CMPT 407: Computational Complexity **Summer 2017**

Lecture 7: Randomized Computation (26 June - End)

Lecturer: Valentine Kabanets *Scribe: Lily Li*

7.1 Randomness and Interaction

Today we will show that $\mathsf{BPP} \subseteq \mathsf{MA} \subseteq \mathsf{AM} \subseteq \prod_2^p$. Remark that the first $\subseteq$ is actually quite straight forward. Please consider what BPP and MA are and explain why (*hint: remember that* MA *is essentially the randomized version of* NP). Remark the verifier Arthur checks $R(x, y, z)$ where x is the input, y is the value the Merlin sends over, and z is the random string.

As before, we can decrease the error probability by choosing k different random strings $z_1, ..., z_k$ and taking the majority of $R(x, y, z_i)$ for all z_i. Let us do the analysis (use Chernoff bound):

1. $x \in L$: then $\exists y, s.t. \forall i \leq i \leq k : \mathbf{Pr}_{z_i}[R(x, y, z_i) = 1] \geq 3/4$. Let the indicator random variable be $X_1, ..., X_k$ where $X_i = R(x, y, z_i)$. Note that $\mu = k \cdot 3/4$. Calculate the probability that fewer than $k/2$ variables accept by letting $X_1 + \cdots + X_n = X$:

$$\begin{aligned}\mathbf{Pr}[X < \frac{k}{2}] &\leq \mathbf{Pr}[X \leq \frac{2}{3} \cdot \mu] \\ &\leq \mathbf{Pr}[|X - \mu| \geq \frac{\mu}{3}] \\ &\leq 2e^{\frac{\mu}{-27}}\end{aligned}$$

   where the last inequality follows from the Chernoff bound as discussed below. Thus this probability is quite low, $\propto 2^{-k}$.

2. $x \notin L$: then $\forall y \forall 1 \leq i \leq k, \mathbf{Pr}_{z_i}[R(x, y, z_i) = 1] \leq \frac{1}{4}$. By a similar application of Chernoff bounds we can show that $\mathbf{Pr}[X > k/2] \leq \mathbf{Pr}[|X - \mu| > k/4] \leq 2e^{k/48}$.

Note: what does the Chernoff bound actually say? It says for independent random variable $X_1, ..., X_n \in \{0, 1\}$ s.t. $\mathbf{Pr}[X_i = 1] = p$ and $\mathbf{Pr}[X_i = 0] = 1 - p$ for some $0 \leq p \leq 1$. If we let $X = \sum_{i=1}^n X_i$, then $\mathsf{Exp}[X] = n \cdot p = \mu$ and $\forall 0 < \epsilon < 1$,

$$\mathbf{Pr}[|X - \mu| > \epsilon \cdot \mu] < 2 \cdot e^{-\epsilon^2 \cdot \mu/3}$$

In our original protocol, Arthur uses m bits of randomness and gets reasonable error. However, if we use the above modification then we can reduced the error to $\propto 2^{-k}$ by using $k \cdot m$ bits of randomness.

Theorem 7.1 $\mathsf{MA} \subseteq \mathsf{AM}$.

Proof: Let us think about this intuitively: in AM, Merlin is allowed more opportunities to cheat since Merlin sees Arthur's random string. Let us think about a language $L \in \mathsf{MA}$ and reduce this to a language in AM. What does it mean for L to be in MA? It means when given some x as input, Merlin sends Arthur

a certificate y. If $x \in L$ then for any random string z, $\mathbf{Pr}[R(x,y,z)=1] \geq 3/4$. And if $x \notin L$ then for any random string z, $\mathbf{Pr}[R(x,y,z)=1] \leq 1/4$. Now we need to turn this into a language in AM, that is Arthur must reveal his random string z to Merlin first before Merlin sends over y.

