Computational Complexity Theory

Lecture 11: PTMs; Classes BPP, RP and ZPP; Sipser-Gacs-Lautemann theorem

Department of Computer Science,
Indian Institute of Science

Randomized computation

- So far, we have used deterministic TMs to model "real-world" computation. But, DTMs don't have the ability to make *random choices* during a computation.

- The usefulness of randomness in computation was realized as early as the 1940s when the first electronic computer, ENIAC, was developed.

Randomized computation

- So far, we have used deterministic TMs to model "real-world" computation. But, DTMs don't have the ability to make *random choices* during a computation.

- The usefulness of randomness in computation was realized as early as the 1940s when the first electronic computer, ENIAC, was developed.
  - The use of statistical methods in a computational model of a thermonuclear reaction for the ENIAC lead to the invention of the ***Monte Carlo methods***.

Randomized computation

- So far, we have used deterministic TMs to model "real-world" computation. But, DTMs don't have the ability to make *random choices* during a computation.

- The usefulness of randomness in computation was realized as early as the 1940s when the first electronic computer, ENIAC, was developed.

- To study randomized computation, we need to give TMs the power of generating random numbers.

Randomized computation

- How realistic such a randomized TM model would be depends on our ability to generate bits that are "close" to being truly random.

1 with probability ½
0 with probability ½

Randomized computation

- How realistic such a randomized TM model would be depends on our ability to generate bits that are "close" to being truly random.
- Many programming languages have built-in random number generator functions.
- Examples of pseudo-random number generators are *linear congruential generators* and von Neumann's *middle-square method*.

Randomized computation

- How realistic such a randomized TM model would be depends on our ability to generate bits that are "close" to being truly random.
- Many programming languages have built-in random number generator functions.
- Examples of pseudo-random number generators are *linear congruential generators* and von Neumann's *middle-square method*.

$$X_{i+1} = aX_i + c \pmod m$$

Randomized computation

- How realistic such a randomized TM model would be depends on our ability to generate bits that are “close” to being truly random.
- Many programming languages have built-in random number generator functions.
- Examples of pseudo-random number generators are *linear congruential generators* and von Neumann’s *middle-square method*.

Square an n bit number to get a $2n$ bit number and take the middle n bits.

Randomized computation

- How realistic such a randomized TM model would be depends on our ability to generate bits that are “close” to being truly random.
- Many programming languages have built-in random number generator functions.
- Examples of pseudo-random number generators are *linear congruential generators* and von Neumann’s *middle-square method*.
- To what extent a PRG is adequate is studied under the topic `*Pseudorandomness*’ in complexity theory.

Randomized computation

- How realistic such a randomized TM model would be depends on our ability to generate bits that are "close" to being truly random.
- Many programming languages have built-in random number generator functions.
- Examples of pseudo-random number generators are *linear congruential generators* and von Neumann's *middle-square method*.
- We'll assume that a TM can generate, or has access to, truly random bits/coins. (We'll touch upon "truly vs biased random bits" at end of the lecture.)

Probabilistic Turing Machines

- Definition. A *probabilistic Turing machine* (PTM) M has two transition functions δ_0 and δ_1. At each step of computation on input $x \in \{0,1\}^*$, M applies one of δ_0 and δ_1 uniformly at random (independent of the previous steps). M outputs either 1 (accept) or 0 (reject).

Probabilistic Turing Machines

- Definition. A *probabilistic Turing machine* (PTM) M has two transition functions δ_0 and δ_1. At each step of computation on input $x \in \{0,1\}^*$, M applies one of δ_0 and δ_1 uniformly at random (independent of the previous steps). M outputs either 1 (accept) or 0 (reject). M runs in $T(n)$ time if M always halts within $T(|x|)$ steps *regardless of its random choices*.

Probabilistic Turing Machines

- Definition. A *probabilistic Turing machine* (PTM) M has two transition functions δ_0 and δ_1. At each step of computation on input $x \in \{0,1\}^*$, M applies one of δ_0 and δ_1 uniformly at random (independent of the previous steps). M outputs either 1 (accept) or 0 (reject). M runs in $T(n)$ time if M always halts within $T(|x|)$ steps *regardless of its random choices*.

- Note. PTMs and NTMs are syntatically similar – both have two transition functions.

Probabilistic Turing Machines

- Definition. A *probabilistic Turing machine* (PTM) M has two transition functions δ_0 and δ_1. At each step of computation on input $x \in \{0,1\}^*$, M applies one of δ_0 and δ_1 uniformly at random (independent of the previous steps). M outputs either 1 (accept) or 0 (reject). M runs in $T(n)$ time if M always halts within $T(|x|)$ steps *regardless of its random choices*.

- Note. But, semantically, they are quite different – unlike NTMs, PTMs are meant to model realistic computation devices.

Probabilistic Turing Machines

- Definition. A *probabilistic Turing machine* (PTM) M has two transition functions δ_0 and δ_1. At each step of computation on input $x \in \{0,1\}^*$, M applies one of δ_0 and δ_1 uniformly at random (independent of the previous steps). M outputs either 1 (accept) or 0 (reject). M runs in T(n) time if M always halts within T(|x|) steps *regardless of its random choices*.

- Note. The above definition allows a PTM M to not halt on some computation paths defined by its random choices (unless we explicitly say that M runs in T(n) time). More on this later when we define ZPP.

Class BPP

- Definition. A PTM M *decides* a language L in time $T(n)$ if M runs in $T(n)$ time, and for every $x \in \{0,1\}^*$,

$$\Pr[M(x) = L(x)] \geq 2/3.$$

- Definition. A language L is in $\mathrm{BPTIME}(T(n))$ if there's PTM that decides L in $O(T(n))$ time.

- Definition. $\mathrm{BPP} = \bigcup_{c > 0} \mathrm{BPTIME}(n^c)$.

- Clearly, $\mathrm{P} \subseteq \mathrm{BPP}$.

Class BPP

- Definition. A PTM M *decides* a language L in time T(n) if M runs in T(n) time, and for every $x \in \{0,1\}^*$,

$$\underbrace{\Pr[M(x) = L(x)]}_{\text{Success probability}} \geq 2/3.$$

- Definition. A language L is in BPTIME(T(n)) if there's PTM that decides L in O(T(n)) time.

- Definition. $\text{BPP} = \bigcup_{c>0} \text{BPTIME}(n^c)$.

- Clearly, $\text{P} \subseteq \text{BPP}$.

Class BPP

- Definition. A PTM M *decides* a language L in time T(n) if M runs in T(n) time, and for every $x \in \{0,1\}^*$,

$$\Pr[M(x) = L(x)] \geq 2/3.$$

- Definition. A language L is in BPTIME(T(n)) if there's PTM that decides L in O(T(n)) time.

- Definition. $\mathrm{BPP} = \bigcup_{c>0} \mathrm{BPTIME}(n^c)$.

- Clearly, $\mathrm{P} \subseteq \mathrm{BPP}$.

