CS221: Computational Complexity Prof. Salil Vadhan

Lecture 17: Randomized Complexity and Identity Testing

10/30 *Scribe: Grant Schoenebeck*

Contents

1 Identity Testing

Given algebraic expressions $p(x_1, x_2, \ldots, x_n), q(x_1, x_2, \ldots, x_n)$ decide is $p \equiv q$?

We can interpret equivalence in two different ways: as formal polynomials or as functions from $\mathbb{Z}^n \to \mathbb{Z}$. For polynomials over $\mathbb{Z}$, these two interpretations are the same.

We gave the following randomized algorithm to solve the problem:

1. Let $m = \max\{|p|, |q|\}, M = 2^m, S = \{1, \ldots, M\}$.
2. Randomly choose $\alpha_1, \alpha_2, \ldots, \alpha_n \Leftarrow S$.
3. Accept if $p(x_1, x_2, \ldots, x_n) = q(x_1, x_2, \ldots, x_n)$.

Proposition 1

if $p \equiv q$ then algorithm always accepts.

if $p \not\equiv q$ then algorithm accepts with probability $\leq \frac{m}{M} = 2^{-\Omega(m)}$ (over its coin tosses).

1) is obvious.

2) follows from the 2 lemmas below.

Definition 2 *The* total degree *of a polynomial is the maximum of the sums of coefficients in each term (when written in canonical form* $\sum_{i_1,\ldots,i_n \in \mathbb{N}} c_{i_1,\ldots,i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$*)*

Example 3 The total degree of $f(x, y) = x^2y^3 + x^4$ is 5.

Lemma 4 *The total degree of expression* $p(x_1, x_2, \ldots, x_n)$ *is at most* $|p|$*.*

Proof:

(by induction on $|p|$)

$\deg(p_1 \cdot p_2) = \deg(p_1) + \deg(p_2) \leq$ (by inductive hypothesis)$|p_1| + |p_2| \leq |p_1 \cdot p_2|$. A similar argument shows that $\deg(p_1 + p_2) \leq |p_1 + p_2|$. ■

Lemma 5 (Schwartz–Zippel) *If* r *is a nonzero polynomial,*

$$\Pr_{(\alpha_1,\alpha_2,\ldots,\alpha_n)\leftarrow S} [r(\alpha_1, \alpha_2, \ldots, \alpha_n) = 0] \leq \frac{\deg r}{|S|}$$

where $\deg r$ *is the total degree of* r*.*

Proof: (By induction on n, the number of variables)

Base case: n = 1. $\Pr_{\alpha\leftarrow S} [r(\alpha) = 0] \leq \frac{\deg r}{|S|}$. This follows from a fundamental theorem of algebra that states the the number of solutions to a non-zero polynomial is at most the degree of the polynomial.

Inductive step. Observe that

$$r(x_1, \ldots, x_n) = \sum_{i=0}^{k} r_i(x_1, x_2, \ldots, x_{n-1}) x_n^i, \tag{1}$$

where $r_k \neq 0$. Then

$$\begin{aligned}
&\Pr[r(\bar{\alpha}) = 0] \\
&\quad\leq \Pr[r_k(\alpha_1, \alpha_2, \ldots, \alpha_{n-1}) = 0] + \Pr[r(\alpha_1, \alpha_2, \ldots, \alpha_n) = 0 | r_k(\alpha_1, \alpha_2, \ldots, \alpha_{n-1}) \neq 0] \\
&\quad\leq \frac{\deg r_k}{|S|} + \frac{k}{|S|}
\end{aligned}$$

The first term follows from the inductive hypothesis, while the second term is a result of the same algebraic property we used in the base case. Note now that $\deg r_k + k \leq \deg r$, by Equation (1). ■

With these two lemmas, to prove proposition just take $r = p - q$.

