18.405J/6.841J Advanced Complexity Theory April 10, 2001

Lecture 16

Lecturer: Daniel A. Spielman *Scribe: Paul M. Chang*

In the last lecture, we showed that interactive proofs exist for difficult problems such as graph non-isomorphism. Our construction depended on the verifier's ability to flip private coins out of the prover's sight. In this lecture, we will show that the same results can be achieved even if the coin flips are public.

1 Introduction to Arthur-Merlin Games

The Prover/Verifier private coin model corresponds to the complexity class IP. It was hoped that IP would contain NP but be a bit harder. A set of public coin complexity classes also developed around the same time, based on the concept of "Arthur/Merlin games". In our model, Merlin is an all-knowing prover attempting to convince Arthur of some fact. We can think of Arthur as a polynomial time Turing Machine. We can then define different complexity classes depending on allowed interactions between Arthur and Merlin.

The complexity class MA corresponds to a system in which Merlin speaks and then Arthur decides whether or not to believe him. This corresponds to the complexity class of publishable proofs that the verifier can probabilistically verify (recall the verification of a polynomial identity). This corresponds also to the class $\Sigma \cdot BP \cdot P$.

The complexity class AM corresponds to the case where Arthur speaks, Merlin speaks, and then Arthur decides, or $BP \cdot \Sigma \cdot P$.

We can similarly define the classes MAM and AMA.

2 Formalism

Definition 1 *A* **p(n)-prover P** *is a function* $P : \{0,1\}^*\{0,1\}^* \to \{0,1\}^*$,

$$P(\omega, T) \to t \in \{0,1\}^{p(|\omega|)}$$

where ω *is the word we are reasoning about,* T *is the transcript of conversation so far, and* t *is the prover's output.*

Our verifier Arthur is a polynomial time Turing Machine. Arthur flips coins, performs some processing, and outputs a string to Merlin. At the end of the conversation, Arthur accepts or rejects. We note that Merlin is more powerful and can simulate any processing Arthur does. Thus, Arthur doesn't need to output anything except the results of his coin flips.

Let A be a polynomial time verifier. A $k(n)$ round conversation $AM(k(n))$ looks as follows:

$r_1 \leftarrow \{0,1\}^{p(|\omega|)}$
$t_1 \leftarrow P(\omega, r_1)$
$r_2 \leftarrow \{0,1\}^{p(|\omega|)}$
$t_2 \leftarrow P(\omega, r_2)$
...
$r_k \leftarrow \{0,1\}^{p(|\omega|)}$
$t_k \leftarrow P(\omega, r_k)$
$A(\omega, r_1t_1, r_2t_2, ...r_kt_k) \to$ accept or reject.

Definition 2 *A language $L \in AM(k(n))$ if $\exists$ a polynomial $p(n)$ and a PTIME verifier A such that*
$\omega \in L \Longrightarrow \exists$ *a $p(n)$-Prover P such that $Pr[(A \longleftrightarrow P) = accept] > \frac{2}{3}$*
$\omega \notin L \Longrightarrow \forall$ *$p(n)$-Provers P, $Pr[(A \longleftrightarrow P) = accept] < \frac{1}{3}$*

Amplifications are possible if you converse for more rounds or play multiple games each round. However, we need to be careful: a notable paper in the field was incorrect because it assumed probability amplification in a case where it did not hold.

3 The Main Theorem

Theorem 3 *Goldwasser and Sipser:*
$MA(k(n)) = IP(k(n))$ for all reasonable $k(n)$.

We will not prove this theorem in this lecture. It is reasonably clear that $MA(k(n)) \subseteq IP(k(n))$. We will, however, show that a special case of graph non-isomorphism is in MA. This proof contains all of the major ideas used in the proof of the reverse direction.

4 Graph Automorphisms

Let us first recall some useful facts about graphs. Two graphs are equal if they have the same vertices and edge lists. Two graphs are isomorphic if there is a permutation of the vertex numbers which makes them equal. Two other concepts that will be useful are the automorphism group of a graph G, $Aut(G)$, and the orbit of a graph, $Orbit(G)$.

Definition 4 $Aut(G) = \{Permutations\ \pi | \pi(G) = G\}$.

Definition 5 $Orbit(G) = \{H | \exists$ *a permutation π such that $\pi(G) = H\}$.*
Note that the orbit of a graph G is its isomorphism class. These are complementary ideas, since $|Aut(G)||Orbit(G)| = n!$.

Consider the following promise problem: Let G and H be connected graphs with a trivial automorphism group. If they have different numbers of nodes, the proof of non-isomorphism is trivial. We can thus assume that G and H each has n nodes.

