Probabilistic Computation

Lecture 13
Understanding BPP
Recap
Recap

- Probabilistic computation
Recap

- Probabilistic computation
- NTM (on “random certificates”) for L:
Recap

- Probabilistic computation
- NTM (on "random certificates") for $L$:
  - $\Pr[M(x) = \text{yes}]$:
Recap

- Probabilistic computation
- NTM (on “random certificates”) for L:
  - $\Pr[M(x) = \text{yes}]:$

$x \notin L$

$x \in L$
Recap

- Probabilistic computation
- NTM (on “random certificates”) for L:
  - $\Pr[M(x) = yes]$: [Diagram showing $x \not\in L$ and $x \in L$.]
  - PTM for L: $Pr[yes]$: [Diagram showing $x \not\in L$ and $x \in L$.]
Recap

- Probabilistic computation
- NTM (on “random certificates”) for L:
  - \( \Pr[M(x) = \text{yes}] \):
  - PTM for L: \( \Pr[\text{yes}] \):
  - BPTM for L: \( \Pr[\text{yes}] \):
Recap

- Probabilistic computation
- NTM (on “random certificates”) for L:
  - $\Pr[M(x)=\text{yes}]:$
  - PTM for L: $\Pr[\text{yes}]:$
  - BPTM for L: $\Pr[\text{yes}]:$
  - RTM for L: $\Pr[\text{yes}]:$
Recap
Recap

PP, RP, co-RP, BPP
Recap

- PP, RP, co-RP, BPP
- PP too powerful: NP \subseteq PP
Recap

- PP, RP, co-RP, BPP
- PP too powerful: NP $\subseteq$ PP
- RP, BPP, with bounded gap
Recap

- PP, RP, co-RP, BPP
  - PP too powerful: NP ⊆ PP
  - RP, BPP, with bounded gap
    - Gap can be boosted from 1/poly to 1−1/exp
Recap

- PP, RP, co-RP, BPP
  - PP too powerful: NP ⊆ PP
  - RP, BPP, with bounded gap
    - Gap can be boosted from 1/poly to 1-1/exp
  - A realistic/useful computational model
Recap

- PP, RP, co-RP, BPP
  - PP too powerful: NP ⊆ PP
  - RP, BPP, with bounded gap
    - Gap can be boosted from $1/poly$ to $1-1/exp$
    - A realistic/useful computational model
- Today:
Recap

- PP, RP, co-RP, BPP
  - PP too powerful: \( \text{NP} \subseteq \text{PP} \)
  - RP, BPP, with bounded gap
    - Gap can be boosted from \( 1/\text{poly} \) to \( 1 - 1/\text{exp} \)
    - A realistic/useful computational model
- Today:
  - \( \text{NP} \not\subseteq \text{BPP} \), unless \( \text{PH} \) collapses
Recap

- PP, RP, co-RP, BPP
  - PP too powerful: NP ⊆ PP
  - RP, BPP, with bounded gap
    - Gap can be boosted from 1/poly to 1-1/exp
    - A realistic/useful computational model

Today:

- NP ⊄ BPP, unless PH collapses
- BPP ⊆ Σ₂^P ∩ Π₂^P
BPP vs. NP
BPP vs. NP

Can randomized algorithms efficiently decide all NP problems?
BPP vs. NP

Can randomized algorithms efficiently decide all NP problems?

Unlikely: $\text{NP} \subseteq \text{BPP} \Rightarrow \text{PH} = \Sigma_2^P$
BPP vs. NP

Can randomized algorithms efficiently decide all NP problems?

Unlikely: $\text{NP} \subseteq \text{BPP} \Rightarrow \text{PH} = \Sigma_2^P$

Will show $\text{BPP} \subseteq \text{P/poly}$
BPP vs. NP

Can randomized algorithms efficiently decide all NP problems?

*Unlikely:* $\text{NP} \subseteq \text{BPP} \Rightarrow \text{PH} = \Sigma_2^P$

Will show $\text{BPP} \subseteq \text{P/poly}$

Then $\text{NP} \subseteq \text{BPP} \Rightarrow \text{NP} \subseteq \text{P/poly}$
BPP vs. NP

Can randomized algorithms efficiently decide all NP problems?

Unlikely: $\text{NP} \subseteq \text{BPP} \Rightarrow \text{PH} = \Sigma_2^P$

Will show $\text{BPP} \subseteq \text{P/poly}$

Then $\text{NP} \subseteq \text{BPP} \Rightarrow \text{NP} \subseteq \text{P/poly}$

$\Rightarrow \text{PH} = \Sigma_2^P$
BPP ⊆ P/poly
BPP \subseteq P/poly

If error probability is sufficiently small, will show there should be at least one random tape which works for all $2^n$ inputs of length $n$
BPP $\subseteq$ P/poly

If error probability is sufficiently small, will show there should be at least one random tape which works for all $2^n$ inputs of length $n$.
BPP ⊆ P/poly

If error probability is sufficiently small, will show there should be at least one random tape which works for all $2^n$ inputs of length $n$.

