Computational Complexity Theory

Lecture 18: Sipser-Gacs-Lautemann theorem; Classes RP and ZPP; Perfect matching in RNC

Department of Computer Science,
Indian Institute of Science

Recap: Probabilistic Turing Machines

- Definition. A *probabilistic Turing machine* (PTM) M has <u>two</u> 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. But, semantically, they are quite different

Recap: 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.

Recap: 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 $\mathrm{BPTIME}(T(n))$ if there's PTM that decides L in $O(T(n))$ time.

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

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

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

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

- 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) $\frac{1}{2}$. We'll discuss this next.

Recap: 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. $\Pr[M'(x) = L(x)] \geq 1 - \exp(-|x|^d)$.

Recap: 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 $x \in \{0,1\}^*$,

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

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

Recap: 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 $SAT \in BPP$?
- Theorem. (*Adleman 1978*) $BPP \subseteq P/poly$.
- So, if $NP \subseteq BPP$ then $PH = \Sigma_2$. *(Karp-Lipton)*

Recap: 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.)

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)* $\mathrm{BPP} \subseteq \Sigma_2$.
- *Proof.* Let $L \in \mathrm{BPP}$. 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

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

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

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. $\bigcup_{i \in [k]} (A_x \oplus u_i) = \{0,1\}^m$.

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 <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. $\bigcup_{i \in [k]} (A_x \oplus u_i) = \{0,1\}^m$. 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 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. $\bigcup_{i \in [k]} (A_x \oplus u_i) = \{0,1\}^m$. 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 $\boxed{k = m/n + 1}$.
- Obs. If $|A_x| \leq 2^{-n}.2^m$ then for <u>*every*</u> collection $u_1, \ldots, u_k \in \{0,1\}^m$, $\bigcup_{i\in[k]} (A_x \oplus u_i) \subsetneq \{0,1\}^m$.

BPP is in PH

- Theorem. *(Sipser-Gacs-Lautemann '83)* $\mathrm{BPP} \subseteq \Sigma_2$.
- *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

  $x \in L \Rightarrow |A_x| \geq (1 - 2^{-n}).2^m$ (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. $\bigcup_{i\in[k]} (A_x \oplus u_i) = \{0,1\}^m$.
- 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

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

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

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

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

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

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

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

- 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.* $\boxed{r \in A_x \text{ iff } M(x, r) = 1.}$ 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

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

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

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

Proof of the Claim

- Claim. If $|A_x| \geq (1 - 2^{-n}).2^m$ then there <u>*exists*</u> a collection $u_1, \ldots, u_k \in \{0,1\}^m$ s.t. $\bigcup_{i \in [k]} (A_x \oplus u_i) = \{0,1\}^m$.
- *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 $\boxed{|A_x| \geq (1 - 2^{-n}).2^m}$ 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, $\Pr_{\underline{u}} [\underbrace{r \oplus u_i} \notin A_x] \leq 2^{-n}$.

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

  $\Pr_{\underline{u}} [\exists r \in \{0,1\}^m \ \ r \oplus u_i \notin A_x$ 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 <u>*exists*</u> a collection $u_1, \ldots, u_k \in \{0,1\}^m$ s.t. $\bigcup_{i\in[k]} (A_x \oplus u_i) = \{0,1\}^m$.
- *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, $\Pr_{\underline{u}} [r \oplus u_i \notin A_x \text{ for every } i \in [k]] < 2^{-m}$.

$\boxed{k = m/n + 1}$

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

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

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 $\Rightarrow$ 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 $\mathrm{BPP} = \mathrm{P}$.

- 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

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

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

- Clearly, $RP \subseteq BPP$.

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

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

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

Randomized **P**oly-time.

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

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

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

Recall, we don't know whether BPP $\subseteq$ 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

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

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

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

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

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

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

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

Class ZPP

- Recall that PTMs are allowed to <u>not</u> 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 <u>not</u> 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 $\mathrm{ZTIME}(T(n))$ 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 $= \bigcup_{c>0} \mathrm{ZTIME}(n^c)$.

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

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

Perfect Matching in RNC

Randomness brings in simplicity

- The use of randomness helps in designing *simple* and efficient algorithms for many problems.
- We'll see one such algorithm in this lecture, namely an efficient randomized, <u>parallel</u> algorithm to check if a given bipartite graph has a perfect matching.

Class RNC

- The use of randomness helps in designing *simple* and efficient algorithms for many problems.
- We'll see one such algorithm in this lecture, namely an efficient randomized, parallel algorithm to check if a given bipartite graph has a perfect matching.
- Definition. A language L is in RNC^i if there's a *randomized* $O((\log n)^i)$-time parallel algorithm M that uses $n^{O(1)}$ parallel processors s.t. for every $x \in \{0,1\}^*$,

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

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

Here, n is the input length.