Remark. The defn of class BPP is robust. The class remains unaltered if we replace 2/3 by any constant ***strictly greater*** than (i.e., ***bounded away*** from) ½. We'll discuss this next.

Class BPP

- Definition. A PTM M *decides* a language L in time T(n) if M runs in T(n) time, and for every $x \in \{0,1\}^*$,

$$\Pr[M(x) = L(x)] \geq 2/3.$$

- Definition. A language L is in BPTIME(T(n)) if there's PTM that decides L in O(T(n)) time.

- Definition. $\text{BPP} = \bigcup_{c>0} \text{BPTIME}(n^c)$.

  Bounded-error **P**robabilistic **P**olynomial-time

- Clearly, $\text{P} \subseteq \text{BPP}$.

Remark. The defn of class BPP is robust. The class remains unaltered if we replace 2/3 by any constant ***strictly greater*** than (i.e., ***bounded away*** from) ½. We'll discuss this next.

Class BPP

- Definition. A PTM M *decides* a language L in time $T(n)$ if M runs in $T(n)$ time, and for every $x \in \{0,1\}^*$,

$$\Pr[M(x) = L(x)] \geq 2/3.$$

- Definition. A language L is in $\mathrm{BPTIME}(T(n))$ if there's PTM that decides L in $O(T(n))$ time.

- Definition. $\mathrm{BPP} = \bigcup_{c>0} \mathrm{BPTIME}(n^c)$.

- Clearly, $\mathrm{P} \subseteq \mathrm{BPP}$.

Remark. Achieving success probability ½ is trivial for any language. If we replace $\geq 2/3$ by $> ½$ then the corresponding class is called PP, which is (presumably) larger than BPP. More on PP later.

Error reduction for BPP

- Lemma. Let $c > 0$ be a constant. Suppose L is decided by a poly-time PTM M s.t. $\Pr[M(x) = L(x)] \geq \frac{1}{2} + |x|^{-c}$. Then, for every constant $d > 0$, L is decided by a poly-time PTM M' s.t. $\Pr[M'(x) = L(x)] \geq 1 - \exp(-|x|^d)$.

Error reduction for BPP

- Lemma. Let $c > 0$ be a constant. Suppose L is decided by a poly-time PTM M s.t. $\Pr[M(x) = L(x)] \geq \frac{1}{2} + |x|^{-c}$. Then, for every constant $d > 0$, L is decided by a poly-time PTM M' s.t. $\Pr[M'(x) = L(x)] \geq 1 - \exp(-|x|^d)$.
- *Proof.* Let $|x| = n$. Think of M' that runs M on input x for $m = 4n^{2c+d}$ times independently. Let $b_1, \ldots, b_m$ be the outputs of these independent executions of M. M' outputs $\mathrm{Majority}(b_1, \ldots, b_m)$.

Error reduction for BPP

- Lemma. Let $c > 0$ be a constant. Suppose L is decided by a poly-time PTM M s.t. $\Pr[M(x) = L(x)] \geq \frac{1}{2} + |x|^{-c}$. Then, for every constant $d > 0$, L is decided by a poly-time PTM M' s.t. $\Pr[M'(x) = L(x)] \geq 1 - \exp(-|x|^d)$.
- *Proof.* Let $|x| = n$ & $m = 4n^{2c+d}$. Let $y_i = 1$ if b_i is correct (i.e., $b_i = L(x)$), otherwise $y_i = 0$. Then M' outputs incorrectly only if $Y = y_1 + \ldots + y_m \leq m/2$.

Error reduction for BPP

- Lemma. Let $c > 0$ be a constant. Suppose L is decided by a poly-time PTM M s.t. $\Pr[M(x) = L(x)] \geq \frac{1}{2} + |x|^{-c}$. Then, for every constant $d > 0$, L is decided by a poly-time PTM M' s.t. $\Pr[M'(x) = L(x)] \geq 1 - \exp(-|x|^d)$.
- *Proof.* Let $|x| = n$ & $m = 4n^{2c+d}$. Let $y_i = 1$ if b_i is correct (i.e., $b_i = L(x)$), otherwise $y_i = 0$. Then M' outputs incorrectly only if $Y = y_1 + \ldots + y_m \leq m/2$.
- $E[y_i] = \Pr[y_i = 1] = \Pr[M(x) = L(x)] = p$ (say). It's given that $p \geq \frac{1}{2} + n^{-c}$. So, $\mu = E[Y] = mp \geq m/2.(1+2n^{-c})$.

Error reduction for BPP

- Lemma. Let $c > 0$ be a constant. Suppose L is decided by a poly-time PTM M s.t. $\Pr[M(x) = L(x)] \geq ½ + |x|^{-c}$. Then, for every constant $d > 0$, L is decided by a poly-time PTM M' s.t. $\Pr[M'(x) = L(x)] \geq 1 – \exp(-|x|^d)$.
- *Proof.* Let $|x| = n$ & $m = 4n^{2c+d}$. Let $y_i = 1$ if b_i is correct (i.e., $b_i = L(x)$), otherwise $y_i = 0$. Then M' outputs incorrectly only if $Y = y_1+\ldots+ y_m \leq m/2$.
- $E[y_i] = \Pr[y_i = 1] = \Pr[M(x) = L(x)] = p$ (say). It's given that $p \geq ½ + n^{-c}$. So, $\mu = E[Y] = mp \geq m/2.(1+2n^{-c})$.
- By Chernoff bound, $\Pr[Y \leq (1-\delta)\mu] \leq \exp(-(\delta^2\mu)/2)$, for any $\delta \in [0,1]$. We'll now fix the value of δ.

Error reduction for BPP

- Lemma. Let $c > 0$ be a constant. Suppose L is decided by a poly-time PTM M s.t. $\Pr[M(x) = L(x)] \geq \frac{1}{2} + |x|^{-c}$. Then, for every constant $d > 0$, L is decided by a poly-time PTM M' s.t. $\Pr[M'(x) = L(x)] \geq 1 - \exp(-|x|^d)$.
- *Proof.* $m = 4n^{2c+d}$, $p \geq \frac{1}{2} + n^{-c}$, $\mu = mp \geq m/2.(1+2n^{-c})$.
- $\Pr[Y \leq (1-\delta)\mu] \leq \exp(-(\delta^2\mu)/2)$, for any $\delta \in [0,1]$.
- M' outputs incorrectly only if $Y \leq m/2$.

Error reduction for BPP

- Lemma. Let $c > 0$ be a constant. Suppose L is decided by a poly-time PTM M s.t. $\Pr[M(x) = L(x)] \geq \frac{1}{2} + |x|^{-c}$. Then, for every constant $d > 0$, L is decided by a poly-time PTM M' s.t. $\Pr[M'(x) = L(x)] \geq 1 - \exp(-|x|^d)$.
- *Proof.* $m = 4n^{2c+d}$, $p \geq \frac{1}{2} + n^{-c}$, $\mu = mp \geq m/2.(1+2n^{-c})$.
- $\Pr[Y \leq (1-\delta)\mu] \leq \exp(-(\delta^2\mu)/2)$, for any $\delta \in [0,1]$.
- M' outputs incorrectly only if $Y \leq m/2$. If we choose δ s.t. $m/2 \leq (1-\delta)\mu$ then $\Pr[Y < m/2] \leq \Pr[Y \leq (1-\delta)\mu]$.

