**6.842 Randomness and Computation** September 25-27, 2017

# Lecture 6 & 7

*Lecturer: Ronitt Rubinfeld* *Scribe: Leo de Castro & Kritkorn Karntikoon*

## 1 Interactive Proof Systems

An interactive proof system is a protocol that defines an interaction between two machines, a prover and a verifier. For this model, we will consider verifiers that run in polynomial-time. The verifier will have access to random coins, which for now we will consider private. The prover, on the other hand, is unbounded in time and space. Without loss of generality, we can consider the prover to be a deterministic machine.

The prover and the verifier have a shared input $x$, and the prover attempts to convince the verifier that $x \in L$ for some language $L$. After the interaction with the prover, the verifier will output *Accept* if it believes that $x \in L$ or *Reject* if it believes $x \notin L$. From this, we can define an interactive proof system and the corresponding complexity class of languages with these proof systems.

**Definition 1** *Interactive Proof Systems (IPS) [Goldwasser, Micali, Rackoff]*

A language $L$ has an Interactive Proof System if there is a protocol for an interaction between a polynomial-time verifier $V$ and an unbounded prover $P$ such that if $V$ and $P$ both follow the protocol and $x \in L$, the probability that $V$ outputs *Accept* is at least $\frac{2}{3}$.

$$Pr[V(x) = Accept \mid x \in L] \geq 2/3$$

Note that this probability is taken over the random coins of the verifier. These proof systems must also meet a soundness condition such that if $x \notin L$ and the verifier $V$ follows the protocol, regardless of what the unbounded prover $P$ does, the probability that the verifier rejects is at least $\frac{2}{3}$.

$$Pr[V(x) = Reject \mid x \notin L] \geq 2/3$$

This probability is also taken over the random coins of the verifier.

**Definition 2** *The complexity class IP*

A language $L$ is in the class $IP$ if it has an Interactive Proof System.

It is clear that the class $NP \subset IP$, since the prover $P$ can always just send the short witness that $x$ is in the language, which can be checked in polynomial time by the verifier.

It is also the case that $IP = PSPACE$, which is much less obvious and will not be proved here. It is also a fact that $PSPACE$ is closed under compliment. In order to illustrate the power of IPS, we will construct an interactive proof for graph non-isomorphism.

## 2 Graph Non-isomorphism

Graph isomorphism is in $NP$, so it is also in $IP$. Since $IP = PSPACE$, which is closed under compliment, we know that there exists a IPS for graph non-isomorphism. Here we will show a particularly simple protocol for which the verifier's random coins are private.

The input to this protocol is two graphs $G$ and $H$, both with $n$ nodes. The verifier is supposed to output *Accept* if $G \ncong H$ and output *Reject* if $G \cong H$. The protocol proceeds as follows, which is looped a constant number of times:

1. $V$ uses its private random coins to compute a graph $G'$ that is isomorphic to $G$ and a graph $H'$ that is isomorphic to $H$.

2. $V$ then flips a coin to decide whether to send $(G, G')$ or $(G, H')$ to the prover.

3. $P$ returns a bit indicating the result of coin flipped by $V$ in step 2.

To better illustrate this protocol, here is a table of possible response and action combinations.

| Coin Flip Result | Correct Response if $G \ncong H$ | P response | V output |
|---|---|---|---|
| H | $\cong$ | $\cong$ | Continue |
| H | $\cong$ | $\ncong$ | Reject and Stop |
| T | $\ncong$ | $\cong$ | Reject and Stop |
| T | $\ncong$ | $\ncong$ | Continue |

The only way that the prover can (consistently) determine the result of the coin flip is if the prover can distinguish between $G'$ and $H'$. If $G$ is not isomorphic to $H$, then $H'$ is not isomorphic to $G$, so the prover will always by able to determine whether it was sent $(G, G')$ or $(G, H')$ by simply testing if the second graph is isomorphic to $G$. Therefore, if the two graphs are not isomorphic and the prover and the verifier both follow the protocol, the prover can always determine the result of the coin flip.

