18.405J/6.841J Advanced Complexity Theory	April 5, 2001
Lecture 15	
Lecturer: Shanghua Teng	*Scribe: Gregory Dennis*

Motivation

When we first learned of NP, we defined it as the class of all languages with polynomical time verifiers. Thus if we have a language $L \in NP$, a word w could be said to be in L if and only if some prover could convince $L's$ verifier that w's membership. This lead us to believe that we could define certain languages as the set of words, each of which a prover could convince a verifier of the word's membership in that language.

Interactive Proofs

So what are Interactive proofs? Imagine two parties and the conversation between them. One party is an "all-powerful" *Prover*, which we denote P. The Prover is a prodigy and has unlimited computational power. The second party is the *Verifier*, which we denote V. The Verifier is a polynomial time Turing Machine, which has access to randomness in the sense that it can toss coins and ask different questions depending on the outcome. The coins are kept *private* to the Verifier. The conversation is a sequence of "questions" from V and "answers" from P in which the Prover is trying to convince the Verifier of some statement by answering the Verifier's questions. The questions and answers are restriced to being polynomial in the length of the input.

Concrete Example: Colored Balls
Say you wanted to convince your colorblind friend that two balls, one red and one blue but otherwise indistinguishable, are dfferent. He cannot for himself tell them apart. Let him hold the balls so that you may see them and say "The ball in your right hand is red." Now, if you are right in that you can distinguish between the balls, you should have no problem telling which ball is which if he put them behind his back and juggled them up. If, on the other hand, you were colorblind like your friend, you would have only a 50% chance of guessing correctly. Hence, by repeating the experiment of exchanging (or not exchanging) the balls behind his back and showing them to you, your friend can be convinced to any degree of certainty that the balls are in fact distinguishable. If both balls were red, however, and therefore indistinguishable to you at a glance, you have again only a 50% chance of guessing right. In this case, your friend can easily tell that you cannot tell the balls apart.

Definition 1 *An **Interactive Proof** consists of a Prover function* P(word, history) *with unlimited computational power, a polynomial-time Verifier function* V(word, history, random bit string)*, and a conversation:*

$$
\begin{aligned}
&V(w, \epsilon, r) \to V_1 \\
&P(w, V_1) \to P_1 \\
&V(w, V_1 \cdot P_1, r) \to V_2 \\
&P(w, V_1 \cdot P_1 \cdot V_2) \to P_2 \\
&\vdots \\
&V(w, V_1 \cdot P_1 \cdot V_2 \cdots P_{f(|w|)}, r) \to \{accept, reject\}
\end{aligned}
$$

We say $L \in IP$ *if* $\exists V \in P$ *s.t.* $\forall w \in L$

$$w \in L \implies \exists\ Prover\ P:\ \Pr_r[P \leftrightarrow V(w,r) = accept] > \tfrac{2}{3}$$
$$w \notin L \implies \forall\ Provers\ P:\ \Pr_r[P \leftrightarrow V(w,r) = accept] < \tfrac{1}{3}$$

For IP *the function f is polynomial. We can also define* IP*(k) with f(n) = k for the languages which accept or reject after a constant, k, number of rounds.*

Theorem 2 $IP(k) = IP(1)$*, for constant k*

This means that nothing is gained by having a constant number of rounds compared to a single round.

Graph Isomoprhism
A classic problem in IP is Graph Non-Isomorphism, which we will define as follows:

NONISO = $\{(G_1, G_2)$: $\forall$ permutations $\pi,\ G_1 \neq \pi(G_2)\}$

Theorem 3 $NONISO \in IP$

Proof This is not exactly a formal proof, but it does address the major ideas. In this case, we can think of the Prover as trying to convince the Verfier that two graphs are non-isomorphic. The interaction between the Prover and the Verfier proceeds as follows:
Verfier: Pick G_i randomly one from $\{G_1, G_2\}$ and then randomize G_i's vertex numbering. Formally, let H = $\pi(G_i)$ where π is a random permutation. Show H to the Prover and ask it of which graph it is a permutation, $G_0 or G_1$.
Prover: Compare H to G_1 and G_2. If H is isomorphic to just one of $\{G_1$,$G_2\}$, then respond with G_i, where $\pi(G_i)$ = H. If H is isomorphic to both G_1 and G_2, then respond randomly with either G_1 or G_2.
Verfier: If the Prover is correct, accept. Otherwise, reject.
This "proof" works for exactly the same reason as it did for the colored balls. For $G_1 \ncong G_2$, the Prover can make the Verfier accept. For $G_1 \cong G_2$, however, the distribution over G_1 is the same as that over G_2. Hence the Prover can guess correctly only 50% of the time. ■

Zero-Knowledge Proofs

Zero knowledge proofs are interactive proofs that yield nothing beyond the validity of the assertion. That is, the only information the Verfier gains from the conversation is that the Prover is correct or that the Prover is incorrect. We will now show that Graph Isomorphism, which we denote ISO, has a zero-knowledge proof.

