18.405J/6.841J Advanced Complexity Theory	April 14, 1998
Lecture 17	
Lecturer: Daniel A. Spielman	*Scribe: Anna Charny*

Today we will begin a completely new topic. We shall study Interactive Proofs. The notion of interactive proofs was designed to capture those languages in which one can be convinced of membership. This conviction may be obtained through communication rather than the mere presentation of a proof.

1 Publishable Proofs

The first topic we will study is *Publishable Proofs*, introduced by Babai. The idea is to reduce the problem to some probabilistic statement that can be (probabilistically) verified. We have actually seen publishable proofs in the last homework. Indeed, $\sum \cdot BP \cdot P$ is really what a publishable proof is. That is, L has publishable proofs (i.e. $L \in \sum \cdot BP \cdot P$) iff there exists a polynomial time Turing Machine V (which we shall call a *verifier*) such that $w \in L \Longrightarrow \exists x$ s.t. for at least 2/3 of all choices of y $V(w, x, y)$ accepts

$w \notin L \Longrightarrow \forall x$ for at most 1/3 of all choices of y $V(w, x, y)$ accepts,

where $|x|$ and $|y|$ are polynomial in $|w|$.

2 Interactive Proofs

Publishable proofs is just one approach to the problem. *Interactive Proofs* is another.

The idea of Interactive Proofs can be described as follows. Suppose you have a conversation between two parties. One party is an "all-powerful" *Prover*, which we denote by P. The Prover is a "really smart being", which can be described by an arbitrary function. The second party to the conversation is *Verifier,* which we denote by V. The Verifier is a polynomial time Turing Machine, which has access to randomness in the sense that it can toss random coins and ask different questions depending on the outcome. The coins are kept *private* to the Verifier. The conversation is a sequence of "questions" and "answers" between V and P in which the Prover is trying to convince the Verifier of some statement by answering the Verifier's questions.

It is essential that V has access to randomness because otherwise we will never escape NP. In the absence of randomness available to V, P can predict exactly what V will ask and precompute the whole sequence of questions and answers, so in fact V would not be needed. If V can toss coins, however, P cannot predict what the question is going to be.

One example of what Interactive Proofs are is graph *non-isomorphism*. Recall that two graphs G, H are called *isomorphic* if there exists a permutation π of vertices of graph G satisfying $\pi(G) = H$.

We define $NONISO$ as a language of all pairs of graphs (G, H) such that G and H are **not** isomorphic. How do we constuct an interactive proof for $NONISO$? In general, how do we construct interactive proofs?

A simple example of an interactive proof is a protocol for convincing a colorblind friend that colors exist. The protocol is as follows. You get two balls of different color (say red and blue) which are otherwise identical. You give the balls to your colorblind friend and tell him that he has a red ball in his right hand and a blue ball in his left hand. The friend then hides the balls behind his back and randomly shuffles them, so that he knows which ball is in which hand but you don't. He then shows them to you and you correctly identify the balls.

We now give an interactive proof for $NONISO$. We specify the protocol by which P will convince V that two graphs G and H are non-isomorphic. We assume that G and H have the same number of vertices, since otherwise the graphs are obviously non-isomorphic. Let n denote the number of vertices of either graph.

Actions for V:

1. Flip a random bit $b \in \{0, 1\}$; b is a "fair" coin with probability $1/2$ of either outcome.

2. Choose a permutation π of $\{1, 2, ...n\}$. If $b = 0$ show the Prover $\pi(G)$. Otherwise show $\pi(H)$

In response the Prover says which graph is being shown to it.

This can be repeated as many times as needed to boost confidence. As long as the prover guesses correctly, the verifier should become confident that graphs are non-isomorphic.

Proof of Correctness.

We use notation (V, P) to denote the outcome of the protoocol.