However, if $G$ and $H$ are isomorphic, then $G'$ and $H'$ are statistically indistinguishable, since they are equivalent up to isomorphism. Therefore, the prover cannot distinguish between $(G, G')$ and $(G, H')$ better than random guessing. More formally, let $q$ be the fraction of random permutations $\pi$ such that the prover outputs that $(G, \pi(G))$ are not isomorphic. For a given round of the protocol, we have the probability that the prover fails to pass the challenge is:

$$Pr[Prover\ fails\ the\ round] = \frac{1}{2}q + \frac{1}{2}(1-q) = \frac{1}{2}$$

This probability is over the graphs produced by the verifier, since both $G'$ and $H'$ are permutations of $G$. Since the prover cannot do better than random guessing when $G \cong H$, repeating the loop above twice will result in the following probabilities:

$$Pr[V(G,H) = Accept \mid G \ncong H] \geq 3/4$$

$$Pr[V(G,H) = Reject \mid G \cong H] \geq 3/4$$

This is sufficient to satisfy the IPS requirements, so this completes the construction of our private-coin IPS for graph non-isomorphism.

## 3 From Private Coins to Public Coins

In our IPS protocol above, it is crucial that the verifier's private coins are not revealed to the prover, otherwise the prover could always cheat and know how $G'$ and $H'$ were generated. However, in an amazing result by Goldwasser and Sipser, it was shown that public coin IPS are just as powerful as private coin IPS.

**Fact 3** *Public Coin IP = Private Coin IP*

Here, we will begin to construct a public coin IPS for graph non-isomorphism. We will provide an intuition here, and then give a more complete protocol in the next section.

## 3.1 Idea

**Definition 4** *Let $[A]$ be a set of all graphs isomorphic to $A$.*

In this lecture, we only consider graphs with no *non-trivial isomorphism*; i.e., any permutation of labels results in a new graph. Thus, $|[A]| = n!$ where $n$ is the number of nodes in $A$. For two graphs $A$ and $B$, the union $U = [A] \cup [B]$ will have a size of $n!$ if $A \cong B$ and $2(n!)$ if $A \ncong B$.
**Intuition**: If the verifier is able to estimate the size of $U$, then he can justify the isomorphism of $A$ and $B$. We need an IPS that gives some indication of the size of $U$.

In order to do this, consider the universe of all distinct graphs (adjacency matrices) with $n$ nodes $Z_n$. The size of $Z_n$ is $2^{n^2}$, which is the number of ways that $n$ nodes can be connected. We will estimate the size of $U$ from the probability that a random graph from $Z_n$ is in $U$. We have provided two approaches to find this probability in the next section.

# 4 A Protocol for Non-Isomorphism with Public Coins

## 4.1 Naive Approach for Sampling

We can motivate the use of pairwise independent hash functions by further analyzing how long it would take $V$ to estimate the size of $U$ by simply sampling graphs directly.
The protocol will proceed as follows:

1. $V$ sends $P$ to a random n-node graph G from $Z_n$
2. If $G \in U$, then $P$ sends a proof that $G \in U$; otherwise, do nothing
3. If $P$ can prove to $V$ enough number of iterations ($\frac{2(n!)}{|Z_n|}$ of all times), then $|U|$ will be $2(n!)$; otherwise, $|U|$ will be $n!$.

To show that this approach is not feasible. First, we observe that the space of all graphs with $n$ nodes has size $2^{n^2}$. In order to estimate the size of a subset $U$ of all graphs of $n$ nodes, the verifier can determine the probability of sampling a random graph and having that graph be in $U$. This probability is equivalent to $\frac{|U|}{2^{n^2}}$. Since the two cases the verifier is trying to distinguish are when $|U| = n!$ and $|U| = 2(n!)$, the greatest this probability can be is $\frac{2(n!)}{2^{n^2}}$.

Since $V$ cannot hope to estimate the size of $U$ before seeing even a single element in $U$, the expected number of graphs $V$ would have to sample before seeing an element in $U$ is at least $\frac{2^{n^2}}{2(n!)}$. This is not feasible, since the time to run $V$ is bounded by a polynomial in $n$, which is asymptotically smaller than $\frac{2^{n^2}}{2(n!)}$.

In order to avoid this problem, we will define a hash function that maps all graphs to a smaller space, where it is easier to estimate the size of a subspace.

## 4.2 Sampling via Hash Function

The verifier can pick a pairwise independent hash function that maps a space of size $2^{n^2}$ to a space of size $2^l$. The criteria that should be met by this hash function is:

1. $|h(U)|$ is big if and only if $|U|$ is big.
2. $\frac{|h(U)|}{2^l}$ is $\frac{1}{poly(n)}$.
3. $h$ is computable in polynomial time.