If you just switched the order of Arthur and Merlin, then with high probability, there exists a certificate y such that given a random string z, $R(x,y,z)=1$. Thus the trick is to reduce the likelyhood that $R(x,y,z)=1$ in general. Let $|y|=m$. We will reduce the error probability to $\leq 1/2^{2m}$. Observe that if $x \in L$ then Merlin can still give a valid y. However suppose $x \notin L$ and Merlin attempted to cheat:

$$\begin{aligned}\mathbf{Pr}_z[\exists y : R(x,y,z)=1] &\leq \sum_y \mathbf{Pr}_z[R(x,y,z)=1] \\ &= 2^m \cdot 2^{-2m} = 2^{-m}\end{aligned}$$

■

7.1.1 Logic of MA and AM

Consider the quantifiers

$$\begin{aligned}\exists \quad & \exists \circ \mathsf{P} = \mathsf{NP} \\ \forall \quad & \forall \circ \mathsf{P} = \mathsf{coNP} \\ \mathsf{Maj}_{2/3} \quad & \mathsf{BP} \circ \mathsf{P} = \mathsf{BPP}\end{aligned}$$

Let us use these as follows (this will be a very informal perusal of this topic). Consider AM: here the majority machine goes first then a non-deterministic machine goes. This can be translated into $\mathsf{BP} \circ \exists \circ \mathsf{P}$. Similarly for MA, we can write $\exists \circ \mathsf{BP} \circ \mathsf{P}$.

Claim 7.2 $\exists \circ \mathsf{BP} \subseteq \mathsf{BP} \circ \exists$

Proof: A moment's though and you will realize that this claim is more or less equivalent to the above theorem that $MA \subseteq AM$. ■

Claim 7.3 $\mathsf{BP} \subseteq \exists \circ \forall \circ \mathsf{P}$ *and* $\mathsf{BP} \subseteq \forall \circ \exists \circ \mathsf{P}$.

Proof: Again this is quite similar to something we have done before, namely $\mathsf{BPP} \subseteq \sum_2^p \cap \prod_2^p$. ■

Using the above conversions and the above laws we can make quick work of the following theorems.

Theorem 7.4 $\mathsf{AM} \subseteq \prod_2^p$.

Proof: First we translate this statement using the quantifier analogies to get:

$$\mathsf{BP} \circ \exists \circ \mathsf{P} \subseteq \forall \circ \exists \circ \mathsf{P}.$$

Then using the above laws:

$$\begin{aligned}\mathsf{BP} \circ \exists \circ \mathsf{P} &\subseteq \forall \circ \exists \circ \exists \circ \mathsf{P} \\ &\subseteq \forall \circ \exists \circ \mathsf{P}\end{aligned}$$

■

Theorem 7.5 $\mathsf{MA} \subseteq \prod_2^p \cap \sum_2^p$.

Proof: We will only show $\mathsf{MA} \subseteq \prod_2^p$ here, $\mathsf{MA} \subseteq \sum_2^p$ is quite similar.

$$\exists \circ \mathsf{BP} \circ \mathsf{P} \subseteq \mathsf{BP} \circ \exists \circ \mathsf{P} \subseteq \forall \circ \exists \circ \exists \circ \mathsf{P} \subseteq \forall \circ \exists \circ \mathsf{P}$$

■

7.2 Graph Non-isomorphism $\in$ AM

Recall the Graph Non-isomorphism (NISO). First it is important to remark that the Graph Non-isomorphism problem is a private key protocol, that is the randomness chosen by Arthur is not reveal to Merlin. However we will show that this can be made into a public key protocol ($\in$ AM).

Note: (in terms of graphs) there is a difference between an automorphism and an isomorphism.

Note: (regarding hash functions) $H = \{h\}$ where $h : U \to M$, for universe U and $M \subset U$, is a random function family if

1. Uniform: $\forall u \in U, \forall a \in M, \mathbf{Pr}_h[h(u) = a] = 1/|M|$.
2. 2-wise Independent: $\forall u, u' \in U, u \neq u', \mathbf{Pr}_h[h(u) = h(u') = a] = 1/|M|^2$.

Lets consider an example hash function.

Example 7.6 *Let the universe be $U = \{0,1\}^n$ and the range be $M = \{0,1\}^k$. We create a hash function by picking a random 0-1 matrix A of dimension $k \times n$ and a random 0-1 vector b of dimension k. Given an input vector x,*

$$h_{A,b}(x) = A \cdot x + b(\mod 2)$$

Theorem 7.7 NISO $\in$ AM.