Then, can give that random tape as advice.
BPP ⊆ P/poly

- If error probability is sufficiently small, will show there should be at least one random tape which works for all $2^n$ inputs of length $n$.

- Then, can give that random tape as advice.

- One such random tape if average (over $x$) error probability is less than $2^{-n}$.
BPP ⊆ P/poly

If error probability is sufficiently small, will show there should be at least one random tape which works for all $2^n$ inputs of length $n$

Then, can give that random tape as advice

One such random tape if average (over $x$) error probability is less than $2^{-n}$

BPP: can make worst error probability < $2^{-n}$
BPP vs. PH
BPP vs. PH

\[ \text{BPP} \subseteq \Sigma_2^P \]
BPP vs. PH

- $\text{BPP} \subseteq \Sigma_2^p$

- So $\text{BPP} \subseteq \Sigma_2^p \cap \Pi_2^p$
\[ \text{BPP} \subseteq \Sigma_2^p \]
$\text{BPP} \subseteq \Sigma_2^p$

$x \in L$: “for almost all” $r$, $M(x,r) = \text{yes}$
\[ \text{BPP} \subseteq \Sigma_2^p \]

- \( x \in L \): “for almost all” \( r \), \( M(x,r) = \text{yes} \)
- \( x \notin L \): \( M(x,r) = \text{yes} \) for very few \( r \)
\[ \text{BPP} \subseteq \Sigma^p_2 \]

- \( x \in L \): “for almost all” \( r \), \( M(x,r)=\text{yes} \)
- \( x \notin L \): \( M(x,r)=\text{yes} \) for very few \( r \)
- \( L = \{ x | \text{for almost all } r, M(x,r)=\text{yes} \} \)
BPP \subseteq \Sigma_2^P

- \( x \in L \): “for almost all” \( r \), \( M(x,r) = \text{yes} \)
- \( x \notin L \): \( M(x,r) = \text{yes} \) for very few \( r \)

\[ L = \{ x | \text{for almost all } r, \ M(x,r) = \text{yes} \} \]

- If it were “for all”, in coNP
\[
BPP \subseteq \Sigma_2^p
\]

\- \( x \in L \): “for almost all” \( r \), \( M(x, r) = yes \)

\- \( x \notin L \): \( M(x, r) = yes \) for very few \( r \)

\[ L = \{ x \mid \text{for almost all } r, \ M(x, r) = yes \} \]

\- If it were “for all”, in \( coNP \)

\[ L = \{ x \mid \exists \text{ a small “neighborhood”, } \forall z, \text{ for some } r \text{ “near” } z, \ M(x, r) = yes \} \]
BPP ⊆ Σ^{2}_{p}

- \( x \in L \): “for almost all” \( r \), \( M(x,r)=yes \)
- \( x \notin L \): \( M(x,r)=yes \) for very few \( r \)
- \( L = \{ x \mid \text{for almost all } r, M(x,r)=yes \} \)

- If it were “for all”, in coNP
  
  \( L = \{ x \mid \exists a \text{ small “neighborhood”, } \forall z, \text{ for some } r \text{ “near” } z, M(x,r)=yes \} \)

- Note: Neighborhood of \( z \) is small (polynomially large), so can go through all of them in polynomial time
\[ \text{BPP} \subseteq \Sigma_2^P \]

Space of random tapes = \( \{0,1\}^m \)

\( \text{Yes}_x = \{r \mid M(x,r) = \text{yes} \} \)
\( \text{BPP} \subseteq \Sigma_2^p \)

\[ x \in L: |\text{Yes}_x| > (1 - 2^{-n})2^m \]

Space of random tapes = \( \{0,1\}^m \)

\[ \text{Yes}_x = \{r | M(x,r) = \text{yes} \} \]
$\text{BPP} \subseteq \Sigma_2^p$

$x \in L: |\text{Yes}_x| > (1-2^{-n})2^m$

$x \notin L: |\text{Yes}_x| < 2^{-n}2^m$

Space of random tapes = $\{0,1\}^m$

$\text{Yes}_x = \{r | M(x,r) = \text{yes} \}$
BPP ⊆ \Sigma_2^P

\[
\begin{align*}
  x \in L: |\text{Yes}_x| &> (1 - 2^{-n})2^m \\
  x \notin L: |\text{Yes}_x| &< 2^{-n}2^m
\end{align*}
\]

