CS 517
Homework 6

Ameyassh Nagarajan

June 2024

1 Problem 1

RP Definition

A language L is in RP (Randomized Polynomial time) if there exists a probabilistic polynomial-time Turing machine M such that:

- If $x \in L$, $M(x)$ accepts with probability at least $\frac{1}{2}$.
- If $x \notin L$, $M(x)$ always rejects.

P/poly Definition

A language L is in P/poly if there exists a language A in P and a set of advice strings $\{a_0, a_1, a_2, \ldots\}$ such that:

- $|a_n| \leq n^{O(1)}$, where a_n is the advice string for inputs of length n.
- For all x of length n, $x \in L$ if and only if $(x, a_n) \in A$.

Alternatively, a language L is in P/poly if there exists a family of circuits $\{C_0, C_1, \ldots\}$ such that:

- $|C_n| \leq n^{O(1)}$.
- For all x of length n, $x \in L$ if and only if $C_n(x) = 1$.

1.1 Proof

Let L be a language in RP. By definition, there exists a probabilistic polynomial-time Turing machine M such that:

- If $x \in L$, $M(x)$ accepts with probability at least $\frac{1}{2}$.
- If $x \notin L$, $M(x)$ always rejects.

For each input length n, consider the probabilistic machine M. By the probabilistic nature of M, there exists a fixed random tape r_n of polynomial length such that $M(x, r_n)$ works correctly for at least half of the inputs of length n. Specifically, $M(x, r_n)$ accepts $x \in L$ with probability at least 1/2 and always rejects $x \notin L$.

Construct a circuit C_n that simulates $M(x, r_n)$ for inputs of length n. This circuit C_n is polynomial in size because it simulates a polynomial-time Turing machine with a fixed random tape. The size of C_n is bounded by some polynomial $p(n)$.

To handle the remaining inputs not correctly decided by C_n, we recursively apply the same process. For the remaining half of the inputs, there exists another random tape r'_n that works for at least half of those inputs. We construct a new circuit C'_n for these inputs. Repeating this process, we eventually cover all inputs of length n.

Thus, for each input length n, we construct a family of circuits $\{C_n\}$ such that for every input x of length n, there is a circuit in the family that correctly decides whether $x \in L$.

Each random tape r_n used in the construction is of polynomial length in n, satisfying the requirement that the advice length is polynomially bounded by the input length.

Hence, $L \in \text{P/poly}$, and we have shown that $\text{RP} \subseteq \text{P/poly}$.

Conclusion

Since we have shown that for any language in RP, there exists a family of polynomial-size circuits that correctly decides the language, we conclude that:

$$\text{RP} \subseteq \text{P/poly}$$

2 Problem 2

EXP is a Subset of XPP

To show that EXP $\subseteq$ XPP, we need to construct a probabilistic Turing machine M for any language $L \in$ EXP such that:

1 $x \in L \implies \Pr[M(x) = 1] \geq 1/2$

2 $x \notin L \implies \Pr[M(x) = 1] < 1/2$

with expected polynomial running time.

Construction of the Probabilistic Turing Machine

Let $L \in$ EXP be decided by a deterministic Turing machine D that runs in time $2^{p(n)}$ for some polynomial $p(n)$. We construct a probabilistic Turing machine M that, on input x of length n:

1 With probability $1 - \frac{1}{2^n}$, M performs a trivial polynomial-time computation (e.g., returns 0).

2 With probability $\frac{1}{2^n}$, M simulates D on x, which takes time $2^{p(n)}$.

Expected Running Time

Let $T_{\text{poly}}(x)$ be the polynomial-time computation and $T_{\text{exp}}(x)$ be the exponential-time computation. The expected running time $E[T(x)]$ is:

$$E[T(x)] = \left(1 - \frac{1}{2^n}\right) T_{\text{poly}}(x) + \frac{1}{2^n} T_{\text{exp}}(x)$$

Given that $T_{\text{poly}}(x)$ is polynomial and $T_{\text{exp}}(x) = 2^{p(n)}$, we have:

$$E[T(x)] = \left(1 - \frac{1}{2^n}\right) \text{poly}(n) + \frac{1}{2^n} 2^{p(n)}$$

Since $\frac{1}{2^n} 2^{p(n)} = 2^{p(n)-n}$ and $p(n)$ is a polynomial, $2^{p(n)-n}$ becomes negligible for large n. Thus, $E[T(x)]$ remains polynomial.

