E0 224 Computational Complexity Theory
Lecture 19 (20 Oct 2014)

Lecturer: Chandan Saha *Scribe: Sarath A Y*

---

1 Introduction

In the last couple of lectures, we have been studying randomized computation. We defined the classes BPP and discussed examples of randomized algorithms. In this lecture, we define the classes $\mathsf{RP}, \mathsf{coRP}$ and ZPP. We introduce techniques for error reduction for randomized algorithms. Also, we show three important results: $\mathsf{ZPP} = \mathsf{RP} \cap \mathsf{coRP}$, $\mathsf{BPP} \subseteq \mathsf{P}_{/poly}$ and $\mathsf{BPP} \subseteq \Sigma_2^{\mathsf{P}} \cap \Pi_2^{\mathsf{P}}$.

2 One sided error: Class $\mathsf{RP}, \mathsf{coRP}$

Recall that we defined the class BPP to capture probabilistic algorithms (for a language L on an input x) that are allowed to make mistakes (with probability less than $1/3$) when $x \in L$ as well as $x \notin L$. Such algorithms are said to have two-sided error. In this section, we define the classes RP and coRP to capture algorithms with one-sided error property.

Definition: (Class RP) A language $L \subseteq \{0,1\}^*$ is in RP if there is a probabilistic polynomial time Turing machine M such that

$$x \in L \Leftrightarrow \mathbb{P}(M(x) = 1) \geq \frac{2}{3}$$
$$x \notin L \Leftrightarrow \mathbb{P}(M(x) = 0) = 1$$

Definition: (Class coRP) A language $L \subseteq \{0,1\}^*$ is in coRP if $\bar{L} \in \mathsf{RP}$. Equivalently, there is a probabilistic polynomial time machine M such that

$$x \in L \Leftrightarrow \mathbb{P}(M(x) = 1) = 1$$
$$x \notin L \Leftrightarrow \mathbb{P}(M(x) = 0) \geq \frac{2}{3}$$

Remark:
1. In the above definitions, the probability $\mathbb{P}(.)$ is over the random choices of the transition functions of the probabilistic machine M.
2. The classes RP and coRP remain the same even if we replace the constant $2/3$ by any constant strictly greater than 0.

Example: It is easy to see that PIT $\in$ coRP. Suppose that the given circuit $\mathcal{C}$ is actually identically zero. Then our randomized algorithm for PIT can never make a mistake since the circuit evaluates to 0 for all (random) choices of the input. On the other hand, if $\mathcal{C}$ is not identically zero, then the algorithm makes an error with small probability.

Observation: $\mathsf{RP}, \mathsf{coRP} \subseteq \mathsf{BPP}$. This can be immediately seen from the definitions of the classes $\mathsf{BPP}, \mathsf{RP}$ and coRP.

Observation: Since an NP machine N needs *only one* nondeterministic choice of transition function that takes N to q_{accept} for accepting a language, it follows that $\mathsf{RP} \subseteq \mathsf{NP}$. However, it is not known whether $\mathsf{BPP} \subseteq \mathsf{NP}$ (if the conjecture $\mathsf{BPP} = \mathsf{P}$ is true then indeed $\mathsf{BPP} \subseteq \mathsf{NP}$).

Next, we define the class ZPP to capture randomized algorithms that always outputs the correct answer.

Definition: (Class ZPP) A language $L \subseteq \{0,1\}^*$ is in ZPP if there is a probabilistic Turing machine M such that M always outputs the correct answer on input string $x \in \{0,1,\}^*$ for any x, however the expected running time of M on input x is bounded by some polynomial function of $|x|$.

Theorem 2.1. $\mathsf{ZPP} = \mathsf{RP} \cap \mathsf{coRP}$

Proof. ($\mathsf{ZPP} \subseteq \mathsf{RP}$): Suppose $L \subseteq \{0,1\}^*$ is a language in ZPP. We need to show that $L \in$ RP$\cap$coRP. Since $L \in \mathsf{ZPP}$, there is a probabilistic TM M with expected running time bounded by (a polynomial function) $q(|x|)$ that correctly decides the membership of every string $x \in \{0,1\}^*$. We describe an RP algorithm (M') as follows:

1. On input x run the machine M for $2q(|x|)$ many steps.
2. If M stops by this time, output whatever M outputs.
3. Otherwise output 0.

