Lecture 8

Lecturer: Dan Spielman *Scribe: Jon Feldman*

This lecture concerns randomized complexity classes. We begin by looking at the problem of finding polynomial identities in P/*poly*, which leads to an interesting open question. We will then give the definitions of the two major randomized complexity classes, RP and BPP. Finally we prove BPP $\subseteq$ P/*poly* using probability amplification.

1 Finding Polynomial Identities in co-RP and in P/*poly*

Zero polynomials are ones in which all coefficients are zero. An algorithm to test whether an input polynomial is zero would be a useful one. In fact you could use such an algorithm to test if two polynomials are equal by just subtracting them and testing if the result is zero.

We will give such an algorithm, but first we need a fact about polynomials that will be useful: if you evaluate a non-zero polynomial at a random point, you usually don't get zero.

1.1 Schwartz/Zippel Lemma

Let P be a non-zero polynomial of degree d with m variables. Choose the values of the variables $x_1, ..., x_m$ at random from some large set S.

$$\Pr[P(x_1...x_m) = 0] \leq \frac{md}{|S|}$$

1.2 The Language POLY-IDENT

Let's define the language for which we want to test membership:

$$\text{POLY-IDENT} = \{\langle P(y_1, ..., y_k), q_1(x_1, ..., x_m), ..., q_k(x_1, ..., x_m)\rangle \mid P(q_1, ..., q_k) = 0\}$$

The Schwartz-Zippel lemma suggests an obvious algorithm to test membership in POLY-IDENT: evaluate the input polynomial on a bunch of random points from a huge field, and if it ever evaluates to something other than zero, reject, otherwise accept. This algorithm has *one-sided error*, since it will always accept if the input is in the language, and it will usually reject if the input is not in the language. So, POLY-IDENT $\in$ co-RP, which will make sense once we define RP later in the lecture. First, let's see why POLY-IDENT $\in$ P/*poly*.

Theorem 1 *POLY-IDENT $\in$ P/poly*

Proof: To show POLY-IDENT $\in$ P/*poly*, we need to define an advice sequnce such that for each polynomial of a certain length, the advice string for that length will let a mchine decide deterministically if the polynomial is zero. We will use a point set as our advice string for each length, so that we can deterministically evaluate the polynomial at those points and accept if we get zero.

So, if we show the following, an advice sequence must exist, and the theorem follows:

- $\forall$ polynomials W s.t. $|W| = n$,
- $\exists X = (x_1...x_m), x_i \in \{1, ..., 5^n\}$ s.t.
- $W \in$POLY-IDENT $\iff W(x_1, ..., x_m) = 0$

To show that such a point set exists, we will use the probabilistic method. This is a way to show the existence of an object by showing that the probability of choosing it randomly is greater than zero. This is a strange concept to get used to, since one usually thinks of it the other way around. But it turns out to be a very clean and concise way to prove many things that would otherwise be very difficult to prove.

So, for a given n, we choose $x_1, ..., x_m$ at random. The event $\mathcal{E}$ we want to occur is:

$$\mathcal{E}\colon \forall W \text{ s.t. } |W| = n,\ W \in \text{POLY-IDENT} \iff W(x_1, ..., x_m) = 0$$

If we show that the probability over all choices of X that $\mathcal{E}$ occurs is greater than zero, then an X must exist that will serve as our advice string. We will do this by working with $\overline{\mathcal{E}}$:

$$\overline{\mathcal{E}}\colon \exists W \text{ s.t. } |W| = n,\ W \notin \text{POLY-IDENT and } W(x_1, ..., x_m) = 0$$

If we show $\underset{X}{Pr}[\overline{\mathcal{E}}] < 1$ then our existence proof is complete. Using a union bound, where the probability of a union of events is less than or equal to the sum of the probabilities of the individual events, we have:

$$\underset{X}{Pr}[\overline{\mathcal{E}}] \leq \sum_{\substack{W \text{s.t.} |W| = n \\ W \notin \text{POLY} - \text{IDENT}}} \underset{X}{Pr}[W(X) = 0]$$

We can bound the number of variables in W by n since we can only write down at most n variables. We can bound the degree of W by 2^n, since we could never write down a greater degree. There are at most 2^n choices for W. So, by Schwartz/Zippel, we have:

$$\underset{X}{Pr}[\overline{\mathcal{E}}] \leq (2^n)\left(\frac{(n)(2^n)}{5^n}\right) < 1$$

□

An interesting open question is whether there is a concrete way to find such a valid set of points in polynomial time.

2 Randomized Complexity Classes

A randomized poly-time TM is a TM that runs in polynomial time and can flip coins to make decisions. The class BPP corresponds to languages that we can decide in polynomial time with a reasonable probability of coming out with the right answer.

Definition 2 *A language L is in the class **BPP** if $\exists$ a randomized poly-time TM M s.t.*

- $x \in L \implies \Pr[M(x) \text{ accepts}] > \frac{2}{3}$
- $x \notin L \implies \Pr[M(x) \text{ accepts}] < \frac{1}{3}$

Let's be a bit more formal just for fun:

Definition 3 *A language L is in the class **BPP** if $\exists\ A \in P$, $f(n) \leq n^{\mathcal{O}(1)}$, s.t.*

- $x \in L \implies \underset{r \text{ s.t. } |r| = f(|x|)}{\Pr}[\langle x, r\rangle \in A] > \frac{2}{3}$

- $x \notin L \implies \Pr_{r \text{ s.t. } |r|=f(|x|)}[\langle x, r\rangle \in A] < \frac{1}{3}$

The class RP will be a bit stronger. Algorithms in RP have the advantage of only one-sided error; if the string is in the language, we still have a good probability of accepting. But if the string isn't in the language, we will always reject.