Error reduction for BPP

- Lemma. Let $c > 0$ be a constant. Suppose L is decided by a poly-time PTM M s.t. $\Pr[M(x) = L(x)] \geq ½ + |x|^{-c}$. Then, for every constant $d > 0$, L is decided by a poly-time PTM M' s.t. $\Pr[M'(x) = L(x)] \geq 1 - \exp(-|x|^d)$.
- *Proof.* $m = 4n^{2c+d}$, $p \geq ½ + n^{-c}$, $\mu = mp \geq m/2.(1+2n^{-c})$.
- $\Pr[Y \leq (1-\delta)\mu] \leq \exp(-(\delta^2\mu)/2)$, for any $\delta \in [0,1]$.
- M' outputs incorrectly only if $Y \leq m/2$. If we choose δ s.t. $m/2 \leq (1-\delta)\mu$ then $\Pr[Y < m/2] \leq \Pr[Y \leq (1-\delta)\mu]$.
- Picking $\delta \leq 2/(n^c+2)$ is sufficient. Set $\delta = n^{-c}$.

Error reduction for BPP

- Lemma. Let $c > 0$ be a constant. Suppose L is decided by a poly-time PTM M s.t. $\Pr[M(x) = L(x)] \geq \frac{1}{2} + |x|^{-c}$. Then, for every constant $d > 0$, L is decided by a poly-time PTM M' s.t. $\Pr[M'(x) = L(x)] \geq 1 - \exp(-|x|^d)$.
- *Proof.* $m = 4n^{2c+d}$, $p \geq \frac{1}{2} + n^{-c}$, $\boldsymbol{\mu} = mp \geq m/2.(1+2n^{-c})$.
- $\Pr[Y \leq (1-\boldsymbol{\delta})\boldsymbol{\mu}] \leq \exp(-(\boldsymbol{\delta}^2\boldsymbol{\mu})/2)$, and $\boldsymbol{\delta} = n^{-c}$.

- Therefore, $\Pr[M'(x) \neq L(x)] \leq \exp(-(\boldsymbol{\delta}^2\boldsymbol{\mu})/2)$,

Error reduction for BPP

- Lemma. Let $c > 0$ be a constant. Suppose L is decided by a poly-time PTM M s.t. $\Pr[M(x) = L(x)] \geq \frac{1}{2} + |x|^{-c}$. Then, for every constant $d > 0$, L is decided by a poly-time PTM M' s.t. $\Pr[M'(x) = L(x)] \geq 1 - \exp(-|x|^d)$.
- *Proof.* $m = 4n^{2c+d}$, $p \geq \frac{1}{2} + n^{-c}$, $\mu = mp \geq m/2.(1+2n^{-c})$.
- $\Pr[Y \leq (1-\boldsymbol{\delta})\mu] \leq \exp(-(\boldsymbol{\delta}^2\mu)/2)$, and $\boldsymbol{\delta} = n^{-c}$.

- Therefore, $\Pr[M'(x) \neq L(x)] \leq \exp(-(\boldsymbol{\delta}^2\mu)/2)$,

$$\leq \exp(-n^d).$$

Alternative definition of BPP

- Definition. A language L in BPP if there's a poly-time <u>DTM</u> $M(. , .)$ and a polynomial function $q(.)$ s.t. for every $x \in \{0,1\}^*$,

$$\Pr_{r \in_R \{0,1\}^{q(|x|)}} [M(x, r) = L(x)] \geq 2/3.$$

- $2/3$ can be replaced by $1 - \exp(-|x|^d)$ as before.

(Easy Homework)

Alternative definition of BPP

- Definition. A language L in BPP if there's a poly-time <u>DTM</u> $M(. , .)$ and a polynomial function $q(.)$ s.t. for every $x \in \{0,1\}^*$,

$$\Pr_{r \in_R \{0,1\}^{q(|x|)}} [M(x, r) = L(x)] \geq 2/3.$$

- Hence, $P \subseteq BPP \subseteq EXP$.

Alternative definition of BPP

- Definition. A language L in BPP if there's a poly-time DTM M(. , .) and a polynomial function q(.) s.t. for every $x \in \{0,1\}^*$,

$$\Pr_{r \in_R \{0,1\}^{q(|x|)}} [M(x, r) = L(x)] \geq 2/3.$$

- Hence, $P \subseteq BPP \subseteq EXP$.
- Sipser-Gacs-Lautemann. $BPP \subseteq \Sigma_2$. (We'll prove this)

Alternative definition of BPP

- Definition. A language L in BPP if there's a poly-time <u>DTM</u> M(. , .) and a polynomial function q(.) s.t. for every $x \in \{0,1\}^*$,

$$\Pr_{r \in_R \{0,1\}^{q(|x|)}} [M(x, r) = L(x)] \geq 2/3.$$

- Hence, $P \subseteq BPP \subseteq EXP$.
- Sipser-Gacs-Lautemann. $BPP \subseteq \Sigma_2$. (We'll prove this)
- How large is BPP? Is $NP \subseteq BPP$? i.e., is $SAT \in BPP$?

Alternative definition of BPP

- Definition. A language L in BPP if there's a poly-time DTM M(. , .) and a polynomial function q(.) s.t. for every $x \in \{0,1\}^*$,

$$\Pr_{r \in_R \{0,1\}^{q(|x|)}} [M(x, r) = L(x)] \geq 2/3.$$

- Hence, $P \subseteq BPP \subseteq EXP$.
- Sipser-Gacs-Lautemann. $BPP \subseteq \Sigma_2$. (We'll prove this)
- How large is BPP? Is $NP \subseteq BPP$? i.e., is $SAT \in BPP$?
- Next we show that $BPP \subseteq P/poly$. So, if $NP \subseteq BPP$ then $PH = \Sigma_2$. *(Karp-Lipton)*

Alternative definition of BPP

- Definition. A language L in BPP if there's a poly-time DTM M(. , .) and a polynomial function q(.) s.t. for every $x \in \{0,1\}^*$,

$$\Pr_{r \in_R \{0,1\}^{q(|x|)}} [M(x, r) = L(x)] \geq 2/3.$$

- Hence, $P \subseteq BPP \subseteq EXP$.
- Sipser-Gacs-Lautemann. $BPP \subseteq \Sigma_2$. (We'll prove this)

- Most complexity theorist believe that P = BPP! (More on this later.)