It is easy to see that this algorithm never makes a mistake when $x \notin L$. We wish to analyze the case when $x \in L$. Let $T(x)$ denote the running time of M on input x. Since $L \in \mathsf{ZPP}$, we have

$$\mathbb{E}(T(x)) \leq q(|x|)$$

where the expectation is taken over the random choices of the machine M. By Markov's inequality, we have

$$\begin{aligned} \mathbb{P}(T(x) \geq 2q(x)) &\leq \frac{\mathbb{E}(T(x))}{2q(x)} \\ &\leq \frac{q(x)}{2q(x)} = \frac{1}{2} \end{aligned}$$

Since the probability that M has running time more than $2q(x)$ is less than $1/2$, it follows that

$$x \in L \Rightarrow \quad \mathbb{P}(M'(x) = 0) < \frac{1}{2}$$

Thus, M' is an RP machine for the language L, and $L \in \mathsf{RP}$. Now, to show that $L \in \mathsf{coRP}$, we simply replace step 3 in the RP algorithm by *otherwise output 1.* Then, it is easy to see that the algorithm never makes a mistake when $x \in L$. On the other hand, if $x \notin L$, the probability that M' makes a mistake is strictly less than $1/2$ (using the same argument), and hence $L \in \mathsf{coRP}$. Therefore, $\mathsf{ZPP} \subseteq \mathsf{RP} \cap \mathsf{coRP}$. □

We leave it as an exercise to show that $\mathsf{RP} \cap \mathsf{coRP} \subseteq \mathsf{ZPP}$

3 Error reduction for BPP and RP

In this section, we discuss how to reduce the error bound in RP, coRP and BPP algorithms.

3.1 Error reduction for one-sided error

Suppose $L \in \mathsf{RP}$. Then by definition, there is a probabilistic Turing machine M such that

$$x \in L \Leftrightarrow \mathbb{P}(M(x) = 1) \geq \frac{2}{3}$$
$$x \notin L \Leftrightarrow \mathbb{P}(M(x) = 1) = 0$$

and running time of M is bounded by (a polynomial function) $q(n)$. We would like to construct an RP algorithm whose error probability is exponentially small. Let $p(.)$ be a polynomial function and consider the following algorithm (M') on an input x (let $n = |x|$).

1. Execute M independently for $p(n)$ times
2. Output the 'OR' of outputs of the $p(n)$ executions of M

Clearly, when $x \notin L$, $M'(x)$ cannot make a mistake (since M will not make a mistake). Suppose $x \in L$. Then $M'(x)$ makes an error when all the $p(n)$ executions of M output $M(x) = 0$. And, since all the $p(n)$ executions of M are independent, the probability of this happening is bounded by $\left(\frac{1}{3}\right)^{p(n)}$. Thus, we have a probabilistic polynomial time machine M' such that

$$x \in L \Leftrightarrow \mathbb{P}(M'(x) = 0) \leq \left(\frac{1}{3}\right)^{p(n)}$$
$$x \notin L \Leftrightarrow \mathbb{P}(M'(x) = 0) = 1$$

Thus, we have an RP algorithm with exponentially small one-sided error probability. We can construct a similar procedure for error reduction for coRP as well.

Remark: This argument works even if the constant $2/3$ (in the definition of class RP) is replaced by any positive constant bounded away from 0. This is because, if $x \in L$, then there is a positive probability for $M(x) = 1$, and by running M for sufficiently (and polynomially) many times, M' would correctly decide the membership of x in L with high probability.