Definition 4 *A language L is in the class **RP** if $\exists$ a randomized poly-time TM M s.t.*

- $x \in L \implies \Pr[M(x) \text{ accepts}] > \frac{2}{3}$
- $x \notin L \implies \Pr[M(x) \text{ accepts}] = 0$

If a TM has one-sided error on the other side, the lagnuage of the TM is in co-RP, as is POLY-IDENT.

2.1 The Chernoff Bound

The **Chernoff Bound** is very useful for showing that the sum of independent identically distributed random variables does not drift far from its expectation:

Let $(y_1, ..., y_n)$ be a set of individual identically distributed random variables in $[0, 1]$. Let $\mu = E[Y]$, where $Y = \sum_{i=1}^{n} y_i$. For any ϵ between 0 and 1,

- $\Pr[Y < (1-\epsilon)\mu\ ] < e^{-\frac{\mu\epsilon^2}{3}}$ and
- $\Pr[Y > (1+\epsilon)\mu\ ] < e^{-\frac{\mu\epsilon^2}{3}}$

2.2 Probability Amplification

We will use the Chernoff Bound to show that even if you have a TM that has a very small gap between the probabilities on either side of it's error, you can make another TM that will "blow up" the chance of successfully accepting or rejecting by running the original TM repeatedly and honing in on its success rate.

So, let's redefine BPP one more time, and formalize this theorem:

Theorem 5 *Let L be a language s.t. $\exists$ a randomized poly-time TM M, functions $c(n)$, $s(n)$, $a > 0$ s.t.*

- $x \in L \implies \Pr[M(x) \text{ accepts}] \geq c(|x|)$
- $x \notin L \implies \Pr[M(x) \text{ accepts}] \leq s(|x|)$
- $c(n) - s(n) > \frac{1}{n^a}$

Then, $\forall b > 0$, $\exists$ a randomized poly-time TM M' s.t.

- $x \in L \implies \Pr[M'(x) \text{ accepts}] \geq 1 - 2^{-|x|^b}$
- $x \notin L \implies \Pr[M'(x) \text{ accepts}] \leq 2^{-|x|^b}$

Proof:
M' does the following on input x:

1. Run M k times, where $k = 12|w|^{3a+b}$.
2. Accept if M accepts more than $\left(\frac{c(|w|)+s(|w|)}{2}\right) k$ times.

Let $(y_1, \ldots, y_k)$ be indicator variables, where $y_i = 1$ if M' accepts on its i^{th} run. Let $\mu = E[Y]$, where $Y = \sum_{i=1}^{k} y_i$. We want to express the event that our algorithm fails when $w \in L$ in terms of the deviation of Y from the mean. If $w \in L, \mu \geq c(|w|)k$. If M' rejects, $Y \leq \left(\frac{c(|w|)+s(|w|)}{2}\right) k$. Set $\epsilon = \frac{1}{2} - \frac{k\ s(w)}{2\mu}$, so $Y \leq (1-\epsilon)\mu$. By the Chernoff bound,

$\Pr[M' \text{ rejects } x] = \Pr[Y \leq (1-\epsilon)\mu] < e^{-\frac{\mu\epsilon^2}{3}} \leq e^{-|x|^b} \leq 2^{-|x|^b}$

A symmetric argument shows the desired result for the case where $x \notin L$. $\square$

2.3 Randomness is Subsumed by Non-Uniformity

A corrolary to the amplification theorem we just proved is BPP $\subseteq$ P/*poly*. We prove this by taking advantage of the fact that the probability of making an error using an amplified BPP TM is very small; so small, in fact, that we can union bound over all the possible inputs of a given length and know that there is a random string for that length that will never make an error. We can use the random strings of each length as an advice sequence.

We first look at a reinterpretation of the amplification theorem, then we apply some simple probability to complete the existence proof of such a random string.

Theorem 6 *BPP $\subseteq$ P/poly*

Proof:
Let L be a language in BPP. By the amplification theorem, $\exists$ function $f(n) = n^{\mathcal{O}(1)}$, $A \in P$ s.t.

- $x \in L \implies \Pr_{r \text{ s.t. } |r|=f(|x|)}[\langle x, r\rangle \in A] > 1 - 2^{-|x|-2}$
- $x \notin L \implies \Pr_{r \text{ s.t. } |r|=f(|x|)}[\langle x, r\rangle \in A] < 2^{-|x|-2}$

We construct a sequence of strings r for each arbitrary length n. We prove the existence of a good string r for length n by choosing r at random of length $f(n)$.

$$\begin{aligned}
\Pr[\forall x \text{ s.t. } |x| = n, x \in L, \langle x, r\rangle \in A] &= 1 - \Pr[\exists x \text{ s.t. } |x| = n, x \in L, \langle x, r\rangle \notin A] \\
&\geq 1 - \sum_{x \in L, |x|=n} \Pr[r\langle x, r\rangle \notin A] \\
&\geq 1 - 2^n 2^{-n-2} \geq \frac{3}{4}
\end{aligned}$$

$$\begin{aligned}
\Pr[\forall x \text{ s.t. } |x| = n, x \notin L, \langle x, r\rangle \notin A] &= 1 - \Pr[\exists x \text{ s.t. } |x| = n, x \notin L, \langle x, r\rangle \in A] \\
&\geq 1 - \sum_{x \notin L, |x|=n} \Pr[\langle x, r\rangle \in A] \\
&\geq 1 - 2^n 2^{-n-2} \geq \frac{3}{4}.
\end{aligned}$$

Therefore, $\exists r$ s.t.

$$\forall x \text{ s.t. } |x| = n, x \in L \iff \langle x, r \rangle \in A$$

Using r as an advice string for length n, the theorem follows. $\square$