BPP is in P/poly

- Theorem. (*Adleman 1978*) BPP $\subseteq$ P/poly .
- *Proof.* Let $L \in$ BPP. Then, there's a poly-time DTM M and a polynomial function $q(.)$ s.t. for every $x \in \{0,1\}^*$,

$$\Pr_{r \in_R \{0,1\}^{q(|x|)}} [M(x, r) = L(x)] \geq 1 - 2^{-(|x|+1)} .$$

BPP is in P/poly

- Theorem. *(Adleman 1978)* $BPP \subseteq P/poly$.
- *Proof.* Let $L \in BPP$. Then, there's a poly-time <u>DTM</u> M and a polynomial function $q(.)$ s.t. for every $x \in \{0,1\}^*$,

$$Pr_{r \in_R \{0,1\}^{q(|x|)}} [M(x, r) = L(x)] \geq 1 - 2^{-(|x|+1)} .$$

- For every $x \in \{0,1\}^n$, at most $2^{-(n+1)}$ fraction of the r's are "bad". (<u>r is bad for x</u> if $M(x,r) \neq L(x)$).

BPP is in P/poly

- Theorem. *(Adleman 1978)* BPP $\subseteq$ P/poly .
- *Proof.* Let $L \in$ BPP. Then, there's a poly-time <u>DTM</u> M and a polynomial function $q(.)$ s.t. for every $x \in \{0,1\}^*$,

$$\Pr_{r \in_R \{0,1\}^{q(|x|)}} [M(x, r) = L(x)] \geq 1 - 2^{-(|x|+1)} .$$

- For every $x \in \{0,1\}^n$, at most $2^{-(n+1)}$ fraction of the r's are "bad". (r is bad for x if $M(x,r) \neq L(x)$).
- Summing over *all* $x \in \{0,1\}^n$, at most $2^n.2^{-(n+1)} = ½$ fraction of the r's are "bad" for some n-bit string x.

BPP is in P/poly

- Theorem. *(Adleman 1978)* BPP $\subseteq$ P/poly .
- *Proof.* Let $L \in$ BPP. Then, there's a poly-time <u>DTM</u> M and a polynomial function q(.) s.t. for every $x \in \{0,1\}^*$,

$$\Pr_{r \in_R \{0,1\}^{q(|x|)}} [M(x, r) = L(x)] \geq 1 - 2^{-(|x|+1)} .$$

- For every $x \in \{0,1\}^n$, at most $2^{-(n+1)}$ fraction of the r's are "bad". (r is bad for x if $M(x,r) \neq L(x)$).
- There's an $r_0 \in \{0,1\}^{q(n)}$ that is "good" for all $x \in \{0,1\}^n$, i.e., $M(x, r_0) = L(x)$ for all $x \in \{0,1\}^n$.

BPP is in P/poly

- Theorem. *(Adleman 1978)* BPP $\subseteq$ P/poly .
- *Proof.* Let L $\in$ BPP. Then, there's a poly-time <u>DTM</u> M and a polynomial function q(.) s.t. for every $x \in \{0,1\}^*$,

  $$\Pr_{r \in_R \{0,1\}^{q(|x|)}} [M(x, r) = L(x)] \geq 1 - 2^{-(|x|+1)} .$$

- For every $x \in \{0,1\}^n$, at most $2^{-(n+1)}$ fraction of the r's are "bad". (r is bad for x if $M(x,r) \neq L(x)$).
- There's an $r_0 \in \{0,1\}^{q(n)}$ that is "good" for all $x \in \{0,1\}^n$, i.e., $M(x, r_0) = L(x)$ for all $x \in \{0,1\}^n$.
- By hardwiring this r_0, the computation of $M(\cdot , r_0)$ can be viewed as a poly(n)-size circuit C. *(Cook-Levin)* ◼

Sipser-Gacs-Lautemann theorem

BPP is in PH

- We saw that $P \subseteq BPP \subseteq EXP$. But, is $BPP \subseteq NP$? Not known! (Yes, people still believe BPP = P.)

- Sipser showed $BPP \subseteq PH$, Gacs strengthened it to $BPP \subseteq \Sigma_2 \cap \Pi_2$, Lautemann gave a simpler proof.

- Theorem. *(Sipser-Gacs-Lautemann '83)* $BPP \subseteq \Sigma_2 \cap \Pi_2$.

BPP is in PH

- We saw that $P \subseteq BPP \subseteq EXP$. But, is $BPP \subseteq NP$? Not known! (Yes, people still believe BPP = P.)

- Sipser showed $BPP \subseteq PH$, Gacs strengthened it to $BPP \subseteq \Sigma_2 \cap \Pi_2$, Lautemann gave a simpler proof.

- Theorem. *(Sipser-Gacs-Lautemann '83)* $BPP \subseteq \Sigma_2 \cap \Pi_2$.
- *Proof.* Observe that BPP = co-BPP *(homework)*. So, it is sufficient to show $BPP \subseteq \Sigma_2$.

BPP is in PH

- Theorem. *(Sipser-Gacs-Lautemann '83)* BPP $\subseteq \Sigma_2$.
- *Proof.* Let $L \in$ BPP. Then, there's a poly-time DTM M and a polynomial function $q(.)$ s.t. for every $x \in \{0,1\}^*$,

$$\Pr_{r \in_R \{0,1\}^{q(|x|)}} [M(x, r) = L(x)] \geq 1 - 2^{-|x|} .$$

- Let $n = |x|$ and $m = q(n)$.

BPP is in PH

- Theorem. *(Sipser-Gacs-Lautemann '83)* BPP $\subseteq \Sigma_2$.
- *Proof.* Let $L \in$ BPP. Then, there's a poly-time <u>DTM</u> M and a polynomial function $q(.)$ s.t. for every $x \in \{0,1\}^*$,

$$\Pr_{r \in_R \{0,1\}^{q(|x|)}} [M(x, r) = L(x)] \geq 1 - 2^{-|x|} .$$

- Let $n = |x|$ and $m = q(n)$. Let $A_x \subseteq \{0,1\}^m$ such that $r \in A_x$ iff $M(x, r) = 1$. Observe that

$x \in L \Rightarrow |A_x| \geq (1 - 2^{-n}).2^m$ (A_x is large)

$x \notin L \Rightarrow |A_x| \leq 2^{-n}.2^m$ (A_x is small).

BPP is in PH

- Theorem. *(Sipser-Gacs-Lautemann '83)* BPP $\subseteq \Sigma_2$.
- *Proof.* Let $L \in$ BPP. Then, there's a poly-time <u>DTM</u> M and a polynomial function $q(.)$ s.t. for every $x \in \{0,1\}^*$,

$$\Pr_{r \in_R \{0,1\}^{q(|x|)}} [M(x, r) = L(x)] \geq 1 - 2^{-|x|} .$$

- Let $n = |x|$ and $m = q(n)$. Let $A_x \subseteq \{0,1\}^m$ such that $r \in A_x$ iff $M(x, r) = 1$. Observe that

$x \in L \Rightarrow |A_x| \geq (1 - 2^{-n}).2^m$ (A_x is large)

$x \notin L \Rightarrow |A_x| \leq 2^{-n}.2^m$ (A_x is small).