3.2 Error reduction for BPP

Let $L \in \mathsf{BPP}$. Then, by definition, there is a probabilistic TM M and a polynomial function $q(.)$ such that

$$\mathbb{P}(M(x) = L(x)) \geq \frac{2}{3}$$

and running time of M is bounded by $q(|x|)$. Let $\mathsf{BPP}_{\frac{1}{2}+\frac{1}{|x|^c}}$ denote the class of languages for which there is a polynomial time probabilistic TM M such that $\mathbb{P}(M(x) = L(x)) \geq 1/2+1/(|x|^c)$ for all inputs $x \in \{0,1\}^*$. Let $L \in \mathsf{BPP}_{\frac{1}{2}+\frac{1}{|x|^c}}$ for some constant $c > 0$. We show that for any constant $d > 0$ there is a probabilistic polynomial time TM M' such that for all $x \in \{0,1\}^*$, $\mathbb{P}(M'(x) = L(x)) \geq 1 - \frac{1}{2^{|x|^d}}$. By definition, there is a probabilistic TM M and a polynomial function q such that

$$\mathbb{P}(M(x) = L(x)) \geq \frac{1}{2} + \frac{1}{|x|^c} \tag{1}$$

and the running time of M is bounded by $q(|x|)$. We describe a probabilistic polynomial time TM M' that does the following:

1. Execute M independently on input x (let $n = |x|$) for k times (k depends on c and d, we will find suitable k later). Let $z_1, \ldots, z_k$ denote the outputs of these independent executions of M.
2. Output the majority of $z_1, \ldots, z_k$

We use Chernoff bound to show that $L \in \mathsf{BPP}_{1-\frac{1}{2^{n^d}}}$.
Let $y_1, \ldots, y_k$ be independent Boolean random variables. Let $\mathbb{P}(y_i = 1) = \rho \ \forall i$. Define

$$Y := \sum_{i=1}^{k} y_i \tag{2}$$

Then, for $0 < \delta < 1$, Chernoff bound says

$$\mathbb{P}(Y \leq (1-\delta)\mathbb{E}(Y)) \leq e^{-\frac{k\rho\delta^2}{3}} \tag{3}$$

Let

$$y_i = \begin{cases} 1, & \text{if } z_i \text{ is the correct output} \\ 0, & \text{otherwise} \end{cases}$$

Since we execute M independently for k times, y_i's are independent random variables. Since $L \in \mathsf{BPP}$, we have

$$\rho := \mathbb{P}(y_i = 1) \geq \frac{1}{2} + \frac{1}{n^c} \quad \forall i$$

Also, let $\mu = \mathbb{E}(Y)$. Then we have

$$\begin{aligned}\mu &= k\mathbb{E}(y_1) \\ &= k\rho \\ &\geq k\left(\frac{1}{2} + \frac{1}{n^c}\right) \end{aligned} \tag{4}$$

The machine M' makes a mistake on input x when majority of the answers of M are wrong, i.e., when $Y < \frac{1}{2}k$. Therefore, consider $\mathbb{P}(Y < \frac{k}{2})$. Since the machine M outputs the correct answer with probability strictly greater than $1/2$, we can find a δ, $0 < \delta < 1$, such that the event $Y < k/2$ implies the event $Y \leq (1-\delta)\rho$ (which in turn implies $\mathbb{P}(Y < k/2) \leq \mathbb{P}(Y \leq (1-\delta)\rho)$). Let us fix a δ such that this happens, i.e.,

$$\begin{aligned} Y < \frac{1}{2}k &\Rightarrow Y \leq (1-\delta)\mathbb{E}(Y) \\ \text{i.e.,} \quad \frac{1}{2}k &\leq (1-\delta)\rho k \\ \Rightarrow \quad \delta &\leq 1 - \frac{1}{2\rho} \end{aligned}$$

