CMPT 407/710 - Complexity Theory: Lecture 13

Valentine Kabanets

June 27, 2017

1 Randomized complexity classes

We have the following inclusions: $\mathsf{ZPP} \subseteq \mathsf{RP} \subseteq \mathsf{BPP} \subseteq \mathsf{MA} \subseteq \mathsf{AM} \subseteq \Pi_2^p$. The first three inclusions are fairly easy. Next we'll show the inclusion $\mathsf{MA} \subseteq \mathsf{AM}$, and leave the last inclusion $\mathsf{AM} \subseteq \Pi_2^p$ as an exercise.

The idea for the inclusion $\mathsf{MA} \subseteq \mathsf{AM}$: First, modify the MA protocol to drive the error down, and then simply swap the order of moves of the two parties.

First, we argue the error reduction part.

1.1 Error Reduction in MA

As in the case of BPP, we can reduce the error probability of any MA protocol to be less than an inverse exponential in the input size. Here's how.

Let $L \in \mathsf{MA}$ be any language. Let $R(x, y, z)$ be a polytime relation for L such that, for every $x \in L$, there is a y with $\mathbf{Pr}_z[R(x, y, z) = 1] \geq 3/4$; and for every $x \notin L$, for every y, it holds that $\mathbf{Pr}_z[R(x, y, z) = 1] \leq 1/4$.

Consider a new protocol where, upon receiving a string y, Arthur randomly and independently chooses k strings $z_1, \dots, z_k$, and accepts iff $R(x, y, z_i) = 1$ for more than half of these k strings.

We use Chernoff bounds to analyze the correctness of the described protocol. Suppose first that $x \in L$. Then Merlin can send Arthur a string y such that $\mathbf{Pr}_z[R(x, y, z) = 1] \geq 3/4$. Every string z_1, $1 \leq i \leq k$, randomly chosen by Arthur has probability at least $3/4$ of satisfying R. The expected number of z_i's that satisfy R is $\mu \geq \frac{3}{4}k$. Let X_i, $1 \leq i \leq k$, be a random variable that is 1 if z_i satisfies R, and 0 otherwise. Let $X = \sum_{i=1}^{k} X_i$. As we just argued, the expectation of X is μ. Using Chernoff bounds, we get that $\mathbf{Pr}[X \leq k/2] \leq \mathbf{Pr}[|X - \mu| > k/4] < 2e^{-k/48}$. Thus, Arthur will accept with probability exponentially close to 1.

On the other hand, suppose that $x \notin L$. Then, whatever y is sent to Arthur, $\mathbf{Pr}_z[R(x, y, z) = 1] \leq 1/4$. Let X_i be random variables as defined above, and let $X = \sum_{i=1}^{k} X_i$. Then the expectation of X is $\leq \frac{1}{4}k$. The probability that Arthur accepts in this case is $\mathbf{Pr}[X > k/2]$, which, by Chernoff bounds, is at most $2e^{-k/48}$. Thus, in this case, Arthur will accept with probability exponentially close to 0.

1.2 Proof of $\mathsf{MA} \subseteq \mathsf{AM}$

We show the following.

Theorem 1. $\mathsf{MA} \subseteq \mathsf{AM}$

Proof. Let $L \in \mathsf{MA}$ be arbitrary. Let $R(x, y, z)$ be a polytime relation for L, as in the definition of MA. By the previous section (on error reduction), we may assume that the Merlin-Arthur protocol for L has exponentially small error probability. More precisely, by analyzing the error reduction argument from the last section, we note that the modified protocol does *not* change the size of Merlin's proof string y, but only increases the size of Arthur's random string. Therefore, the error probability of an MA protocol for L can be made exponentially small in the size of Merlin's string y, e.g., $2^{-2|y|}$.

So, let $R(x, y, z)$ be such that, for every x of length n, we have

$$x \in L \Rightarrow \exists y : \ \mathbf{Pr}_z[R(x, y, z) = 1] \geq 1 - 2^{-2|y|},$$
$$x \notin L \Rightarrow \forall y : \ \mathbf{Pr}_z[R(x, y, z) = 1] \leq 2^{-2|y|}.$$

The new, Arthur-Merlin protocol for L will be: For an input x, Arthur sends to Merlin a random string z, and receives from Merlin a string y. Arthur accepts iff $R(x, y, z) = 1$. In other words, we just switch the order of moves in the previous Merlin-Arthur protocol (but only after making sure that the Merlin-Arthur protocol has tiny error probability!)