Class RNC

- The use of randomness helps in designing *simple* and efficient algorithms for many problems.
- We'll see one such algorithm in this lecture, namely an efficient randomized, parallel algorithm to check if a given bipartite graph has a perfect matching.

- Definition. $\mathrm{RNC} = \bigcup_{i > 0} \mathrm{RNC}^i$.

- RNC stands for Randomized NC. We can alternatively define RNC using (uniform) circuits.

Perfect matching in RNC

- Let PerfectMatching = {Bipartite graph G : G has a perfect matching}.
- Theorem. *(Lovasz 1979)* PerfectMatching $\in$ RNC^2.
- The input $G = (L \dot{\cup} R, E)$ is given as a $n \times n$ *biadjacency matrix* $A = (a_{ij})_{i,j \in n}$, where $n = |L| = |R|$.

Perfect matching in RNC

- Let PerfectMatching = {Bipartite graph G : G has a perfect matching}.
- Theorem. *(Lovasz 1979)* PerfectMatching $\in$ RNC^2.
- The input $G = (L \dot\cup R, E)$ is given as a $n \times n$ *biadjacency matrix* $A = (a_{ij})_{i,j \in n}$, where $n = |L| = |R|$.

$a_{ij} = 1$ if there's an edge from the i-th vertex in L to the j-th vertex in R, otherwise $a_{ij} = 0$.

Perfect matching in RNC

- Let PerfectMatching = {Bipartite graph G : G has a perfect matching}.
- Theorem. *(Lovasz 1979)* PerfectMatching $\in$ RNC^2.
- The input $G = (L \dot\cup R, E)$ is given as a $n \times n$ *biadjacency matrix* $A = (a_{ij})_{i,j\in n}$, where $n = |L| = |R|$.
- Algorithm.

1. Construct $B = (b_{ij})_{i,j\in n}$ as follows: If $a_{ij}=0$, then $b_{ij}=0$. Else, pick b_{ij} independently and uniformly <u>at random</u> from $[2n]$.
2. Compute $\det(B)$.
3. If $\det(B) \neq 0$ output "yes", else output "no".

Perfect matching in RNC

- Let PerfectMatching = {Bipartite graph G : G has a perfect matching}.
- Theorem. *(Lovasz 1979)* PerfectMatching $\in$ RNC^2.
- The input $G = (L \dot\cup R, E)$ is given as a $n \times n$ *biadjacency matrix* $A = (a_{ij})_{i,j\in n}$, where $n = |L| = |R|$.
- Algorithm. (RNC^2 algorithm)

1. Construct $B = (b_{ij})_{i,j\in n}$ as follows: If a_{ij}=0, then b_{ij}=0. Else, pick b_{ij} independently and uniformly <u>at random</u> from $[2n]$. (This can be done using n^2 processors.)
2. Compute $\det(B)$. (determinant is in NC^2, *Csanky '76*)
3. If $\det(B) \neq 0$ output "yes", else output "no".

Perfect matching in RNC

- Let PerfectMatching = {Bipartite graph G : G has a perfect matching}.
- Theorem. *(Lovasz 1979)* PerfectMatching $\in$ RNC^2.
- Correctness of the Algorithm.

1. Define $X = (x_{ij})_{i,j\in n}$ as follows: If $a_{ij}=0$, then $x_{ij}=0$. Else, x_{ij} is a formal variable.
2. $\det(X) = \sum_{\sigma\in S_n} (-1)^{\mathrm{sign}(\sigma)} \prod_{i\in[n]} x_{i\,\sigma(i)}$.

- S_n is the set of all permutations on $[n]$.

Perfect matching in RNC

- Let PerfectMatching = {Bipartite graph G : G has a perfect matching}.
- Theorem. *(Lovasz 1979)* PerfectMatching ∈ RNC^2.
- Correctness of the Algorithm.

1. Define $X = (x_{ij})_{i,j\in n}$ as follows: If $a_{ij}=0$, then $x_{ij}=0$. Else, x_{ij} is a formal variable.
2. $\det(X) = \sum_{\sigma\in S_n} (-1)^{\mathrm{sign}(\sigma)} \prod_{i\in[n]} x_{i\,\sigma(i)}$.

- Obs. $\det(X) \neq 0 \iff$ G has a perfect matching.

Perfect matching in RNC

- Let PerfectMatching = {Bipartite graph G : G has a perfect matching}.
- Theorem. *(Lovasz 1979)* PerfectMatching $\in$ RNC^2.
- Correctness of the Algorithm.

1. Define $X = (x_{ij})_{i,j\in n}$ as follows: If $a_{ij}=0$, then $x_{ij}=0$. Else, x_{ij} is a formal variable.
2. $\det(X) = \sum_{\sigma\in S_n} (-1)^{\mathrm{sign}(\sigma)} \prod_{i\in[n]} x_{i\,\sigma(i)}$.