Thus, we fix

$$\delta = 1 - \frac{1}{2\rho} \tag{5}$$

and we have

$$\begin{aligned}\mathbb{P}(Y < \frac{1}{2}k) &\leq \mathbb{P}(Y \leq (1-\delta)\mathbb{E}(Y)) \\ &\leq e^{-\frac{k\rho\delta^2}{3}} \\ &= e^{-\frac{k\rho}{3}(1-\frac{1}{\rho}+\frac{1}{4\rho^2})} \\ &= e^{-\frac{k}{3}(\rho-1+\frac{1}{4\rho})} \\ &\leq e^{-\frac{k}{3}(\frac{1}{2}+\frac{1}{n^c}-1+\frac{1}{2+\frac{4}{n^c}})} \\ &\leq e^{-\frac{k}{3n^c}} \end{aligned} \tag{6}$$

where the second inequality follows from Chernoff bound, and the first equality from the choice of δ. Now, let us choose $k = 3n^{c+d}$. Then we get

$$\mathbb{P}(Y < \frac{1}{2}k) \leq e^{-n^d} \tag{7}$$

This shows that the probability that the machine M' makes an error is exponentially smaller. Also, note that by the choice of k, M' runs in polynomial time. Thus, we have proved the following theorem.

Theorem 3.1. *For any constants $c, d > 0$, $\mathsf{BPP}_{\frac{1}{2}+\frac{1}{n^c}} = \mathsf{BPP}_{1-\frac{1}{2^{n^d}}}$*

4 BPP and other classes

In this section, we study relationships between BPP and other complexity classes. We already observed that $\mathsf{RP}, \mathsf{coRP} \subseteq \mathsf{BPP}$. We now show two important results: $\mathsf{BPP} \subseteq \mathsf{P}_{/poly}$ and $\mathsf{BPP} \subseteq \Sigma_2^{\mathsf{P}} \cap \Pi_2^{\mathsf{P}}$.

Theorem 4.1. $\mathsf{BPP} \subseteq \mathsf{P}_{/poly}$

Proof. Suppose $L \in \mathsf{BPP}$. Then by using the alternative definition of BPP and the error reduction procedure, there is a deterministic polynomial time machine M such that, on any input $x \in \{0,1\}^*$

$$\mathbb{P}(M(x,r) \neq L(x)) \leq \frac{1}{2^{|x|^2}} \tag{8}$$

where the probability is taken over the random choices of the string r (let $m = |r|$, then m is a polynomial function of $|x|$). Now, it suffices to show the existence of a single string r such that $M(x,r) = L(x)$ for all inputs $x \in \{0,1\}^n$, since such a string r can be hardwired into a circuit $\mathcal{C}_n$ of polynomial size such that $\mathcal{C}_n(x) = L(x)$ for all $x \in \{0,1\}^n$. By finding an r for each n, we get a polynomial sized circuit family $\{\mathcal{C}_n\}_{n\geq 1}$ that decides L and this would imply that $L \in \mathsf{P}_{/poly}$ and hence $\mathsf{BPP} \subseteq \mathsf{P}_{/poly}$.

We show the existence of such a string r using a counting argument. Let $|x| = n$. We say that a string r is bad for x if $M(x,r) \neq L(x)$, otherwise we say r is good for x. By statement 8, it follows that the fraction of bad strings for input x is at most $\frac{1}{2^{n^2}}$. Since r is m bit long, the total number of bad strings for an x is at most $\frac{2^m}{2^{n^2}}$. Therefore, the number of strings that are bad for some x is at most $\frac{2^m}{2^{n^2}} 2^n$. But,

$$\frac{2^m}{2^{n^2}} 2^n = 2^{m+n-n^2} < 2^m \tag{9}$$

and since the number of strings of length m is 2^m, it follows that there exists at least one m bit string r that is good for all $x \in \{0,1\}^n$. This completes the proof. □

Theorem 4.2. *(Sipser-Gács)* $\mathsf{BPP} \subseteq \Sigma_2^{\mathsf{P}} \cap \Pi_2^{\mathsf{P}}$

Proof. Since $\mathsf{BPP} = \mathsf{coBPP}$, it suffices to show that $\mathsf{BPP} \subseteq \Sigma_2^{\mathsf{P}}$. Let L be a language in BPP. By definition (and the error reduction procedure for BPP), there is a deterministic polynomial time TM M such that

$$\mathbb{P}(M(x,r) \neq L(x)) \leq \frac{1}{2^{|x|^2}}$$

where the probability is taken over random choices of string r. More precisely, we have

$$x \in L \Rightarrow \mathbb{P}(M(x,r) = 1) \geq 1 - \frac{1}{2^{|x|^2}} \tag{10}$$

$$x \notin L \Rightarrow \mathbb{P}(M(x,r) = 1) \leq \frac{1}{2^{|x|^2}} \tag{11}$$