Probability Analysis Using Chernoff Bound

Define X_i as a Bernoulli random variable which is 1 if the i-th trial runs in exponential time, and 0 otherwise. Let $X = \sum_{i=1}^{n} X_i$ be the sum of n trials. The expected value $E[X]$ is:

$$E[X] = n \cdot \frac{1}{2^n}$$

Applying the Chernoff bound, for $\delta > 0$:

$$\Pr[X \geq (1+\delta)E[X]] \leq \exp\left(-\frac{\delta^2 E[X]}{2+\delta}\right)$$

Choosing $\delta = \frac{1}{2}$, we get:

$$\Pr\left[X \geq \frac{3}{2}E[X]\right] \leq \exp\left(-\frac{\left(\frac{1}{2}\right)^2 E[X]}{2+\frac{1}{2}}\right) = \exp\left(-\frac{E[X]}{10}\right)$$

Since $E[X] = n \cdot \frac{1}{2^n}$, we have:

$$\Pr\left[X \geq \frac{3}{2}E[X]\right] \leq \exp\left(-\frac{n}{10 \cdot 2^n}\right)$$

This probability is extremely small for large n.

Conclusion for EXP

By carefully balancing the probabilities and running times, we ensure the expected running time remains polynomial while the correctness conditions for $x \in L$ and $x \notin L$ are satisfied. This approach effectively "sneaks" in the exponential computation with sufficiently low probability, thereby proving EXP $\subseteq$ XPP.

Pushing Beyond EXP

To push the method beyond EXP, we consider classes such as 2-EXP, the class of languages decidable by deterministic Turing machines in doubly exponential time, $2^{2^{p(n)}}$ for some polynomial $p(n)$.

Construction of the Probabilistic Turing Machine

Let $L \in$ 2-EXP be decided by a deterministic Turing machine D that runs in time $2^{2^{p(n)}}$ for some polynomial $p(n)$. We construct a probabilistic Turing machine M that, on input x of length n:

1 With probability $1 - \frac{1}{2^{2^n}}$, M performs a trivial polynomial-time computation.

2 With probability $\frac{1}{2^{2^n}}$, M simulates D on x, which takes time $2^{2^{p(n)}}$.

Expected Running Time

Let $T_{\text{poly}}(x)$ be the polynomial-time computation and $T_{\exp}(x)$ be the doubly exponential-time computation. The expected running time $E[T(x)]$ is:

$$E[T(x)] = \left(1 - \frac{1}{2^{2^n}}\right) T_{\text{poly}}(x) + \frac{1}{2^{2^n}} T_{\exp}(x)$$

Given that $T_{\text{poly}}(x)$ is polynomial and $T_{\exp}(x) = 2^{2^{p(n)}}$, we have:

$$E[T(x)] = \left(1 - \frac{1}{2^{2^n}}\right) \text{poly}(n) + \frac{1}{2^{2^n}} 2^{2^{p(n)}}$$

Since $\frac{1}{2^{2^n}} 2^{2^{p(n)}} = 2^{2^{p(n)} - 2^n}$ and $2^{p(n)} - 2^n$ is extremely negative for large n, this term becomes negligible. Thus, $E[T(x)]$ remains polynomial.

Probability Analysis Using Chernoff Bound

Define X_i as a Bernoulli random variable which is 1 if the i-th trial runs in exponential time, and 0 otherwise. Let $X = \sum_{i=1}^{n} X_i$ be the sum of n trials. The expected value $E[X]$ is:

$$E[X] = n \cdot \frac{1}{2^{2^n}}$$

Applying the Chernoff bound, for $\delta > 0$:

$$\Pr[X \geq (1+\delta)E[X]] \leq \exp\left(-\frac{\delta^2 E[X]}{2+\delta}\right)$$

To illustrate the effect of increasing δ, we calculate the bound for different values of δ:

1. For $\delta = \frac{1}{2}$:

$$\Pr[X \geq \frac{3}{2}E[X]] \leq \exp\left(-\frac{\left(\frac{1}{2}\right)^2 E[X]}{2 + \frac{1}{2}}\right) = \exp\left(-\frac{E[X]}{10}\right)$$

2. For $\delta = \frac{3}{2}$:

$$\Pr\left[X \geq \left(1 + \frac{3}{2}\right) E[X]\right] \leq \exp\left(-\frac{\left(\frac{3}{2}\right)^2 E[X]}{2 + \frac{3}{2}}\right) = \exp\left(-\frac{9E[X]}{14}\right)$$

3. For $\delta = 2$:

$$\Pr[X \geq 3E[X]] \leq \exp\left(-\frac{4E[X]}{4}\right) = \exp\left(-E[X]\right)$$

In general, as δ increases, the term $\frac{\delta^2 E[X]}{2+\delta}$ increases, making $\exp\left(-\frac{\delta^2 E[X]}{2+\delta}\right)$ smaller. This means the probability of X deviating from $E[X]$ by a factor of $1+\delta$ decreases exponentially.

Conclusion for Pushing Beyond EXP

By carefully balancing the probabilities and running times, we ensure the expected running time remains polynomial while the correctness conditions for $x \in L$ and $x \notin L$ are satisfied. This approach effectively "sneaks" in the exponential computation with sufficiently low probability, thereby proving EXP $\subseteq$ XPP and extending the method to higher complexity classes like 2-EXP.