Let's analyze the error probability of the described Arthur-Merlin protocol. If $x \in L$, then, by the definition of MA, there is a string y such that $R(x, y, z) = 1$ for almost all random z's. So, by answering Arthur's challenge z, with this string y, Merlin is guaranteed to make Arthur accept with probability at least $1 - 2^{-2|y|} \geq 3/4$.

Now suppose that $x \notin L$. Then every possible string y has at most $2^{-2|y|}$ fraction of z's such that $R(x, y, z) = 1$. So, overall, there are at most $2^{|y|}2^{-2|y|} = 2^{-|y|}$ fraction of z's such that $R(x, y, z) = 1$ for some y. In other words, for at most $2^{-|y|} \leq 1/4$ of Arthur's random challenges z, can Merlin answer with a string y that makes Arthur accept. This proves that the described Arthur-Merlin protocol is indeed correct. □

2 Logical view of MA and AM

It is possible to define quantifiers $\exists$, $\forall$, and BP, where $\exists$ corresponds to nondeterministic guessing, $\forall$ to co-nondeterministic guessing, and BP to the "large majority". Then, using this notation, we can express

- $\mathsf{MA} = \exists \circ \mathsf{BP} \circ \mathsf{P}$, and
- $\mathsf{AM} = \mathsf{BP} \circ \exists \circ \mathsf{P}$.

Our proof of $\mathsf{MA} \subseteq \mathsf{AM}$ can be used to show that the part "$\exists \circ \mathsf{BP}$" can be also written as "$\mathsf{BP} \circ \exists$". That is, the quantifier BP may jump over the preceding $\exists$ quantifier.

Our earlier proof that $\mathsf{BPP} \subseteq \Sigma_2^p \cap \Pi_2^p$ can be used to show that the BP quantifier can be also written as either $\exists \circ \forall$ or $\forall \circ \exists$.

As a consequence of these properties of the quantifiers introduced above, we immediately get:

Theorem 2. $\mathsf{AM} \subseteq \Pi_2^p$.

Proof. We have

$$\begin{aligned} \mathsf{AM} &= \mathsf{BP} \circ \exists \circ \mathsf{P} && \text{(by definition of } \mathsf{AM}\text{)} \\ &\subseteq \forall \circ \exists \circ \exists \circ \mathsf{P} && \text{(replacing } \mathsf{BP} \text{ with } \forall \circ \exists\text{)} \\ &= \Pi_2^p. && \text{(merging } \exists \circ \exists \text{ into } \exists\text{)} \end{aligned}$$

□

Theorem 3. $\mathsf{MA} \subseteq \Sigma_2^p \cap \Pi_2^p$.

Proof. We have

$$\begin{aligned} \mathsf{MA} &= \exists \circ \mathsf{BP} \circ \mathsf{P} && \text{(by definition of } \mathsf{MA}\text{)} \\ &\subseteq \exists \circ \exists \circ \forall \circ \mathsf{P} && \text{(replacing } \mathsf{BP} \text{ with } \exists \circ \forall\text{)} \\ &= \Sigma_2^p. && \text{(merging } \exists \circ \exists \text{ into } \exists\text{)} \end{aligned}$$

We also have

$$\begin{aligned} \mathsf{MA} &= \exists \circ \mathsf{BP} \circ \mathsf{P} && \text{(by definition of } \mathsf{MA}\text{)} \\ &\subseteq \mathsf{BP} \circ \exists \circ \mathsf{P} && \text{(}\mathsf{BP} \text{ jumps over the preceding } \exists\text{)} \\ &\subseteq \forall \circ \exists \circ \exists \circ \mathsf{P} && \text{(replacing } \mathsf{BP} \text{ with } \forall \circ \exists\text{)} \\ &= \Pi_2^p. && \text{(merging } \exists \circ \exists \text{ into } \exists\text{)} \end{aligned}$$

□

(Thus, to solve the last problem on HW 3, it suffices to justify the properties of the quantifiers stated above. But you can also solve that HW problem without using the quantifiers.)