Space of random tapes = \{0,1\}^m

\[\text{Yes}_x = \{r | M(x,r) = \text{yes}\}\]

\[\text{\textbullet x} \in L: \text{Will show that there exist a small set of shifts of Yes}_x \text{ that cover all } z\]
$\text{BPP} \subseteq \Sigma_2^p$

$x \in L: |\text{Yes}_x| > (1 - 2^{-n}) 2^m$

$x \notin L: |\text{Yes}_x| < 2^{-n} 2^m$

Space of random tapes = $\{0,1\}^m$

$\text{Yes}_x = \{r | M(x,r) = \text{yes} \}$

$\hat{\diamond} x \in L: \text{Will show that there exist a small set of shifts of } \text{Yes}_x \text{ that cover all } z$

$\hat{\diamond} \text{If } z \text{ is a shift of } r \in \text{Yes}_x, \text{ } r \text{ is in the neighborhood of } z$
$\textbf{BPP} \subseteq \Sigma_2^P$

$x \in L$: $|\text{Yes}_x| > (1-2^{-n})2^m$

$x \notin L$: $|\text{Yes}_x| < 2^{-n}2^m$

Space of random tapes = $\{0,1\}^m$

$\text{Yes}_x = \{r | \text{M}(x,r) = \text{yes}\}$

$x \in L$: Will show that there exist a small set of shifts of $\text{Yes}_x$ that cover all $z$

- If $z$ is a shift of $r \in \text{Yes}_x$, $r$ is in the neighborhood of $z$

$x \notin L$: $\text{Yes}_x$ very small, so its few shifts cover only a small region
\text{BPP} \subseteq \Sigma^p_{2}
BPP \subseteq \Sigma_2^P

“A small set of shifts”: \( P = \{u_1, u_2, ..., u_k\} \)
BPP \subseteq \Sigma_2^P

“A small set of shifts”: \( P = \{u_1, u_2, \ldots, u_k\} \)

\( P(r) = \{ r\oplus u_1, r\oplus u_2, \ldots, r\oplus u_k \} \) where \( r, u_i \) are \( m \)-bit strings, and \( k \) is “small” (poly(n))
$\text{BPP} \subseteq \Sigma_2^P$

“A small set of shifts”: $P = \{u_1, u_2, \ldots, u_k\}$

$P(r) = \{r \oplus u_1, r \oplus u_2, \ldots, r \oplus u_k\}$ where $r, u_i$ are $m$-bit strings, and $k$ is “small” (poly(n))

For each $x \in L$, does there exist a $P$ s.t. $P(\text{Yes}_x) := \cup_{r \in \text{Yes}(x)} P(r) = \{0, 1\}^m$?
\[ \text{BPP} \subseteq \Sigma_2^P \]

"A small set of shifts": \( P = \{u_1, u_2, \ldots, u_k\} \)

\( P(r) = \{ r \oplus u_1, r \oplus u_2, \ldots, r \oplus u_k \} \) where \( r, u_i \) are \( m \)-bit strings, and \( k \) is "small" (\( \text{poly}(n) \))

For each \( x \in L \), does there exist a \( P \) s.t. \( P(\text{Yes}_x) := \bigcup_{r \in \text{Yes}_x} P(r) = \{0,1\}^m \)?

Yes! For all large \( S \) (like \( \text{Yes}_x \)) can indeed find a \( P \) s.t. \( P(S) = \{0,1\}^m \)
\[ \text{BPP} \subseteq \Sigma_2^P \]

“A small set of shifts”: \( P = \{u_1, u_2, ..., u_k\} \)

\( P(r) = \{ r \oplus u_1, r \oplus u_2, ..., r \oplus u_k \} \) where \( r, u_i \) are m-bit strings, and \( k \) is “small” (\( \text{poly}(n) \))

For each \( x \in L \), does there exist a \( P \) s.t. \( P(\text{Yes}_x) := \bigcup_{r \in \text{Yes}(x)} P(r) = \{0,1\}^m \)?

Yes! For all large \( S \) (like \( \text{Yes}_x \)) can indeed find a \( P \) s.t. \( P(S) = \{0,1\}^m \)

In fact, most \( P \) work (if \( k \) big enough)!
$\text{BPP} \subseteq \Sigma^p_2$
\( \text{BPP} \subseteq \Sigma_2^P \)

- Probabilistic Method (finding hay in haystack)
$\text{BPP} \subseteq \Sigma_2^P$

- Probabilistic Method (finding hay in haystack)
- To prove $\exists \mathbf{P}$ with some property
$\text{BPP} \subseteq \Sigma_2^P$

- Probabilistic Method (finding hay in haystack)
- To prove $\exists P$ with some property
- Define a probability distribution over all candidate $P$'s and prove that the property holds with positive probability (often even close to one)
\[ \text{BPP} \subseteq \sum_2^P \]