The public coin protocol will proceed as follows:

1. Find an integer $l$ such that $n! < 2^{l-1} < 2(n!)$
2. Given $H$, a collection of pairwise independent hash functions mapping $\{0,1\}^{n^2} \rightarrow \{0,1\}^l$, $V$ randomly selects a function $h$ from this family and sends $h$ to $P$.
3. $P$ sends to $V$ a graph $x \in U$ such that $h(x) = 0^l$ along with a proof that $x \in U$.

We will show in the next lemma that if the graphs $A$ and $B$ are not isomorphic, then with high probability the image of the union $[A] \cup [B] = U$ in the hash function will contain $0^l$. If $A$ is isomorphic to $B$, then this occurs with low probability.

**Lemma 5** *Let $h : \{0,1\}^m \rightarrow \{0,1\}^l$ be a pairwise independent function chosen uniformly from $H$, and let $U \subseteq \{0,1\}^m$ and $a = \frac{|U|}{2^l}$ then*

$$a - \frac{a^2}{2} \leq \Pr[0^l \in h(U)] \leq a$$

**Proof of Lemma**

- **Right-Hand Side**: Since $h$ is pairwise independent, then for all $x$ we have $\Pr_h[h(x) = 0^l] = 2^{-l}$. Using union bound, we get

$$\Pr[0^l \in h(U)] \leq \sum_{x \in U} \Pr[0^l = h(x)] \leq \frac{|U|}{2^l} = a$$

- **Left-Hand Side**: Recall the inclusion-exclusion principle:

$$\Pr[\bigcup_{i=1}^{n} A_i] = \sum_{i=1}^{n} \Pr[A_i] - \sum_{1 \leq i < j \leq n} \Pr[A_i \cap A_j] + \sum_{1 \leq i < j < k \leq n} \Pr A_i \cap A_j \cap A_k - \ldots + (-1)^{n-1} \Pr[A_1 \cap A_2 \cap \ldots A_n]$$

From Boole's inequality (cutting tail of inclusion-exclusion formula), we have

$$\Pr[\bigcup_{i=1}^{n} A_i] \geq \sum_{i=1}^{n} \Pr[A_i] - \sum_{1 \leq i < j \leq n} \Pr[A_i \cap A_j]$$

Applying previous inequality on set $U$ where $A_x$ denotes the event that $h(x) = 0^l$, we get

$$\begin{aligned} \Pr[0^l \in h(U)] &\geq \sum_{x \in U} \Pr[0^l = h(x)] - \frac{1}{2} \sum_{\substack{x \neq y \\ x,y \in U}} \Pr[h(x) = h(y) = 0^l] \\ &= \sum_{x \in U} 2^{-l} - \frac{1}{2} \sum_{\substack{x \neq y \\ x,y \in U}} 2^{-2l} \\ &= \frac{|U|}{2^l} - \binom{|U|}{2} \frac{1}{2^{2l}} \\ &\geq a - \frac{|U|^2}{2} \cdot \frac{1}{2^{2l}} \\ &= a - \frac{a^2}{2} \end{aligned}$$

■

**Finishing Up**
Follow the protocol described above, we will use Pr[Verifier accepts in each round] to justify the isomorphism

- If $A \ncong B$, then $|U| = 2(n!)$. Since $2^{l-1} \le 2(n!) < 2^l$, we have $\frac{1}{2} \le a \le 1$. Thus,

$$\Pr[\text{Verifier accepts in each round}] \ge a - \frac{a^2}{2} \ge \frac{3}{8}$$

- If $A \cong B$, then $|U| = n!$. Since $n! < 2^{l-1}$, we have $a \le \frac{1}{2}$. Thus,

$$\Pr[\text{Verifier accepts in each round}] \le \frac{1}{2}$$

**Problem!** If Pr[V accepts in each round] $\in \left[\frac{3}{8}, \frac{1}{2}\right]$, we will not be able to distinguish these two cases[1].
**Idea**: We can multiply the difference between two cases by considering $(Z_n)^m$ instead of $Z_n$, and $U_m = U^m$. Thus, the size of $U_m$ will be $(2k)^m$ when $A \ncong B$ and $k^m$ otherwise.

This idea can be used in general theorem: $IP_{privatecoins} = IP_{publiccoins}$ by arguing that the size of accepting region probability is large comparing to another one. (Need to be able to verify a conversation/random coin if it is in accept region)

[1]The protocol we're describing here is the Goldwasser-Sipser lower bound protocol. For more notes on the complete protocol, see http://zoo.cs.yale.edu/classes/cs468/spr15/lectures/GSLB.pdf (or just Google search).