- Idea. If A_x is large then there exists a "small" collection $u_1, \ldots, u_k \in \{0,1\}^m$ s.t. $\bigcup_{i \in [k]} (A_x \oplus u_i) = \{0,1\}^m$.

bit-wise Xor

BPP is in PH

- Theorem. *(Sipser-Gacs-Lautemann '83)* BPP $\subseteq \Sigma_2$.
- *Proof.* Let $L \in$ BPP. Then, there's a poly-time DTM M and a polynomial function $q(.)$ s.t. for every $x \in \{0,1\}^*$,

$$\Pr_{r \in_R \{0,1\}^{q(|x|)}} [M(x, r) = L(x)] \geq 1 - 2^{-|x|} .$$

- Let $n = |x|$ and $m = q(n)$. Let $A_x \subseteq \{0,1\}^m$ such that $r \in A_x$ iff $M(x, r) = 1$. Observe that

$$x \in L \Rightarrow |A_x| \geq (1 - 2^{-n}).2^m \quad (A_x \text{ is large})$$

$$x \notin L \Rightarrow |A_x| \leq 2^{-n}.2^m \quad (A_x \text{ is small}).$$

- Idea. If A_x is large then there exists a "small" collection $u_1, \ldots, u_k \in \{0,1\}^m$ s.t. $\bigcup_{i \in [k]} (A_x \oplus u_i) = \{0,1\}^m$. No such collection exists if $|A_x|$ is small.

BPP is in PH

- Theorem. *(Sipser-Gacs-Lautemann '83)* BPP $\subseteq \Sigma_2$.
- *Proof.* Let $L \in$ BPP. Then, there's a poly-time <u>DTM</u> M and a polynomial function $q(.)$ s.t. for every $x \in \{0,1\}^*$,

$$\Pr_{r \in_R \{0,1\}^{q(|x|)}} [M(x, r) = L(x)] \geq 1 - 2^{-|x|} .$$

- Let $n = |x|$ and $m = q(n)$. Let $A_x \subseteq \{0,1\}^m$ such that $r \in A_x$ iff $M(x, r) = 1$. Observe that

$$x \in L \Rightarrow |A_x| \geq (1 - 2^{-n}).2^m \quad (A_x \text{ is large})$$

$$x \notin L \Rightarrow |A_x| \leq 2^{-n}.2^m \quad (A_x \text{ is small}).$$

- Idea. If A_x is large then there exists a "small" collection $u_1, \ldots, u_k \in \{0,1\}^m$ s.t. $\bigcup_{i \in [k]} (A_x \oplus u_i) = \{0,1\}^m$. Capture this property with a Σ_2 statement.

BPP is in PH

- Theorem. *(Sipser-Gacs-Lautemann '83)* BPP $\subseteq \Sigma_2$.
- *Proof.* $r \in A_x$ iff $M(x, r) = 1$. Then

  $x \in L \Rightarrow |A_x| \geq (1 - 2^{-n}).2^m$ (A_x is large)

  $x \notin L \Rightarrow |A_x| \leq 2^{-n}.2^m$ (A_x is small).
- Set $k = m/n + 1$.
- Obs. If $|A_x| \leq 2^{-n}.2^m$ then for *every* collection $u_1, \ldots, u_k \in \{0,1\}^m$, $\bigcup_{i \in [k]} (A_x \oplus u_i) \subsetneq \{0,1\}^m$.

BPP is in PH

- Theorem. *(Sipser-Gacs-Lautemann '83)* $BPP \subseteq \Sigma_2$.
- *Proof.* $r \in A_x$ iff $M(x, r) = 1$. Then

  $x \in L \Rightarrow |A_x| \geq (1 - 2^{-n}).2^m$ (A_x is large)

  $x \notin L \Rightarrow |A_x| \leq 2^{-n}.2^m$ (A_x is small).
- Set $k = m/n + 1$.
- Obs. If $|A_x| \leq 2^{-n}.2^m$ then for *<u>every</u>* collection $u_1, \ldots, u_k \in \{0,1\}^m$, $\bigcup_{i \in [k]} (A_x \oplus u_i) \subsetneq \{0,1\}^m$.
- *Proof.* As $|A_x| \leq 2^{-n}.2^m$, $|\bigcup_{i \in [k]} (A_x \oplus u_i)| \leq k.2^{m-n} < 2^m$ for sufficiently large n.

BPP is in PH

- Theorem. *(Sipser-Gacs-Lautemann '83)* BPP $\subseteq \Sigma_2$.
- *Proof.* $r \in A_x$ iff $M(x, r) = 1$. Then

  $x \in L \Rightarrow |A_x| \geq (1 - 2^{-n}).2^m$ (A_x is large)

  $x \notin L \Rightarrow |A_x| \leq 2^{-n}.2^m$ (A_x is small).
- Set $k = m/n + 1$.
- Claim. If $|A_x| \geq (1 - 2^{-n}).2^m$ then there *exists* a collection $u_1, \ldots, u_k \in \{0,1\}^m$ s.t. $\bigcup_{i\in[k]} (A_x \oplus u_i) = \{0,1\}^m$.

- Let us complete the proof of the theorem assuming the claim – we'll proof it shortly.

BPP is in PH

- Theorem. *(Sipser-Gacs-Lautemann '83)* $BPP \subseteq \Sigma_2$.
- *Proof.* $r \in A_x$ iff $M(x, r) = 1$. Then

$$x \in L \Rightarrow |A_x| \geq (1 - 2^{-n}).2^m \quad (A_x \text{ is large})$$

$$x \notin L \Rightarrow |A_x| \leq 2^{-n}.2^m \quad (A_x \text{ is small}).$$

- Set $k = m/n + 1$.
- Claim. If $|A_x| \geq (1 - 2^{-n}).2^m$ then there <u>*exists*</u> a collection $u_1, \ldots, u_k \in \{0,1\}^m$ s.t. $\bigcup_{i\in[k]} (A_x \oplus u_i) = \{0,1\}^m$.
- The observation and the claim imply the following:

$$x \in L \Rightarrow \exists u_1, \ldots, u_k \in \{0,1\}^m \quad \bigcup_{i\in[k]} (A_x \oplus u_i) = \{0,1\}^m$$

$$x \notin L \Rightarrow \forall u_1, \ldots, u_k \in \{0,1\}^m \quad \bigcup_{i\in[k]} (A_x \oplus u_i) \subsetneq \{0,1\}^m.$$

BPP is in PH

- Theorem. *(Sipser-Gacs-Lautemann '83)* BPP $\subseteq \Sigma_2$.
- *Proof.* $r \in A_x$ iff $M(x, r) = 1$. Then

  $x \in L \Rightarrow |A_x| \geq (1 - 2^{-n}).2^m$ (A_x is large)

  $x \notin L \Rightarrow |A_x| \leq 2^{-n}.2^m$ (A_x is small).
- Set $k = m/n + 1$.
- Claim. If $|A_x| \geq (1 - 2^{-n}).2^m$ then there <u>*exists*</u> a collection $u_1, \ldots, u_k \in \{0,1\}^m$ s.t. $\bigcup_{i\in[k]} (A_x \oplus u_i) = \{0,1\}^m$.
- The observation and the claim imply the following:

  $x \in L \Leftrightarrow \exists u_1, \ldots, u_k \in \{0,1\}^m \quad \bigcup_{i\in[k]} (A_x \oplus u_i) = \{0,1\}^m$.