3 AM protocol for Graph Non-Isomorphism

Last time we saw a private-randomness protocol for Graph Non-Isomorphism (NISO). We'll show how to make into a public-randomness protocol, using approximate counting (or hashing).

Here's the idea. For two given graphs G_1 and G_2 on n nodes each, define the set $W = \{G' \mid G' \text{ is isomorphic to } G_1 \text{ or } G_2\}$. For simplicity, assume that neither G_1 nor G_2 has any non-trivial automorphisms. This means that if you take G_1 apply all $n!$ permutations to its nodes, you get $n!$ distinct graphs (all isomorphic to G_1, of course).

Thus, if G_1 and G_2 are isomorphic, the size of W is $(n!)$. On the other hand, if G_1 and G_2 are not isomorphic, the size of W is $2(n!)$. In other words, W is either large or small, depending on whether G_1 and G_2 are non-isomorphic.

The AM protocol will try to convince Arthur that the set W is "large". The idea is to use hashing to distinguish between large and small sets. We'll provide the details next.

Let's define $\mathsf{AM}[k]$ to be the class of languages decided by an Arthur-Merlin protocol with at most k rounds of communication. Note that $\mathsf{AM} = \mathsf{AM}[2]$.

From the definition of $\mathsf{IP}[2]$, it seems that $\mathsf{IP}[2]$ is more powerful than AM. It is easy to simulate Arthur-Merlin protocol in $\mathsf{IP}[2]$, but it's not at all clear how to simulate $\mathsf{IP}[2]$ with AM. The surprising result of Goldwasser and Sipser shows that in fact the two classes are the same! More generally, we have:

Theorem 4 (Goldwasser-Sipser)**.** *For any efficiently computable $k : \mathbb{N} \to \mathbb{N}$,*

$$\mathsf{IP}[k] \subseteq \mathsf{AM}[k+2].$$

The proof of this theorem is too involved to be presented here. To get a glimpse of the proof technique used in the proof, we'll construct an AM protocol for the Graph Non-Isomorphism problem,

$$NISO = \{(G_1, G_2) \mid \ G_1 \text{ and } G_2 \text{ are not isomorphic}\}.$$

Theorem 5 (Goldwasser-Sipser)**.** *$NISO \in \mathsf{AM}$*

For the proof, we'll need some terminology. An *automorphism* of a graph G is an isomorphism between G and G. A *trivial automorphism* of G is the identity function. Note that a graph G on n vertices without any non-trivial automorphism has exactly $n!$ distinct isomorphic graphs. (In general, if G has k automorphisms, then G has exactly $n!/k$ distinct isomorphic graphs.)

We'll use universal hash function families. A family $H = H_n$ of hash functions $h : U \to M$ is called universal if it has two properties:

1. (uniformity) for any $u \in U$ and any $a \in M$, $\mathbf{Pr}_h[h(u) = a] = 1/|M|$,
2. (pairwise independence) for any u, u', with $u \neq u'$, and for any $a \in M$, $\mathbf{Pr}_h[h(u) = a \wedge h(u') = a] = 1/|M|^2$.

We assume that our family of hash functions is efficient in the sense that each h can be evaluated efficiently, and we can efficiently sample a random h from the family H. (Such hash function families are known to exist. We'll see an example later.)

Proof of Theorem 5. With loss of generality, assume that given input graphs (G_1, G_2) have no non-trivial automorphisms.

Define the set $W = \{G' \mid \ G' \text{ is isomorphic to } G_1 \text{ or to } G_2\}$. Observe that, if G_1 and G_2 are isomorphic, then $|W| = n!$. If G_1 and G_2 are not isomorphic, then $|W| = 2(n!)$.

Define $Y = W \times W$ be the cross-product of W with itself. Then, for isomorphic G_1 and G_2, $|Y| = (n!)^2$, whereas for non-isomorphic G_1 and G_2, $|Y| = 4(n!)^2$.

Let $M = \{0, 1, 2, \dots, 4(n!)^2 - 1\}$ be a set of size $4(n!)^2$. Let H be a family of universal hash functions $h : U \to M$, where $Y \subset U$ (i.e., U is a set of binary strings that contains binary encodings of all elements in Y.)

