6.842 Randomness and Computation	February 27, 2012

Lecture 6

Lecturer: Ronitt Rubinfeld *Scribe: Osbert Bastani*

Today, we will cover

1. the definition of **IP**,
2. Graph $\ncong \in$ **IP**,
3. **IP** (public coins) v.s. **IP** (private coins),
4. protocol for lower bounding set size.

1 The Complexity Class IP

Recall the definition of the class **NP**:

Definition 1. The class **NP** is the class of all decision problems for which "yes" answers can be verified in polynomial time by a deterministic Turing machine.

Similarly, the complexity class **IP** is the class of languages for which there is a short "interactive" proof that $x \in L$. To formalize the notion of an interactive proof, we define an **interactive proof system**.

Definition 2. Consider the following model (see 1):

1. a deterministic unbounded time prover P,
2. a randomized polynomial time **verifier** V,
3. a pair of conversation tapes on which P and V exchange information.

An **interactive proof system** (IPS, due to Goldwasser, Micali, and Rackoff) for a language L, is a protocol between P and V where

1. P and V are given an input x,
2. through an exchange of messages, P tries to prove to V that $x \in L$,
3. at the end of the interaction, V outputs either "accept" if the proof is satisfactory or "reject" if not.

We require that

1. if both P and V follow the protocol and $x \in L$, then
$$\Pr[V \text{ accepts } x] \geq \frac{2}{3},$$
2. if $x \notin L$ and V follows the protocol, then, regardless of what P does,
$$\Pr[V \text{ rejects } x] \geq \frac{2}{3}.$$

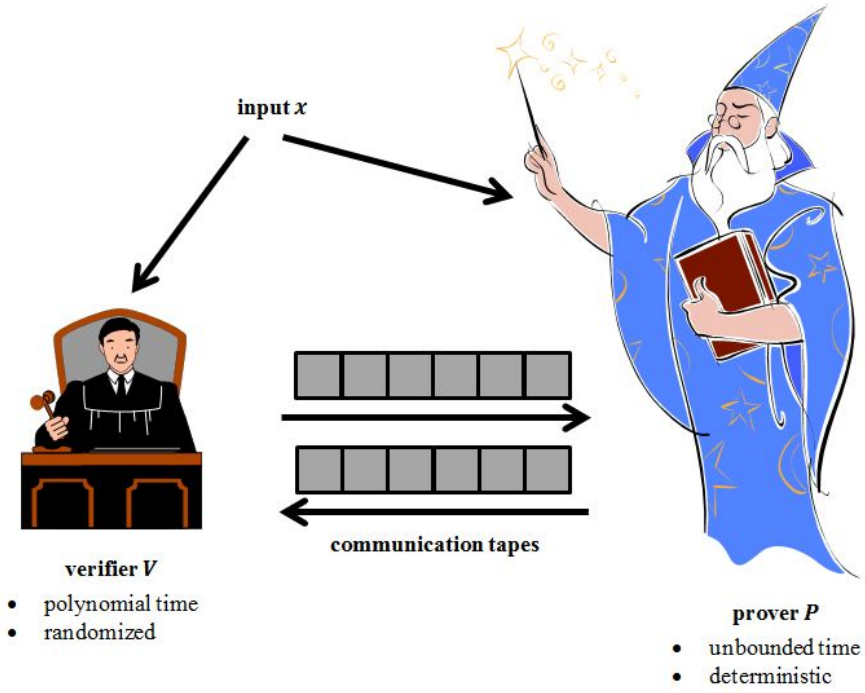

Figure 1: The model for the interactive proof system.

Remark The probabilities are taken over the coins of the verifier. In particular, they are independent of the action of P in the case $x \notin L$. As usual, the probabilities can be boosted as in the case of randomized algorithms.

Remark It does not help the prover to be randomized. Since he has unbounded time, he can compute probabilities and determine the best strategy (i.e. try over all possible random coins).