BPP is in PH

- Theorem. *(Sipser-Gacs-Lautemann '83)* BPP $\subseteq \Sigma_2$.
- *Proof.* $r \in A_x$ iff $M(x, r) = 1$. Set $k = m/n + 1$.

$x \in L \Leftrightarrow \exists u_1, \ldots, u_k \in \{0,1\}^m \quad \bigcup_{i \in [k]} (A_x \oplus u_i) = \{0,1\}^m$

BPP is in PH

- Theorem. *(Sipser-Gacs-Lautemann '83)* BPP $\subseteq \Sigma_2$.
- *Proof.* $r \in A_x$ iff $M(x, r) = 1$. Set $k = m/n + 1$.

$x \in L \Leftrightarrow \exists u_1, \ldots, u_k \in \{0,1\}^m \quad \bigcup_{i \in [k]} (A_x \oplus u_i) = \{0,1\}^m$

$x \in L \Leftrightarrow \exists u_1, \ldots, u_k \in \{0,1\}^m \; \forall r \in \{0,1\}^m \quad r \in \bigcup_{i \in [k]} (A_x \oplus u_i)$

BPP is in PH

- Theorem. *(Sipser-Gacs-Lautemann '83)* BPP $\subseteq \Sigma_2$.
- *Proof.* $r \in A_x$ iff $M(x, r) = 1$. Set $k = m/n + 1$.

$x \in L \Leftrightarrow \exists u_1, \ldots, u_k \in \{0,1\}^m \quad \bigcup_{i \in [k]} (A_x \oplus u_i) = \{0,1\}^m$

$x \in L \Leftrightarrow \exists u_1, \ldots, u_k \in \{0,1\}^m \ \forall r \in \{0,1\}^m \ \ r \in \bigcup_{i \in [k]} (A_x \oplus u_i)$

$x \in L \Leftrightarrow \exists u_1, \ldots, u_k \in \{0,1\}^m \ \forall r \in \{0,1\}^m \ \bigvee_{i \in [k]} [r \oplus u_i \in A_x]$

BPP is in PH

- Theorem. *(Sipser-Gacs-Lautemann '83)* $BPP \subseteq \Sigma_2$.
- *Proof.* $r \in A_x$ iff $M(x, r) = 1$. Set $k = m/n + 1$.

$x \in L \Leftrightarrow \exists u_1, \ldots, u_k \in \{0,1\}^m \quad \bigcup_{i \in [k]} (A_x \oplus u_i) = \{0,1\}^m$

$x \in L \Leftrightarrow \exists u_1, \ldots, u_k \in \{0,1\}^m \; \forall r \in \{0,1\}^m \quad r \in \bigcup_{i \in [k]} (A_x \oplus u_i)$

$x \in L \Leftrightarrow \exists u_1, \ldots, u_k \in \{0,1\}^m \; \forall r \in \{0,1\}^m \quad \bigvee_{i \in [k]} [r \oplus u_i \in A_x]$

$x \in L \Leftrightarrow \exists u_1, \ldots, u_k \in \{0,1\}^m \; \forall r \in \{0,1\}^m \quad \bigvee_{i \in [k]} M(x, r \oplus u_i) = 1$

BPP is in PH

- Theorem. *(Sipser-Gacs-Lautemann '83)* BPP $\subseteq \Sigma_2$.
- *Proof.* $r \in A_x$ iff $M(x, r) = 1$. Set $k = m/n + 1$.

$$x \in L \Leftrightarrow \exists u_1, \ldots, u_k \in \{0,1\}^m \quad \bigcup_{i \in [k]} (A_x \oplus u_i) = \{0,1\}^m$$

$$x \in L \Leftrightarrow \exists u_1, \ldots, u_k \in \{0,1\}^m \ \forall r \in \{0,1\}^m \ \ r \in \bigcup_{i \in [k]} (A_x \oplus u_i)$$

$$x \in L \Leftrightarrow \exists u_1, \ldots, u_k \in \{0,1\}^m \ \forall r \in \{0,1\}^m \ \bigvee_{i \in [k]} [r \oplus u_i \in A_x]$$

$$x \in L \Leftrightarrow \exists u_1, \ldots, u_k \in \{0,1\}^m \ \forall r \in \{0,1\}^m \ \bigvee_{i \in [k]} M(x, r \oplus u_i) = 1$$

- Think of a DTM N that takes input $x, u_1, \ldots, u_m, r$, and outputs 1 iff $M(x, r \oplus u_i) = 1$ for some $i \in [k]$. Observe that N is a poly-time DTM.

BPP is in PH

- Theorem. *(Sipser-Gacs-Lautemann '83)* BPP $\subseteq \Sigma_2$.
- *Proof.* $r \in A_x$ iff $M(x, r) = 1$. Set $k = m/n + 1$.

$x \in L \Leftrightarrow \exists u_1, \ldots, u_k \in \{0,1\}^m \quad \bigcup_{i \in [k]} (A_x \oplus u_i) = \{0,1\}^m$

$x \in L \Leftrightarrow \exists u_1, \ldots, u_k \in \{0,1\}^m \; \forall r \in \{0,1\}^m \quad r \in \bigcup_{i \in [k]} (A_x \oplus u_i)$

$x \in L \Leftrightarrow \exists u_1, \ldots, u_k \in \{0,1\}^m \; \forall r \in \{0,1\}^m \quad \bigvee_{i \in [k]} [r \oplus u_i \in A_x]$

$x \in L \Leftrightarrow \exists u_1, \ldots, u_k \in \{0,1\}^m \; \forall r \in \{0,1\}^m \quad N(x, \underline{\mathbf{u}}, r) = 1.$

$\underline{\mathbf{u}} = \{u_1, \ldots, u_k\}$

- Therefore, $L \in \Sigma_2$.

Proof of the Claim

- Claim. If $|A_x| \geq (1 - 2^{-n}).2^m$ then there ***exists*** a collection $u_1, \ldots, u_k \in \{0,1\}^m$ s.t. $\bigcup_{i \in [k]} (A_x \oplus u_i) = \{0,1\}^m$.
- *Proof.* The proof of this uses the probabilistic method.