- Obs. $\det(X) \neq 0 \iff$ G has a perfect matching.

Polynomial in the x_{ij} variables.

Perfect matching in RNC

- Let PerfectMatching = {Bipartite graph G : G has a perfect matching}.
- Theorem. *(Lovasz 1979)* PerfectMatching $\in$ RNC^2.
- Correctness of the Algorithm.

1. Define $X = (x_{ij})_{i,j \in n}$ as follows: If $a_{ij}=0$, then $x_{ij}=0$. Else, x_{ij} is a formal variable.
2. $\det(X) = \sum_{\sigma \in S_n} (-1)^{\mathrm{sign}(\sigma)} \prod_{i \in [n]} x_{i\,\sigma(i)}$.

- Obs. $\det(X) \neq 0$ $\Longleftrightarrow$ G has a perfect matching.
- In the algorithm, we set $x_{ij} = b_{ij}$, where b_{ij} is picked randomly from $[2n]$ if $x_{ij} \neq 0$, otherwise $b_{ij} = 0$.

Perfect matching in RNC

- Let PerfectMatching = {Bipartite graph G : G has a perfect matching}.
- Theorem. *(Lovasz 1979)* PerfectMatching $\in$ RNC^2.
- Correctness of the Algorithm.

1. Define $X = (x_{ij})_{i,j\in n}$ as follows: If $a_{ij}=0$, then $x_{ij}=0$. Else, x_{ij} is a formal variable.
2. $\det(X) = \sum_{\sigma\in S_n} (-1)^{\mathrm{sign}(\sigma)} \prod_{i\in[n]} x_{i\,\sigma(i)}$.

- Obs. $\det(X) \neq 0$ $\Longleftrightarrow$ G has a perfect matching.
- If $\det(X) = 0$ then $\det(B) = 0$. (So, the algorithm has one-sided error.)

Perfect matching in RNC

- Let PerfectMatching = {Bipartite graph G : G has a perfect matching}.
- Theorem. *(Lovasz 1979)* PerfectMatching $\in$ RNC^2.
- Correctness of the Algorithm.

1. Define $X = (x_{ij})_{i,j\in n}$ as follows: If $a_{ij}=0$, then $x_{ij}=0$. Else, x_{ij} is a formal variable.
2. $$\det(X) = \sum_{\sigma\in S_n} (-1)^{sign(\sigma)} \prod_{i\in[n]} x_{i\,\sigma(i)} \,.$$

- Obs. $\det(X) \neq 0$ $\Longleftrightarrow$ G has a perfect matching.
- If $\det(X) \neq 0$, what is the probability that $\det(B) \neq 0$?

The answer is given by the **Schwartz-Zippel lemma**

Schwartz-Zippel lemma

- Lemma. *(Schwartz 1980, Zippel 1979)* Let $f(x_1, \ldots, x_n) \neq 0$ be a multivariate polynomial of (total) degree at most d over a field F. Let $S \subseteq F$ be finite, and $(a_1, \ldots, a_n) \in S^n$ such that each a_i is chosen independently and uniformly at random from S. Then,

$$\Pr_{(a_1, \ldots, a_n) \in_r S^n} [f(a_1, \ldots, a_n) = 0] \leq d/|S|.$$

- *Proof idea.* Roots are far fewer than non-roots. Use induction on the number of variables.

(Homework / reading exercise)

Perfect matching in RNC

- Let PerfectMatching = {Bipartite graph G : G has a perfect matching}.
- Theorem. *(Lovasz 1979)* PerfectMatching $\in$ RNC^2.
- Correctness of the Algorithm.

1. Define $X = (x_{ij})_{i,j\in n}$ as follows: If $a_{ij}=0$, then $x_{ij}=0$. Else, x_{ij} is a formal variable.
2. $\det(X) = \sum_{\sigma\in S_n} (-1)^{\mathrm{sign}(\sigma)} \prod_{i\in[n]} x_{i\,\sigma(i)}$.

- Obs. $\det(X) \neq 0 \iff G$ has a perfect matching.
- If $\det(X) \neq 0$, then $\Pr[\det(B) \neq 0] \geq ½$ as degree of $\det(X) = n$ (by the Schwartz-Zippel lemma). $\blacksquare$

Perfect matching in RNC

- Theorem. *(Mulmuley, Vazirani, Vazirani 1987)* Finding a maximum matching in a general graph is in RNC^2.

- Is finding maximum matching in NC ? Open!

Perfect matching in RNC

- Theorem. *(Mulmuley, Vazirani, Vazirani 1987)* Finding a maximum matching in a general graph is in RNC^2.

- Is finding maximum matching in NC ? Open!
- Theorem. *(Fenner, Gurjar, Thierauf 2016; Svensson, Tarnawski 2017)* Finding a maximum matching in a general graph is in quasi-NC.

In $O((\log n)^3)$ time using $\exp(O((\log n)^3))$ processors,