Undergraduate Complexity Theory
Lecture 22: BPP

Marcythm

July 24, 2022

1 Lecture Notes

Definition 1.1. BPP, bounded error probabilistic computation (two sided-error prob poly time), $L \in \mathsf{BPP}$ if $\exists$ PTM N s.t.

$$x \in L \implies \mathbf{Pr}[N(x) \text{ accept}] \geq 2/3$$
$$x \notin L \implies \mathbf{Pr}[N(x) \text{ accept}] \leq 1/3$$

current hierarchy: $\mathsf{P} \subseteq \mathsf{ZPP} \subseteq \mathsf{RP}, \mathsf{coRP} \subseteq \mathsf{RP} \cup \mathsf{coRP} \subseteq \mathsf{BPP}$, all these are believed to be equal!
Alternative View: DTM $M(x, r)$ with input x and a random tape input r.

Lemma 1.2. $\mathsf{BPP} \subseteq \mathsf{EXP}$.

Corollary 1.3. $\mathsf{BPP} \subseteq \mathsf{PSPACE}$.

$\mathsf{BPP} \subseteq \mathsf{NP}$? not known. Even cannot separate BPP from NEXP.

Theorem 1.4. $\mathsf{P} = \mathsf{NP} \implies \mathsf{P} = \mathsf{BPP}$ *(contra:* $\mathsf{P} \neq \mathsf{BPP} \implies \mathsf{P} \neq \mathsf{NP}$*).*

Definition 1.5. $\mathsf{P/poly}$ is the class of languages with a circuit family of poly size deciding it.

Theorem 1.6. $\mathsf{BPP} \subseteq \mathsf{P/poly}$.

Proof. $L \in \mathsf{BPP}$, $\exists$ DTM M, $M(x, r)$ acc with $p \geq 1 - 2^{-2|x|}$ if $x \in L$, acc with $p \leq 2^{-2|x|}$ if $x \notin L$ in poly time. M can be translated into poly size circuit C_M, which has two kinds of inputs, x and r. The r input is what we should get rid of.
For each fixed $x \in \{0, 1\}^n$, all but $1/4^n$ of random coins r yield correct answer. Since only 2^n possible x and $1/4^n$ bad r for each, there are at most $1/2^n$ rs are bad for some x, i.e. most of rs are simultaneously good for all x, so find them and hardwire them into circuit. $\square$

Derandomization:

Theorem 1.7 ('98)**.** *If* $\mathsf{3SAT}$ *requires circuit family of size* $2^{\delta n}$ *for some* $\delta > 0$*, then* $\mathsf{P} = \mathsf{BPP}$.

two major steps: (worst-case hardness) to (strong average-case hardness) to (PRNG)

2 Reading

2.1 sipser 10.2 (Probabilistic Algorithms)

2.1.1 Read-Once Branching Programs

proof of $EQ_{\mathsf{ROBP}} \in \mathsf{BPP}$: construct polynomial, randomly select an element in finite field.