Definition 3. The class **IP** is the class of languages L such that there exists an IPS for L.

Remark For now we assume that the random coins are private, i.e. prover does not know what the random bits are, but we will later show that **IP** with private random coins = **IP** with public random coins.

Remark We can then think of **NP** as an IPS where P sends V exactly one message. Hence **NP**$\subset$**IP**. In general the interaction may cut down the size of the proof.

2 Graph Nonisomorphism

Consider the following decision problems:

Example 4. Consider the problem **Graph Isomorphism** (GI):

1. **Input:** a pair of graphs (G, H)

2. **Output:** whether or not $G \cong H$, i.e.

$$\exists \pi \in S_{|V_G|} \text{ s.t. } (u,v) \in E_G \Leftrightarrow (\pi(u), \pi(v)) \in E_H.$$

GI $\in$ **NP**, since we can simply give π. Hence GI $\in$ **IP**.

Example 5. Consider the problem **Graph Nonisomorphism** (GNI):

1. **Input:** a pair of graphs (G, H)
2. **Output:** whether or not $G \ncong H$.

By the above, GNI $\in$ **co-NP**, but it is not known whether GNI $\in$ **NP**. However, we know that GNI $\in$ **IP** (due to Goldreich, Micali, and Wigderson).

Theorem 6. *We have GNI* $\in$ **IP**.

Proof. We show that GNI has an IPS.

Algorithm 7. Consider the following protocol:

1. **Input:** graphs G, H,
2. V computes $G' =$ a random permutation of G, and $H' =$ a random permutation of H,
3. V flips a coin c,
   (a) if $c = 0$: V sends (G, G') to P,
   (b) if $c = 1$: V sends (G, H') to P,
4. P sends V either $\delta =$ " $\cong''$ or $\delta =$ " $\ncong''$,
   (a) if ($c = 0$ and $\delta =$ " $\cong''$) or ($c = 1$ and $\delta =$ " $\ncong''$), then V accepts,
   (b) if ($c = 0$ and $\delta =$ " $\ncong''$) or ($c = 1$ and $\delta =$ " $\cong''$), then V rejects.

Note that if G and H are isomorphic, then a random permutation of G has exactly the same distribution as a random permutation of H, so the prover cannot tell the difference and will fail with probability $\frac{1}{2}$. More precisely, we have

Lemma 8. *The protocol is an IPS for GNI.*

Proof. 1. If $x \in L$, i.e. $G \ncong H$, then since P has unbounded time, he can always answer correctly so that V continues.

2. If $x \notin L$, i.e. $G \cong H$, then the distribution of V's messages is the identical in both cases $c = 0$ and $c = 1$. Recall that we assume that P is deterministic. Let

$$q := \text{fraction of random permutations that cause } P(G, \pi(G)) \text{ to answer } \delta = \ncong .$$

Then

$$\Pr[\text{fail in any round}] = \frac{1}{2}q + \frac{1}{2}(1-q) = \frac{1}{2}.$$

□

Since there exists an IPS for GNI, we have GNI $\in$ **IP** as desired. □

Remark After k rounds, if $x \in L$, then

1. if $x \in L$, then $\Pr[V \text{ accepts}] = 1$,
2. if $x \notin L$, then $\Pr[V \text{ accepts}] \leq 2^{-k}$.

Remark In fact, we have **IP** = **PSPACE** (due to Shamir, building on [LFKN]). The proof uses O(size of problem) rounds in the interaction. Intuitively each round reduces the size of the problem by 1 by eliminating a single variable.

3 Arthur-Merlin Games

Definition 9. An Arthur-Merlin game is an IPS such that V's random tape is public.

Theorem 10. *(Goldwasser, Sipser) GNI has an IPS using only public coins.*

Remark In fact, **IP** with public coins = **IP** with private coins. There is a protocol that turns a protocol with k rounds using private coins into a protocol with $k+2$ rounds using public coins.