Let S denote the set of strings r for which $M(x,r) = 1$, and let $|x| = n$. Then from the statements 10-11 we see that $|S| \geq (1 - \frac{1}{2^{n^2}})2^m$ or $|S| < \frac{2^m}{2^{n^2}}$ depending on whether $x \in L$ or $x \notin L$

respectively. We show that, in the first case, all possible strings $r \in \{0,1\}^m$ can be obtained by translating (see the definition of translation below) the set S using polynomially many strings (Claim 4.4), whereas in the second case we cannot cover all possible r's by translating the set S (Claim 4.3). The essential idea to show $L \in \Sigma_2^{\mathsf{P}}$ is to express the membership of x in L by extra quantifiers by making use of Claims 4.3-4.4.

Definition: For a string $u \in \{0,1\}^m$, define the translation of the set S using u as $S \oplus u := \{s \oplus u : s \in S\}$ where $s \oplus u$ denotes the 'XOR' operation.

Let $k = \frac{m}{n} + 1$. We claim the following:

Claim 4.3. *If $|S| < \frac{1}{2^{n^2}} 2^m$ then for every $u_1, u_2, \ldots, u_k$ where $u_i \in \{0,1\}^m$, there exists an r that does not belong to $\cup_{i=1}^k (S \oplus u_i)$.*

Proof. We have

$$\begin{aligned} \left| \bigcup_{i=1}^{k} (S \oplus u_i) \right| &\leq k|S| \\ &< \left(\frac{m}{n} + 1\right) \left(\frac{1}{2^{n^2}} 2^m\right) \qquad (\text{as } m = poly(n)) \\ &< 2^m \end{aligned} \tag{12}$$

where the first inequality follows from union bound, and the second inequality from the choice of k and the assumption on $|S|$. Thus, $\cup_{i=1}^k (S \oplus u_i)$ cannot cover all the strings of length m and hence there exists an m length string r such that $r \notin \cup_{i=1}^k (S \oplus u_i)$ □

Claim 4.4. *If $|S| \geq (1 - \frac{1}{2^{n^2}}) 2^m$ then there exists $u_1, u_2, \ldots, u_k$ such that for every $r \in \{0,1\}^m$, $r \in \cup_{i=1}^k (S \oplus u_i)$.*

Proof. We show the following. Suppose that $u_1, u_2, \ldots, u_k$ are chosen independently and uniformly at random from $\{0,1\}^m$, then

$$\mathbb{P}\left(\exists r, r \notin \cup_{i=1}^k (S \oplus u_i)\right) < 1 \tag{13}$$

where the probability is over the random choices of strings $u_1, \ldots, u_k$. First, we observe that for a fixed $r \in \{0,1\}^m$, $r \oplus u_i$ is uniformly distributed over the space of $\{0,1\}^m$ strings for all $i = 1, 2, \ldots, k$, because $u_1, u_2, \ldots, u_k$ are picked uniformly and independently at random from $\{0,1\}^m$. Let us compute the probability that $r \notin \cup_{i=1}^k (S \oplus u_i)$. Observe that

$$r \notin \cup_{i=1}^k (S \oplus u_i) \Leftrightarrow r \notin S \oplus u_1 \text{ and } r \notin S \oplus u_2 \text{ and} \ldots r \notin S \oplus u_k$$

But,

$$r \notin S \oplus u_i \Leftrightarrow r \oplus u_i \notin S \quad \forall i = 1, 2, \ldots, k$$

Therefore,

$$\begin{aligned}\mathop{\mathbb{P}}_{u_1}(r \notin S \oplus u_1) &= \mathop{\mathbb{P}}_{u_1}(r \oplus u_1 \notin S) \\ &\leq \frac{1}{2^{n^2}}\end{aligned}$$