Remark 6 *This works over any field or integral domain (because those have property that a polynomial of degree d has at most d roots).*

Definition 7 *An* integral domain *is a set with addition (+) and multiplication ($\cdot$) such that $a \cdot b = 0 \Leftrightarrow [a = 0$ or $b = 0]$.*

Example 8 For example the integers modulo a prime p. $a \cdot b \equiv 0 \pmod{p} \Leftrightarrow p | a \cdot b \Leftrightarrow p | a$ or $p | b \Leftrightarrow a \equiv 0 \pmod{p}$ or $b \equiv 0 \pmod{p}$.

Note this property doesn't hold modulo a composite: $2 \cdot 3 \equiv 0 \pmod{6}$.

Remark 9 *The algorithm only requires that we are given a polynomial in a form such that:*

- *can bound the degree of the polynomial (even exponentially is fine).*
- *can evaluate the polynomial in polynomial time.*

Example 10 We can test identities like Vandemonde:

$$\det \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \vdots & \vdots & \cdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1}) \end{bmatrix} = \pm \prod_{i<j} (x_i - x_j)$$

note: we can bound the degree and can evaluate it in polynomial time.

2 Randomized Complexity Classes

Definition 11 *A* probabilistic TM *is a TM M with a special "coin flip" state q_{flip} such that when M goes into q_{flip} the current cell is replaced randomly with a 0 or 1.*

Alternatively: we can start the TM with an extra "random tape" filled with random bits.

Definition 12 (Random Polynomial Time) $L \in$ **RP** *if $\exists$ a probabilistic polynomial-time TM M such that*

1. *If* $x \in L \Rightarrow \Pr_{\textit{coin flips}}[M(x) \textit{ accepts }] \geq \frac{1}{2}$.
2. *If* $x \notin L \Rightarrow \Pr_{\textit{coin flips}}[M(x) \textit{ accepts }] = 0$

Thus, the algorithm can make errors only on YES instances. We stress that the above holds for *all* inputs; the randomness is only over the algorithm's coin tosses. We have shown:

Theorem 13 *IdentityTest is in* **co-RP**.

Lemma 14 (RP Amplification) *If* $\mathbf{L} \in \mathbf{RP} \Rightarrow \forall$ *polynomial* p $\exists$ *an* **RP** *algorithm for* L *with error probability* $\leq 2^{-p(n)}$

Proof: Given error $\frac{1}{2}$ algorithm M for L, run $M(x)$ $p(n)$ times and accept if M ever accepts, otherwise reject.

The probability that M accepts if $x \notin L$ is still 0. But if $x \in L$ the probability that it rejects is at most $(\frac{1}{2})^{p(n)}$. ■

What about 2-sided error?

Definition 15 (Bounded-error probabilistic polynomial time) $L \in \mathbf{BPP}$ *if* $\exists$ *a polynomial time probabilistic TM* M *such that*

1. *If* $x \in L \Rightarrow \Pr_{\textit{coin flips}}[M(x) \textit{ accepts }] \geq \frac{2}{3}$.
2. *If* $x \notin L \Rightarrow \Pr_{\textit{coin flips}}[M(x) \textit{ accepts }] \leq \frac{1}{3}$.

Lemma 16 (BPP Amplification) *If* $L \in \mathbf{BPP} \Rightarrow \forall$ *polynomial* $p \exists$ *a probabilistic polynomial-time algorithm for* L *with error probability* $\leq 2^{-p(n)}$

Proof: Given a **BPP** algorithm M and input x, run M on x k times and decide by majority vote.

Fact: The error decreases as $2^{-\Omega(k)}$. This is proved using a fact from probability known as the Chernoff Bound (Lemma 11.9 in Pap.). So by picking an appropriate k (polynomial in the length of x) we get the desired bound. ■

Does **BPP** = **P**? We don't yet know but there is a lot of evidence that if they are not equal they are very close.

We do know that **RP** $\subseteq$ **NP** because the certificate could just be the flips that give you an accepting computation. For the same reason **co-RP** $\subseteq$ **co-NP**. We will prove that **BPP** $\subset$ $\mathbf{\Sigma_2 P}$ next time.