CS 6810: Theory of Computing **Fall 2021**

Lecture 13: Oct 7, 2021

Lecturer: Eshan Chattopadhyay *Scribe: Atul Ganju*

1 Introduction

In computational complexity theorem, there exists a family of randomized complexity classes for which decision problems are taken to be solved by a deterministic Turing Machine given a polynomial number of random coin flips in the size of its input. Recall that we have discussed two such complexity classes, **BPP** and **RP** defined below:

Definition 1.1 (**BPP**). *A language L is in* **BPP** *if and only if there exists a deterministic Turing machine M that runs in polynomial time in the length of its input for all inputs, such that:*

- $x \in L \iff \Pr_{r \sim \{0,1\}^{p(|x|)}}[M(x,r) = 1] \geq 2/3$
- $x \notin L \iff \Pr_{r \sim \{0,1\}^{p(|x|)}}[M(x,r) = 1] < 1/3$

Definition 1.2 (**RP**). *A language L is in* **RP** *if and only if there exists a deterministic Turing machine M that runs in polynomial time in the length of its input for all inputs, such that:*

- $x \in L \iff \Pr_{r \sim \{0,1\}^{p(|x|)}}[M(x,r) = 1] \geq 1/2$
- $x \notin L \iff \Pr_{r \sim \{0,1\}^{p(|x|)}}[M(x,r) = 1] = 0$

These definitions can be rewritten without random bit strings and with probabilistic Turing Machines, although it is the same machinery under the hood.

2 Error Reduction by Repetition

Can we reduce our error? It's easy to see with the same computational resources we can reduce the error we make when deciding $x \in L$ for any $L \in$ **BPP**. Take an arbitrary language $L \in$ **BPP**. Then there exists a deterministic Turing machine M that decides L with error probability $1/2$ in polynomial time with respect to the length of its input. Then, what if we defined a new Turing machine M' that does the following computation:

$$M'(x, (r_1, r_2, \ldots, r_t)) = M(x, r_1) \vee M(x, r_2) \vee \ldots \vee M(x, r_t)$$

where each $r_i \in \{0,1\}^{p(|x|)}$ and t is polynomial with respect to $|x|$. Clearly, M' also runs in polynomial time with respect to length of its input. So, now lets bound its error probability. By the definition of **RP**, we have:

$$\begin{aligned} x \in L &\implies \Pr[M(x, r_1) = 1] = \ldots = \Pr[M(x, r_t) = 1] \geq 1/2 \\ &\implies \Pr[M'(x, (r_1, r_2, \ldots, r_t)) = 1] \geq 1 - 2^{-p(|x|)} \end{aligned}$$

and,

$$\begin{aligned} x \notin L &\implies M(x, r_1) = \ldots = M(x, r_t) = 0 \\ &\implies M'(x, (r_1, r_2, \ldots, r_t)) = 0 \vee \ldots \vee 0 = 0 \end{aligned}$$

implying we can guarantee an exponentially small error by having M' make $t = O(n)$ calls to M. The same can be done for **BPP**.

Take an arbitrary language $L \in \mathbf{BPP}$. Then, again lets define a new Turing machine M' that does the following computation:

$$M'(x, (r_1, r_2, \ldots, r_t)) = \text{Majority}(M(x, r_1), M(x, r_2), \ldots, M(x, r_t))$$

where each $r_i \in \{0,1\}^{p(|x|)}$ and t is polynomial with respect to $|x|$. Clearly, M' also runs in polynomial time with respect to length of its input. So, now lets bound its error probability. Observe that the $M(x, r_i)$ are a bunch of identically distributed indicator random variables. Therefore, when $x \in L$, by linearity of expectation, we have

$$E\left[\sum_{i=1}^{t} M(x, r_i)\right] = \sum_{i=1}^{t} E[M(x, r_i)] \geq \frac{2t}{3}.$$

Now, using Chernoff's Bound, we can write:

$$\begin{aligned} \Pr[M'(x, (r_1, r_2, \ldots, r_t)) = 1] &= \Pr[\text{Majority}(M(x, r_1), M(x, r_2), \ldots, M(x, r_t)) = 1] \\ &= 1 - \Pr\left[\sum_{i=1}^{t} M(x, r_i) - E\left[\sum_{i=1}^{t} M(x, r_i)\right] \geq \frac{t}{6}\right] \\ &\geq 1 - 2e^{-t/18} \end{aligned}$$

implying that again we can guarantee an exponentially small error by having M' make $t = O(n)$ calls to M.

3 Relationship Between BPP and P/poly

Theorem 3.1. BPP $\subset$ P/poly

Proof. Take $L \in \mathbf{BPP}$. By error reduction, we then have a Turing machine M with the property that

$$\Pr_{\substack{x \sim \{0,1\}^n \\ r \sim \{0,1\}^{p(n)}}} [M(x, r) = L(x)] \geq 1 - 2^{-|x|^2}.$$

It then follows that there exists some r^* for which:

$$\Pr_{x \sim \{0,1\}^n} [M(x, r^*) \neq L(x)] \leq 2^{-|x|^2}$$

Now, this implies that

$$\Pr_{x \sim \{0,1\}^n} [M(x, r^*) \neq L(x)] = 0$$

since if $M(x, r^*)$ failed for even a single $x \in \{0,1\}^n$, then we'd have the probability that M failed over all x was at least 2^{-n}, a contradiction. So, we have shown a single r^* works for all $x \in L$. Therefore, we could encode r^* into an advice function for a deterministic Turing machine that is always correct implying that $L \in \mathbf{P/poly}$ (note here r^* depends on $|x|$).