Similarly,

$$\begin{aligned}\mathop{\mathbb{P}}_{u_2}(r \notin S \oplus u_2) &= \mathop{\mathbb{P}}_{u_2}(r \oplus u_2 \notin S) \\ &\leq \frac{1}{2^{n^2}}\end{aligned}$$

and so on. Also, note that the events $r \oplus u_i \notin S$ and $r \oplus u_j \notin S$ are independent for $i \neq j$, since all the u_i's are picked independently. Thus,

$$\begin{aligned}\mathop{\mathbb{P}}_{u_1,\ldots,u_k}(r \notin \cup_{i=1}^{k}(S \oplus u_i)) &= \prod_{i=1}^{k} \mathop{\mathbb{P}}_{u_i}(r \notin S \oplus u_i) \\ &\leq \left(\frac{1}{2^{n^2}}\right)^k \\ &\leq \frac{1}{2^{mn}}\end{aligned}$$

where the last inequality follows from the choice of k. Now, using the union bound,

$$\begin{aligned}\mathop{\mathbb{P}}_{u_1,\ldots,u_k}\left(\exists r, r \notin \cup_{i=1}^{k}(S \oplus u_i)\right) &\leq \sum_{r:r\in\{0,1\}^m} \mathop{\mathbb{P}}_{u_1,\ldots,u_k}(r \notin \cup_{i=1}^{k}(S \oplus u_i)) \\ &\leq \frac{1}{2^{mn}} 2^m \\ &= \frac{1}{2^{(n-1)m}} \qquad (14)\end{aligned}$$

which is strictly less than 1 for nontrivial values of m and n. Therefore, we conclude that there exists a choice of $u_1, \ldots, u_k$ such that for any string $r \in \{0,1\}^m$, $r \in \cup_{i=1}^{k}(S \oplus u_i)$. □

Using the above two claims, we see that

$$x \in L \Leftrightarrow \exists u_1, u_2, \ldots, u_k \;\; s.t. \;\; \forall r \in \{0,1\}^m \left[r \in \bigcup_{i=1}^{k}(S \oplus u_i) \right]$$

That is,

$$\begin{aligned}x \in L &\Leftrightarrow \exists u_1, u_2, \ldots, u_k \in \{0,1\}^m \;\; \forall r \in \{0,1\}^m \, [r \oplus u_i \in S \text{ for at least one } i \in [k]] \\ &\Leftrightarrow \exists u_1, u_2, \ldots, u_k \in \{0,1\}^m \;\; \forall r \in \{0,1\}^m \;\; \left[\bigvee_{i=1}^{k} (M(x, r \oplus u_i) = 1) \right] \qquad (15)\end{aligned}$$

Note that the expression in squared brackets in the above expression can be evaluated in polynomial time, by the choice of k. Thus, 15 is a Σ_2^{P} expression and we conclude that $L \in \Sigma_2^{\mathsf{P}}$. Hence, $\mathsf{BPP} \subseteq \Sigma_2^{\mathsf{P}} \cap \Pi_2^{\mathsf{P}}$. □

Remark: 1. We have shown $\mathsf{P} \subseteq \mathsf{BPP} \subseteq \mathsf{P}_{/poly}$, and also $\mathsf{BPP} \subseteq \Sigma_2^{\mathsf{P}} \cap \Pi_2^{\mathsf{P}}$. Thus, if $\mathsf{P} = \mathsf{NP}$, it follows that $\mathsf{BPP} = \mathsf{P}$. However, even though we believe that $\mathsf{P} \neq \mathsf{NP}$, we conjecture that $\mathsf{P} = \mathsf{BPP}$.

2. Since $\mathsf{BPP} \subseteq \mathsf{P}_{/poly}$, the Karp-Lipton theorem tells us that if we can solve SAT in probabilistic polynomial time, then the polynomial hierarchy collapses to level 2.

3. Recently, it has been show that BPP is contained in a class that is weaker than $\Sigma_2^{\mathsf{P}} \cap \Pi_2^{\mathsf{P}}$.

References

[1] Sanjeev Arora, and Boaz Barak. *Computational Complexity: A Modern Approach*, Cambridge University Press, 2009.