatenation of these two vectors. Then, given $y, y' \in \{0,1\}^k$, we have

$$\Pr_{h_{a,b}\sim\mathcal{H}_{m,k}}[h_{a,b}(x)=y \text{ and } h_{a,b}(x')=y'] = \bigcup_{w,w'\in\{0,1\}^{m-k}} \Pr_{a,b\sim \mathrm{GF}(2^m)}[a\cdot x+b=y|w \text{ and } a\cdot x'+b=y'|w'] \tag{12.15}$$

$$= 2^{2(m-k)}\frac{1}{2^{2m}} \tag{12.16}$$

$$= \frac{1}{2^{2k}}. \tag{12.17}$$

Appendix B: Working with Finite Fields

Some parts are missing from this, if you want further references on this material then please contact the instructor! This assumes more exposure to algebra than is required for the course, so consider this supplementary material that will not be tested in an assignment.

To efficiently work with $\mathrm{GF}(2^m)$, consider the following way to construct $\mathrm{GF}(2^m)$. Let $f(x) \in \mathbb{Z}/2\mathbb{Z}[x]$ be an irreducible polynomial with degree exactly m over the ring of integers modulo 2. By irreducible, we mean that whenever $f(x) = g(x)\cdot h(x)$ where $g(x), h(x) \in \mathbb{Z}/2\mathbb{Z}[x]$ then either $g(x)$ or $h(x)$ is a constant (i.e. $f(x)$ has no nontrivial factors). There is a simple, polynomial-time, deterministic algorithm that, given a proposed $f(x)$, will decide if it is irreducible or not.

Furthermore, there are exactly $\frac{2^m-1}{m}$ irreducible polynomials of degree m. So after, say, $\Theta(m^2)$ samples of degree-m polynomials we will have sampled an irreducible polynomial with probability at least $1-2^{-m}$. We can be certain which one is irreducible (if any) by running the deterministic test mentioned above.

Now one representation of $\mathrm{GF}(2^m)$ can be succinctly described as the quotient ring $\mathbb{Z}/2\mathbb{Z}[x]/(f(x))$ where $f(x)$ is an irreducible polynomial of degree m. That is, the field is just arithmetic in $\mathbb{Z}/2\mathbb{Z}[x]$ modulo $f(x)$.

One can prove that since $f(x)$ is irreducible then every polynomial $g(x)$ that does not vanish modulo $f(x)$ has an inverse $h(x)$ satisfying $g(x)\cdot h(x) \equiv 1 \bmod f(x)$. This can be proven using, say, the Euclidean algorithm for polynomials where we use the following "division algorithm" fact: for any polynomials $g(x), h(x)$ with $h(x) \neq 0$ we have that there are unique polynomials $q(x), r(x)$ (quotient and remainder) satisfying $g(x) = q(x)\cdot h(x)+r(x)$ where $\deg(r) < \deg(h)$ and q has 1 as its leading coefficient. Such $q(x), r(x)$ can be computed efficiently using a grade-school division algorithm.

The 2^m distinct polynomials with degree $\leq m-1$ form a system of distinct representatives for the elements of $\mathbb{Z}/2\mathbb{Z}[x]/(f(x))$ and these correspond in a natural way to $\{0,1\}^m$ by having each bit describe a coefficient of such a polynomial.

Summary
With extremely high probability, in $\mathrm{poly}(m)$ time we can sample an irreducible polynomial $f(x)$ of degree m and verify that it is indeed irreducible. Every $\{0,1\}^m$ is interpreted as a polynomial over integers mod 2 that itself is reduced modulo $f(x)$. Addition, subtraction, multiplication, and inverses are computed using arithmetic of these polynomials modulo $f(x)$. The inverses can be efficiently computed using the Euclidean algorithm for polynomials.