Arthur will use a randomly chosen hash function to test if Y is large or small. More formally, Arthur picks a random hash function h and sends it to Merlin. Merlin sends back a string $y \in Y$ with a proof that $y \in Y$ (note that such a proof is short: it's just a pair of isomorphisms). Arthur checks that Merlin's proof of $y \in Y$ is correct, and he checks that $h(y) = 0$. If the checks pass, then Arthur accepts; otherwise, he rejects.

Analysis In case G_1 and G_2 are isomorphic, we have $|Y| = (n!)^2 = |M|/4$, and so

$$\begin{aligned}\mathbf{Pr}_h[\exists y \in Y :\ h(y) = 0] &\leq \sum_{y \in Y} \mathbf{Pr}_h[h(y) = 0] \\ &= |Y|/|M| = 1/4,\end{aligned}$$

where the first inequality is by the "union bound", and the second equality by uniformity property of our hash function family.

In case G_1 and G_2 are non-isomorphic, we have $|Y| = 4(n!)^2 = |M|$. So,

$$\begin{aligned}\mathbf{Pr}_h[0 \in h(Y)] &= \mathbf{Pr}_h[h(y_1) = 0 \vee h(y_2) = 0 \vee \cdots \vee h(y_{|Y|}) = 0] \\ &\geq \sum_{y \in Y} \mathbf{Pr}_h[h(y) = 0] - \sum_{\{y,z\} \in Y} \mathbf{Pr}_h[h(y) = 0 \wedge h(z) = 0] \\ &= |Y|/|M| - |Y| * (|Y| - 1)/(2|M|^2) \\ &\geq 1 - 1/2 = 1/2,\end{aligned}$$

where the first inequality is by the Inclusion-Exclusion Principle, and the second equality uses uniformity and pairwise independence of our family of hash functions.

So, if G_1 and G_2 are not isomorphic, Arthur can be made to accept with probability at least 1/2. If, on the other hand, G_1 and G_2 are isomorphic, Arthur accepts with probability at most 1/4. (Correctness probability can be amplified by repeating the protocol several times in parallel.) So, we have an AM protocol for NISO.

It remains to show what to do in the case the input graphs have non-trivial automorphisms. Let us re-define

$$W = \{(G', \pi) \mid\ G' \text{ is isomorphic to } G_1 \text{ or } G_2, \text{ and } \pi \text{ is an automorphism of } G'\}.$$

It is left as exercise to check that, for isomorphic G_1 and G_2, $|W| = n!$; and for non-isomorphic G_1 and G_2, $|W| = 2(n!)$. The rest of the proof is the same as before. □

3.1 Hash functions

There are many constructions of universal hash function families. We'll mention only one here. Let $U = \{0,1\}^n$ and $M = \{0,1\}^k$. We'll define a random hash function h_r from U to M as follows: Pick a random 0/1 matrix A of dimensions k by n (i.e., k rows and n columns), and a random 0/1 column-vector b of dimension k. (Set $r = A; b$.) For any vector $x \in \{0,1\}^n$, define $h_{A,b}(x) = Ax + b$ where all arithmetic is done modulo 2.

It is left as an exercise to show the family $\{h_{A,b}\}$ over random A, b is indeed a universal hash family.

Note that each hash function in the family is described with "few" random bits of $O(kn)$, and each hash function can be efficiently evaluated at any given input x. Such hash functions can be used in the NISO protocol above.

Remark 6. *Actually, there is some technicality: the size of the set M in the protocol above is set $4(n!)^2$, where n is the number of vertices of the graph. For the hash family defined above, the size of the image set M must be a power of 2, i.e., of the size 2^k for some k. It may happen that $4(n!)^2$ is not a power of two. So, in this case, we choose k so that $2^{k-2} < 4(n!)^2 \leq 2^{k-1}$ (which is always possible), and use hash functions from U to $\{0,1\}^k$. The same argument as before will apply, and we'll get that Arthur accepts on non-isomorphic graphs with probability at least $3/16$, and that Arthur accept on isomorphic graphs with probability at most $1/8 < 3/16$. (We leave it as an exercise to verify these probability bounds.)*