Theorem 4 *ISO has a Zero-Knowledge Proof*

Proof This is not exactly a formal proof, but it does address the major ideas. For two graphs G_0 and G_1, we think of the Prover as trying to convince the Verifier that they are isomorphic. If they are isomorphic, assume without loss of generality that $\pi(G_0) = G_1$. The interaction between the Prover and the Verifier proceeds as follows:
Prover: Picks a random permutation σ and sends $H = \sigma(G_0)$ to the Verifier.
Verifier: Flips its coin to obtain a random bit $r \in \{0,1\}$. The Verifier then asks the Prover for a permutation that proves $G_r \cong H$.

Prover: If r = 0, the Prover responds with σ. If r = 1, it responds with $\sigma\pi^{-1}$.
Verifier: If the permutation sent by the Prover shows $G_r \cong H$, then accept. Otherwise, reject. (**Note:** Since $\pi(G_0) = G_1$ and $H = \sigma(G_0)$, $H = \sigma\pi^{-1}(G_1)$.)

If $G_0 \cong G_1$, then the Prover will always be able to convince the Verifier of the isomorphism. And if $G_0 \ncong G_1$ and $r = 0$, the Verifier will think the Prover if correct, because $H = \sigma(G_0)$ regardless. But if $G_0 \ncong G_1$ and $r = 1$, the Prover will not be able to create a permutation $\sigma\pi^{-1}$, because therre is no π for which $G_0 \cong G_1$. Thus, if $G_0 \ncong G_1$, the Prover will be right at most 50% of the time. More importantly, notice that V never learns what the actual permutation π is, and thus it learns nothing from the conversation other than whether G_0 and G_1 are ismorphic. We have, therefore, created a zero-knowledge proof for ISO. ■

Arthur Merlin Games

If we look back at the definition of IP, we can see that the Prover P does not have access to the random bits, r. The Prover gets only the word and the conversation history. What would happen if P did have access to r? That is, what would happen if we gave the Verifier V a public coin instead of a private one? This type of model is referred to as an Arhur Merlin game.

Definition 5 *An* ***Arthur-Merlin Game*** *consists of a Prover function* M(word, sequence of random bit strings) *with unlimited computational power, a polynomial-time Verifier function* A(word)*, and a conversation:*

$$
\begin{aligned}
&A(w) \to r_1\\
&M(w, r_1) \to M_1\\
&A(w) \to r_2\\
&M(w, r_1, r_2) \to M_2\\
&\vdots\\
&A(w, r_1, ..., r_{f(|w|)}, M_1, ..., M_{f(|w|)}) \to \{accept, reject\}
\end{aligned}
$$

We say $L \in AM$ *if* $\exists A \in P$ *s.t.* $\forall w \in L$

$$
\begin{aligned}
w \in L &\implies \exists M: \Pr_{r_1,\ldots,r_{f(|w|)}}[M \leftrightarrow A(w, r_i) = accept] > \tfrac{2}{3}\\
w \notin L &\implies \forall M: \Pr_{r_1,\ldots,r_{f(|w|)}}[M \leftrightarrow A(w, r_i) = accept] < \tfrac{1}{3}
\end{aligned}
$$

For AM *the function f is polynomial. We can also define* AM*(k) with f(n) = k for the languages which accept or reject after a constant, k, number of rounds.*

Theorem 6 $AM(k) = AM(1)$ *for constant k*

This means that nothing is gained by having a constant number of rounds compared to a single round.

Theorem 7 $AM(f(n)) = AM(\frac{f(n)}{2} + 1)$

This means that it is always possible to speedup the protocol by a factor of 2. You can apply this theorem recursively, but only a constant number of times, because otherwise the size of each message between Arthur and Merlin blows up exponentially.

Theorem 8 $AM(\Theta(f(n))) = IP(f(n))$

This means that private coins are equivalent to public coins.

Theorem 9 *If ISO is NP-complete,* $\Sigma_3^P = \Pi_3^P$*.*