Proof: We will first solve this problem with certain assumption (not true). Suppose that we have two graphs G_1 and G_2 which do not have nontrivial automorphisms. Construct the set $W = \{\}$ ■

Theorem 7.8 $\forall k \geq 2$

Proof:

■ *Remark:* k needs to be a constant. Every time you move BP in-front of $\exists$, the predicate increases by a polynomial amount.

It is conjectured that BPP = P and AM = MA = NP. That is: randomness does not give you any more power than just solving the de-randomized problem.

What evidence do we have for thinking this way? Consider the following theorems involving circuit complexity.

Theorem 7.9

It is also unknown if SAT*then* requires $2^{\Omega(n)}$ circuit complexity. If this is shown to be true, then we have BPP = P. SAT*then* is an unusually tricky problem. Our best algorithm has exponential running time as is our circuit size.

7.3 Polynomial Identity Test

(The last remaining problem which is in BPP and unknown if it is in P or not). The input are arithmetic formulas $f(x_1, ..., x_n)$ and $g(x_1, ..., x_n)$. We want to determine if $f \cong g$. You should remember this theorem from both the previous complexity and linear algebra courses. Hint: we use Schwartz-Zippel Lemma.

Claim 7.10 *Arithmetic formula of size $\leq A$ computes a polynomial of degree $\leq A$ where the size of an arithmetic formula is the number of leaves in its corresponding tree.*

Proof: By induction on A (this is actually quite straight forward). The base case of one variable can be computed by a degree one polynomial. Suppose the claim holds for $n = A > 1$. Show that the claim holds for $n = A + 1$. Consider any arithmetic formula with size n. This formula is of the form $F_1 \oplus F_2$ where F_1 and F_2 are child formulas of an arithmetic operator $\oplus \in \{\cdot, +\}$. ■

Next we want to consider equality for two formulas given by circuits of size A. The standard technique turns out to not work since the degree of the arithmetic formula associated with the circuit could be quite large. The trick is: modular arithmetics.

Lemma 7.11 *3-CNF $\phi(x_1, ..., x_n)$ with m clauses can be turned into a polynomial of degree $\leq 3m$.*

Proof: We can introduce a polynomial for each elementary boolean statement. Notably: $\neg x$ is simply $(1 - x)$, $p_1 \vee p_2$ is ■

Theorem 7.12 *Multi-round #SAT then.*

Proof: You should remember this theorem from before. Let $f(x)$ be the input 3-CNF. Both the prover and verifier calculate the associated boolean polynomial $\phi_f(x)$. The prover sends the verifier the value s which the prover claims is number of satisfying assignments for f. The verifier needs to send a set of challenges to the prover to see if he is lying. Remark that $\phi_f(x) = \sum_{x_1=0}^{1} \cdots \sum_{x_n=1}^{1} f(x_1, ..., x_n)$. The verifier creates the polynomial $f_1(r_1) = \sum_{x_2=0}^{1} \cdots \sum_{x_n=0}^{1} f(r_1, x_2, ..., x_n)$ and choses a random r_1 in some large range $1, 2, ..., 2^n$. ■

Theorem 7.13 IP = PSPACE

Proof: In one direction IP $\subseteq$ PSPACE is quite simple.

In the other direction PSPACE $\subseteq$ IP we need a PSPACE complete problem problem. Enter QBF : $\forall x_1 \exists x_2 \cdots$ ■

Multiple interactive proof. property testing???

7.4 Probabilistically Checkable Proof

Theorem 7.14 *(PCP Theorem) Every language $L \in$ NP has a verifier V such that*

- *V is a randomized polytime algorithm,*

- *V reads only a constant number of symbols in a give proof such that fer every $x \in \{0,1\}^n$: if $x \in L$, then there is a proof ϕ such that $V^\pi(x)$ accepts with probability at least $2/3$, and if $x \notin L$, every candidate proof π is such that V^π rejects with probability at least $2/3$.*

Note there that V^π means that the verifier V has random access to the proof of π and can read constantly many symbol of the proof.