If G is not isomorphic to H then there exists a prover P s.t. $Pr[(V, P) \text{ accept}] = 1$

If G is isomorphic to H, then for any P, on k runs of the protocol $Pr[(V, P)$ accept$] \leq \frac{1}{2^k}$. The latter is because P cannot distinguish between two isomorphic graphs.

In general, the key elements of an Interactive Proof are

1. players V and P take turns

2. after a certain number of turns they stop and V decides whether to accept or reject.

We can treat V as a function $V(w, r, C) \to \{S, accept, reject\}$, where w is an input string, r is a random string and C and S are strings denoting respectively the conversation so far and V's response. More specifically, the output of this function is defined as follows:

if $C = \varepsilon$ output $S = V_1$

if $C = V_1, P_1, ...V_i, P_i$ and $i < k$ (where k is the chosen total number of rounds of the protocol) then output V_{i+1}

if $C = V_1, P_1, ...V_k, P_k$ output *accept* or *reject*

Similarly, we can treat P as a function $P(w, C) \rightarrow S$, where S is P's response string. If $C = V_1, P_1, ...V_i$, then P outputs P_i.

Now we can define class $IP(k)$ as a class of all languages that have interactive proofs consisting of k rounds. We say that a language $L \in IP(k)$ if there exists a polynomial-time Turing Machine V that rejects or accepts after k rounds s.t.

$w \in L \Longrightarrow \exists P$ s.t. $\underset{r}{Prob}\left[(V(w,r), P(w)) \text{ accept}\right] \geq 2/3$

$w \notin L \Longrightarrow \forall P\ \underset{r}{Prob}\left[(V(w,r), P(w)) \text{ accept}\right] \leq 1/3$

The IP model is sometimes called a "private coins" model because the prover does not get the coin-flips r of the verifier as input. We now discuss "public coins".

3 Public Coins

Public coins model is also known as Arthur-Merlin games. The idea is that Arthur (the verifier) flips random coins at each round, but unlike the private coins model, the coins are visible to Merlin (the prover) as well. Merlin then tries to persuade Arthur of the answer. Note that it is important that the coins are tossed at each round, rather than all at once, because otherwise we can't escape $BP \cdot \sum \cdot P$ (for the same reason as the fact that if we had no randmoness at all in interactive proofs, we wouldn't escape NP). Note also that since Merlin is all-powerful, he does not need to see anything but the coin-tosses of Arthur, because once Merlin knows the coin tosses, he can compute what Arthur is going to ask.

The behavior of the Verifier $V(w, C)$ (Arthur) can then be described as follows:

if all k rounds are done V decides whether to accept or reject

if less then k rounds are done, V outputs some random bits

This leads to complexity classes $AM(k)$ and $MA(k)$, where k is the number of rounds and the order of letters A and M is determined by who (Arther or Merlin) starts the conversation. We will use AM for $AM(1)$ and MA for $MA(1)$ (this notation will become clear once we prove one of the theorems stated below)."

It turns out that

$\sum \cdot BP \cdot P = MA$

$BP \cdot \sum \cdot P = AM$

We will later prove several theorems.

Theorem 1 *$AM(k) = AM$ for all constants k*

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

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

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 3 $AM(f(n)) = IP(f(n))$

This means that private coins are equivalent to public coins

These results can be used to prove the following theorem about the graph isomorphism problem (ISO):

Theorem 4 *If ISO is NP-complete then* $\sum_3 P = \Pi_3 P$

Proof.

It follows from the protocol for $NONISO$ and the proof of its correctness that $NONISO \in IP(4)$. Hence, from theorems 1-3 it follows that $IP(4) \subseteq AM(4) \subseteq AM$. Recalling that $BP \cdot \sum \cdot P = AM$, by homework we get $AM \subseteq NP/poly$. If ISO is NP-complete, then $NONISO$ is $co-NP$ complete, and hence $co-NP \subseteq NP/poly$. Therefore, by a homework problem, $\sum_3 P = \Pi_3 P$.

Note: In fact, one can show NONISO is NP-complete implies $\sum_2 P = \prod_2 P$