Proof of the Claim

- Claim. If $|A_x| \geq (1 - 2^{-n}).2^m$ then there ***exists*** a collection $u_1, \ldots, u_k \in \{0,1\}^m$ s.t. $\bigcup_{i\in[k]} (A_x \oplus u_i) = \{0,1\}^m$.
- *Proof.* We'll show if $u_1, \ldots, u_k$ are picked independently and uniformly at random then

$$\Pr_{\underline{u}} [\forall r \in \{0,1\}^m \;\; r \in \bigcup_{i\in[k]} (A_x \oplus u_i)] > 0 .$$

Proof of the Claim

- Claim. If $|A_x| \geq (1 - 2^{-n}).2^m$ then there ***exists*** a collection $u_1, \ldots, u_k \in \{0,1\}^m$ s.t. $\bigcup_{i \in [k]} (A_x \oplus u_i) = \{0,1\}^m$.
- *Proof.* We'll show if $u_1, \ldots, u_k$ are picked independently and uniformly at random then

$$\Pr_{\underline{\mathbf{u}}} [\exists r \in \{0,1\}^m \ \ r \notin \bigcup_{i \in [k]} (A_x \oplus u_i)] < 1 .$$

Proof of the Claim

- Claim. If $|A_x| \geq (1 - 2^{-n}).2^m$ then there ***exists*** a collection $u_1, \ldots, u_k \in \{0,1\}^m$ s.t. $\bigcup_{i\in[k]} (A_x \oplus u_i) = \{0,1\}^m$.
- *Proof.* We'll show if $u_1, \ldots, u_k$ are picked independently and uniformly at random then

$$\Pr_{\mathbf{u}} [\exists r \in \{0,1\}^m \;\; r \notin (A_x \oplus u_i) \text{ for every } i \in [k]] < 1 .$$

Proof of the Claim

- Claim. If $|A_x| \geq (1 - 2^{-n}).2^m$ then there ***exists*** a collection $u_1, \ldots, u_k \in \{0,1\}^m$ s.t. $\bigcup_{i\in[k]} (A_x \oplus u_i) = \{0,1\}^m$.
- *Proof.* We'll show if $u_1, \ldots, u_k$ are picked independently and uniformly at random then

$$\Pr_{\underline{\mathbf{u}}} [\exists r \in \{0,1\}^m \;\; r \oplus u_i \notin A_x \text{ for every } i \in [k]] < 1 .$$

Proof of the Claim

- Claim. If $|A_x| \geq (1 - 2^{-n}).2^m$ then there *exists* a collection $u_1, \ldots, u_k \in \{0,1\}^m$ s.t. $\bigcup_{i\in[k]} (A_x \oplus u_i) = \{0,1\}^m$.
- *Proof.* We'll show if $u_1, \ldots, u_k$ are picked independently and uniformly at random then

$$\Pr_{\mathbf{u}} [\exists r\in\{0,1\}^m \ \ r \oplus u_i \notin A_x \text{ for every } i\in [k]] < 1 .$$

- Fix an $r\in\{0,1\}^m$ (we'll apply a union bound later). Fix an $i\in [k]$. Then, $\Pr_{\mathbf{u}} [r \oplus u_i \notin A_x] \leq 2^{-n}$.

Distributed uniformly inside $\{0,1\}^m$ as r is fixed and u_i is picked uniformly at random from $\{0,1\}^m$.

Proof of the Claim

- Claim. If $|A_x| \geq (1 - 2^{-n}).2^m$ then there ***exists*** a collection $u_1, \ldots, u_k \in \{0,1\}^m$ s.t. $\bigcup_{i\in[k]} (A_x \oplus u_i) = \{0,1\}^m$.
- *Proof.* We'll show if $u_1, \ldots, u_k$ are picked independently and uniformly at random then

  $\Pr_{\underline{u}} [\exists r \in \{0,1\}^m \;\; r \oplus u_i \notin A_x \text{ for every } i \in [k]] < 1$.
- Fix an $r \in \{0,1\}^m$ (we'll apply a union bound later). Fix an $i \in [k]$. Then, $\Pr_{\underline{u}} [r \oplus u_i \notin A_x] \leq 2^{-n}$. As $u_1, \ldots, u_k$ are independent, $\Pr_{\underline{u}} [r \oplus u_i \notin A_x$ for every $i \in [k]] \leq 2^{-kn}$.

Proof of the Claim

- Claim. If $|A_x| \geq (1 - 2^{-n}).2^m$ then there ***exists*** a collection $u_1, \ldots, u_k \in \{0,1\}^m$ s.t. $\bigcup_{i \in [k]} (A_x \oplus u_i) = \{0,1\}^m$.
- *Proof.* We'll show if $u_1, \ldots, u_k$ are picked independently and uniformly at random then

  $Pr_{\underline{\mathbf{u}}} [\exists r \in \{0,1\}^m \;\; r \oplus u_i \notin A_x$ for every $i \in [k]] < 1$.
- Fix an $r \in \{0,1\}^m$ (we'll apply a union bound later). Fix an $i \in [k]$. Then, $Pr_{\underline{\mathbf{u}}} [r \oplus u_i \notin A_x] \leq 2^{-n}$. As $u_1, \ldots, u_k$ are independent, $Pr_{\underline{\mathbf{u}}} [r \oplus u_i \notin A_x$ for every $i \in [k]] < 2^{-m}$.

$k = m/n + 1$

Proof of the Claim

- Claim. If $|A_x| \geq (1 - 2^{-n}).2^m$ then there ***exists*** a collection $u_1, \ldots, u_k \in \{0,1\}^m$ s.t. $\bigcup_{i\in[k]} (A_x \oplus u_i) = \{0,1\}^m$.
- *Proof.* We'll show if $u_1, \ldots, u_k$ are picked independently and uniformly at random then

  $$\Pr_{\mathbf{u}} [\exists r\in\{0,1\}^m \;\; r \oplus u_i \notin A_x \text{ for every } i\in [k]] < 1 .$$

- Fix an $r\in\{0,1\}^m$ (we'll apply a union bound later). Fix an $i\in [k]$. Then, $\Pr_{\mathbf{u}} [r \oplus u_i \notin A_x] \leq 2^{-n}$. As $u_1, \ldots, u_k$ are independent, $\Pr_{\mathbf{u}} [r \oplus u_i \notin A_x$ for every $i\in [k]] < 2^{-m}$.
- Applying union bound,

  $$\Pr_{\mathbf{u}} [\exists r\in\{0,1\}^m \;\; r \oplus u_i \notin A_x \text{ for every } i\in [k]] < 2^m 2^{-m}$$

Proof of the Claim

- Claim. If $|A_x| \geq (1 - 2^{-n}).2^m$ then there *exists* a collection $u_1, \ldots, u_k \in \{0,1\}^m$ s.t. $\bigcup_{i\in[k]} (A_x \oplus u_i) = \{0,1\}^m$.
- *Proof.* We'll show if $u_1, \ldots, u_k$ are picked independently and uniformly at random then

  $\Pr_{\underline{u}} [\exists r \in \{0,1\}^m \ \ r \oplus u_i \notin A_x$ for every $i \in [k]] < 1$.
- Fix an $r \in \{0,1\}^m$ (we'll apply a union bound later). Fix an $i \in [k]$. Then, $\Pr_{\underline{u}} [r \oplus u_i \notin A_x] \leq 2^{-n}$. As $u_1, \ldots, u_k$ are independent, $\Pr_{\underline{u}} [r \oplus u_i \notin A_x$ for every $i \in [k]] < 2^{-m}$.
- Applying union bound,

  $\Pr_{\underline{u}} [\exists r \in \{0,1\}^m \ \ r \oplus u_i \notin A_x$ for every $i \in [k]] < 1$.