Proof. Let

$$[A] := \text{ graphs that are } \cong \text{ to } A.$$

For convenience, assume that A and B are graphs with n nodes and with no nontrivial automorphisms (i.e. isomorphisms onto itself). Then

$$|[A]| = |[B]| = n!.$$

Let $U = [A] \cup [B]$, so that

$$|U| = \begin{cases} n! \text{ if } A \cong B, \text{ i.e. } U \text{ is small} \\ 2n! \text{ if } A \ncong B, \text{ i.e. } U \text{ is big} \end{cases}.$$

We have to show is that if U is big, then the prover can convince us that U is big, and if U is small, then the prover cannot convince us that U is big. One idea is to use random sampling. Consider the following algorithm:

1. V sends P a random n-node graph G,
2. P sends V
   (a) if $G \in U$, π such that $\pi(G) = A$ or B,
   (b) if $G \notin U$, then $\emptyset$,
3. V verifies π.

This is repeated several times, and V outputs

$$\delta = \frac{\#\text{successful trials}}{\#\text{trials}}.$$

Unfortunately, U is tiny compared to the space of all n-node graphs, so the expected number of rounds needed to bound the error is exponentially large. More precisely,

$$\frac{|U|}{|\#\{n\text{-node graphs}\}} \leq \frac{2n!}{2^{n^2}} \approx 2^{n \log n - n^2}.$$

To remedy this, we will use universal hashing. Note that the size of all n-node graphs is approximately $2^m = 2^{n^2}$. We need a hash function $h : [2^m] \to [2^l]$ such that

1. $|h(U)| \approx U$,
2. $h(U)$ is large if and only if U is large,
3. $h(U)$ is not too small, i.e.

$$\frac{|h(U)|}{2^l} = \frac{1}{\text{poly}(m)},$$

4. h is computable in polynomial time.

Definition 11. Recall that H is **pairwise independent** if

$$\forall x \neq y \in \{0,1\}^m \forall a, b \in \{0,1\}^l \Pr[h(x) = a \text{ and } h(y) = b] = 2^{-2l}.$$

Let H be a collection of pairwise independent functions mapping $\{0,1\}^m \to \{0,1\}^l$. We need to choose l such that

$$2^{l-2} < n! < 2^{l-1} < 2n! < 2^l.$$

If $G \ncong H$, then $|U| = 2n!$. If $G \cong H$, then $|U| = n!$. Since h is chosen from a family of pairwise independent hash functions, whether a given node is covered is independent of whether another given node is covered. More precisely,

1. V picks $h \in_R H$ and sends it to P,
2. P sends V
   (a) $x \in U$ such that $h(x) = 0^l$ along with a proof that $x \in U$ (i.e. the isomorphism to A or B), if such an x exists,
   (b) $\emptyset$ otherwise,

Lemma 12. *Assume H is pairwise independent. Let $U \subset \Sigma^m$ and $a = \frac{|U|}{2^l}$. Then*

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

Proof. To see the right-hand side, note that

$$\forall x \Pr_{h \in H}[0^l = h(x)] = 2^{-l},$$

so

$$\Pr_{h \in H}[0^l \in h(U)] \leq \sum_{x \in U} \Pr[0^l h(x)] = |U| \cdot 2^{-l} = a.$$

To see the left-hand side, note that by the inclusion-exclusion principle,

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

□

Finally, pick l such that $2^{l-1} \leq 2n! \leq 2^l$. If $G \ncong H$, then $|U| = 2n!$, so $\frac{1}{2} \leq a \leq 1$, so

$$\Pr[V \text{ accepts}] \geq a - \frac{a^2}{2} \geq \frac{3}{8} =: \alpha.$$

If $G \cong H$, then $|U| = n!$, so $\frac{1}{4} \leq a \leq \frac{1}{2}$, so

$$\Pr[V \text{ accepts}] \leq a \leq \frac{1}{2} =: \beta.$$

Unfortunately, we need $\alpha > \beta$. This will be fixed on the homework. □