Define $G \cup H$ to be the graph obtained by taking the union of the vertex and edge sets of G and H. Intuitively, a picture of $G \cup H$ is obtained by drawing G next to H. Assume that the vertex sets of G and H are disjoint, so that $G \cup H$ has $2n$ nodes. Then, $G \cup H$ is a disconnected graph with two components. If G and H are non-isomorphic, $Aut(G \cup H) = Aut(G) * Aut(H) = I$. If G and H are isomorphic, $Aut(G \cup H) = \{I, flip\}$, where flip is a permutation which exchanges the vertices of G with the vertices of H. Then,

- If G and H are not isomorphic, then $|Orbit(G \cup H)| = (2n)!$.
- If G and H are isomorphic, then $|Orbit(G \cup H)| = (2n)!/2$.

5 The Main Idea

Let us first consider an approach which seems plausible but doesn't quite work, and modify it to make it work. We notice that if G and H are non-isomorphic, a random graph J on $2n$ nodes is twice as likely to be in the orbit of $G \cup H$ than if G and H were isomorphic. We also note that if J is in the orbit of $G \cup H$, Merlin can convince Arthur of this fact by sending the permutation π such that $\pi(G \cup H) = J$. Suppose Arthur's method is to pick k random graphs in the universe U and ask the prover if any of them are in the orbit of $G \cup H$. The prover is 2^k times more likely to be able to respond affirmatively if the graphs are non-isomorphic. Unfortunately, there are 2^{2^n} graphs on 2n vertices, so the size of the universe is much greater than the size of the orbit, and we need an exponential number of rounds to reach a conclusion.

The solution is to use hashing to reduce the size of the universe. The set $G \cup H$ is a tiny fraction of U. But, if we hash U down to a small set $S = h(U)$, the image of $G \cup H$ will be a much larger fraction of S. We use the following procedure. Merlin first sends an appropriate hash function. Arthur then picks k elements in the image. Merlin sends the preimage in the orbit of $G \cup H$ for as many points as is possible. If A and B are not isomorphic, then the image of $G \cup H$ under the hash function will be a big, and this will often be possible. If G and H are isomorphic, then the image of $G \cup H$ will be only a small fraction of S, and it will be possible less often. It remains to find the hash function.

6 Universal Hashing

We will first consider the behavior of hash functions. Let $T = Orbit(G \cup H)$. Let H be a set of universal hash functions from U to S. Let $\alpha = \frac{|T|}{|S|}$.

Lemma 6 *$\alpha \geq \frac{E[|h(T)|]}{|S|} \geq \alpha - \frac{\alpha^2}{2}$ over all $h \in H$.*

Proof The first inequality is easy, since $\forall h \in H, |h(T)| \leq |T|$. The second is more tricky. Let us pick some $a \in S$. Over the hash functions $h \in H$,

$$\begin{aligned}
\Pr[[]a \in h(T)] &\geq \sum_{x \in T} \Pr[[]a = h(x)] - \sum_{\{x,y\} \in T} \Pr[[]a = h(x) = h(y)] \\
&= \sum_{x \in T} \frac{1}{|S|} - \sum_{\{x,y\} \in T} \frac{1}{|S|^2} \\
&= \frac{|T|}{|S|} - \frac{\binom{|T|}{2}}{|S|^2} \\
&> \alpha - \frac{\alpha^2}{2}
\end{aligned}$$

The first line uses an inclusion-exclusion expansion truncated after two terms. The second and third lines follow from the definition of universal hashing and the linearity of expectations. The final line follows from the definition of α.

■

Note that as α grows greater than 1, the lower bound $\alpha - \frac{\alpha^2}{2}$ begins to drop from 1/2. However, since $E[|h(T)|]/|S|$ must be non-decreasing as the set T grows, 1/2 is still a valid lower bound. We will use this fact.

7 Procedure

Using the lemma, we find a hash function with $S = \{0,1\}^k$ for which $2(2n)! < 2^k < 4(2n)!$. Our procedure is as follows:

Merlin presents a hash function h from the universal family defined previously. He picks h such that $\alpha = \frac{(2n)!}{2^k}$, so that $\frac{1}{2} < \alpha < \frac{1}{4}$.

If the graphs are non-isomorphic, $\exists h$ such that $\frac{|h(T)|}{2^k} \geq \alpha - \frac{\alpha^2}{2} \geq \frac{3}{4}\alpha$.
If the graphs are isomorphic, $\forall h$, $\frac{|h(T)|}{2^k} \leq \frac{\alpha}{2}$.

Arthur chooses k random elements x_1 and x_2 from S and sends them to Merlin.

Merlin finds $a_{i1}, ..., a_{iq}$ in T such that $h(a_{ij}) = x_j$ for as many x_j as possible. Merlin sends these values along with the proofs $\pi_{i1}, ..., \pi_{iq}$, where $\pi_{ij}(G \cup H) = a_{ij}$.

Arthur accepts if Merlin has sent the required information in the correct form, and the above conditions hold, and $\frac{q}{m} > \frac{5\alpha}{8}$. We can see that the is the correct condition by applying a Chernoff bound.

The general case (of problems other than graph non-isomorphism) uses many of the same ideas and hinges on the fact that there are many choices of coin flips that make the prover accept.