Complete derandomization of BPP ?

- Can the Sipser-Gacs-Lautemann theorem be strengthened? How low in the PH does BPP lie ?

- Theorem. *(Nisan & Wigderson 1988,…, Umans 2003)* If there's a $L \in \mathrm{DTIME}(2^{O(n)})$ and a constant $\varepsilon > 0$ such that any circuit C_n that decides $L \cap \{0,1\}^n$ requires size $2^{\varepsilon n}$, then BPP = P .

- Lower bounds ➡ Derandomization !

Complete derandomization of BPP ?

- Can the Sipser-Gacs-Lautemann theorem be strengthened? How low in the PH does BPP lie ?

- Theorem. *(Nisan & Wigderson 1988,…, Umans 2003)* If there's a $L \in \mathrm{DTIME}(2^{O(n)})$ and a constant $\varepsilon > 0$ such that any circuit C_n that decides $L \cap \{0,1\}^n$ requires size $2^{\varepsilon n}$, then BPP = P .

- Lower bounds ➡ Derandomization !
- Caution: Shouldn't interpret this result as "randomness is useless".

Classes RP, co-RP and ZPP

Class RP

- Class RP is the <u>one-sided error</u> version of BPP.

- Definition. A language L is in RTIME(T(n)) if there's a PTM M that decides L in O(T(n)) time such that

  $x \in L \Rightarrow \Pr[M(x) = 1] \geq 2/3$

  $x \notin L \Rightarrow \Pr[M(x) = 0] = 1.$

- Definition. $\mathrm{RP} = \bigcup_{c > 0} \mathrm{RTIME}(n^c)$.

- Clearly, $\mathrm{RP} \subseteq \mathrm{BPP}$.

Class RP

- Class RP is the one-sided error version of BPP.

- **Definition.** A language L is in $\text{RTIME}(T(n))$ if there's a PTM M that decides L in $O(T(n))$ time such that

$$x \in L \Rightarrow \Pr[M(x) = 1] \geq 2/3$$

$$x \notin L \Rightarrow \Pr[M(x) = 0] = 1.$$

Remark. The defn of class RP is robust. The class remains unaltered if we replace $2/3$ by $|x|^{-c}$ for any constant $c > 0$. The succ. prob. can then be amplified to $1\text{-}\exp(-|x|^d)$.

(Easy Homework)

- **Definition.** $\text{RP} = \bigcup_{c > 0} \text{RTIME}(n^c)$.

  RP → **R**andomized **P**oly-time.

- Clearly, $\text{RP} \subseteq \text{BPP}$.

Class RP

- Class RP is the one-sided error version of BPP.

- Definition. A language L is in $\mathrm{RTIME}(T(n))$ if there's a PTM M that decides L in $O(T(n))$ time such that

$$x \in L \implies \Pr[M(x) = 1] \geq 2/3$$

$$x \notin L \implies \Pr[M(x) = 0] = 1.$$

- Definition. $\mathrm{RP} = \bigcup_{c > 0} \mathrm{RTIME}(n^c)$.

- Clearly, $\mathrm{RP} \subseteq \mathrm{BPP}$. Obs. $\mathrm{RP} \subseteq \mathrm{NP}$. *(Easy Homework)*

Recall, we don't know whether $\mathrm{BPP} \subseteq \mathrm{NP}$.

Class co-RP

- Definition. co-RP = $\{L : \overline{L} \in$ RP$\}$.

- Obs. A language L is in co-RP if there's a PTM M that decides L in poly-time such that

$$x \in L \implies \Pr[M(x) = 1] = 1$$
$$x \notin L \implies \Pr[M(x) = 0] \geq 2/3.$$

- Obs. co-RP $\subseteq$ BPP .

Class co-RP

- Definition. co-RP = $\{L : \overline{L} \in \text{RP}\}$.

- Obs. A language L is in co-RP if there's a PTM M that decides L in poly-time such that

$$x \in L \Rightarrow \Pr[M(x) = 1] = 1$$

$$x \notin L \Rightarrow \Pr[M(x) = 0] \geq 2/3.$$

- Obs. co-RP $\subseteq$ BPP .

- Is RP$\cap$co-RP in P ? Not known!

Class ZPP

- Recall that PTMs are allowed to not halt on some computation paths defined by its random choices.
- We say that a PTM M has *expected running time* $T(n)$ if the expected running time of M on input x is at most $T(n)$ for all $x \in \{0,1\}^n$.

Class ZPP

- Recall that PTMs are allowed to not halt on some computation paths defined by its random choices.
- We say that a PTM M has *expected running time* T(n) if the expected running time of M on input x is at most T(n) for all $x \in \{0,1\}^n$.

- Definition. A language L is in ZTIME(T(n)) if there's a PTM M s.t. on every input x, M(x) = L(x) whenever M halts, and M has expected running time O(T(n)).
- Definition. $\text{ZPP} = \bigcup_{c>0} \text{ZTIME}(n^c)$.

Zero-error **P**robabilistic **P**oly-time.

Class ZPP

- Definition. A language L is in $ZTIME(T(n))$ if there's a PTM M s.t. on every input x, $M(x) = L(x)$ whenever M halts, and M has expected running time $O(T(n))$.
- Definition. $ZPP = \bigcup_{c>0} ZTIME(n^c)$.
- Problems in ZPP are said to have poly-time *Las Vegas algorithms*, whereas those in BPP are said to have poly-time *Monte-Carlo algorithms*.

- Theorem. $ZPP = RP \cap co\text{-}RP \subseteq BPP$. *(Homework)*

- Note. If $P = BPP$ then $P = ZPP = BPP$.

Why truly random bits?

- A PTM is defined using truly random bits. Is the definition sufficiently powerful? Do *biased* random bits give any additional computational power?

Why truly random bits?

- A PTM is defined using truly random bits. Is the definition sufficiently powerful? Do *biased* random bits give any additional computational power?

- Claim. A random bit with $Pr[1] = p$ can be simulated by a PTM in expected $O(1)$ time if the i-th bit of p can be computed in *poly*(i) time. *(Homework)*

- There's a p and a PTM M with access to p-biased random bits s.t. M decides an undecidable language!

Why truly random bits?

- On the other hand, we can obtain truly random bits from biased random bits.

- Claim. *(von-Neumann 1951)* A truly random bit can be simulated by a PTM with access to p-biased random bits in expected $O(p^{-1}(1-p)^{-1})$ time. *(Homework)*