- Probabilistic Method (finding hay in haystack)
  - To prove \( \exists P \) with some property
  - Define a probability distribution over all candidate \( P \)'s and prove that the property holds with positive probability (often even close to one)
  - Distribution s.t. easy to prove positive probability of property holding
\( \text{BPP} \subseteq \Sigma_2^p \)
\[ \text{BPP} \subseteq \Sigma_2^P \]

Probabilistic method to find \( P = \{u_1, u_2, \ldots, u_k\} \), s.t. for all large \( S \),
\( P(S) = \{0, 1\}^m \)
\[ \text{BPP} \subseteq \Sigma_2^P \]

- Probabilistic method to find \( P = \{u_1, u_2, \ldots, u_k\} \), s.t. for all large \( S \), 
  \( P(S) = \{0,1\}^m \)
- Distribution over \( P \)'s: randomized experiment to generate \( P \)
\[ \text{BPP} \subseteq \Sigma_2^p \]

- Probabilistic method to find \( \mathbf{P} = \{u_1, u_2, \ldots, u_k\} \), s.t. for all large \( S \), \( \mathbf{P}(S) = \{0,1\}^m \)

- Distribution over \( \mathbf{P} \)'s: randomized experiment to generate \( \mathbf{P} \)

- Pick each \( u_i \) independently, and uniformly at random from \( \{0,1\}^m \)
\( \text{BPP} \subseteq \Sigma_2^p \)

- Probabilistic method to find \( P = \{u_1, u_2, \ldots, u_k\} \), s.t. for all large \( S \), \( P(S) = \{0,1\}^m \)

- Distribution over \( P \)'s: randomized experiment to generate \( P \)

- Pick each \( u_i \) independently, and uniformly at random from \( \{0,1\}^m \)

- \( \Pr_{(\text{over } P)}[P(S) \neq \{0,1\}^m] = \Pr_{(\text{over } P)}[\exists z \; z \notin P(S)] \)
\[ \text{BPP} \subseteq \Sigma_2^P \]

\footnotesize{Probabilistic method to find \( P = \{u_1, u_2, \ldots, u_k\} \), s.t. for all large \( S \), \( P(S) = \{0,1\}^m \)

\footnotesize{Distribution over \( P \)'s: randomized experiment to generate \( P \)

\footnotesize{Pick each \( u_i \) independently, and uniformly at random from \( \{0,1\}^m \)

\footnotesize{\( \text{Pr}(\text{over } P)[P(S) \neq \{0,1\}^m] = \text{Pr}(\text{over } P)[\exists z \ z \notin P(S)] \leq \sum_z \text{Pr}(\text{over } P)[z \notin P(S)] \)}
BPP ⊆ Σ²

Probabilistic method to find \( P = \{u_1, u_2, \ldots, u_k\} \), s.t. for all large \( S \), \( P(S) = \{0, 1\}^m \)

Distribution over \( P \)'s: randomized experiment to generate \( P \)

Pick each \( u_i \) independently, and uniformly at random from \( \{0, 1\}^m \)

\[ \Pr(\text{over } P)[P(S) \neq \{0, 1\}^m] = \Pr(\text{over } P)[\exists z \ z \notin P(S)] \]
\[ \leq \Sigma z \ \Pr(\text{over } p)[z \notin P(S)] = \Sigma z \ \Pr(\text{over } u_1 \ldots u_k)[\forall i \ z \oplus u_i \notin S] \]
\( \text{BPP} \subseteq \Sigma_2^P \)

- Probabilistic method to find \( P = \{u_1, u_2, \ldots, u_k\} \), s.t. for all large \( S \), \( P(S) = \{0,1\}^m \)

- Distribution over \( P \)'s: randomized experiment to generate \( P \)

- Pick each \( u_i \) independently, and uniformly at random from \( \{0,1\}^m \)

- \( \Pr_{\text{over } P}[P(S) \neq \{0,1\}^m] = \Pr_{\text{over } P}[\exists z \ z \notin P(S)] \)
  \[ \leq \sum_z \Pr_{\text{over } P}[z \notin P(S)] = \sum_z \Pr_{\text{over } u_1..u_k}[\forall i \ z \oplus u_i \notin S] \]
  \[ = \sum_z \prod_i \Pr_{\text{over } u_i}[z \oplus u_i \notin S] \]
BPP \subseteq \Sigma_2^P

- Probabilistic method to find $P = \{u_1,u_2,\ldots,u_k\}$, s.t. for all large $S$, $P(S) = \{0,1\}^m$

- Distribution over $P$'s: randomized experiment to generate $P$

- Pick each $u_i$ independently, and uniformly at random from $\{0,1\}^m$

- $\Pr_{\text{over } P}[P(S) \neq \{0,1\}^m] = \Pr_{\text{over } P}[\exists z \ z \notin P(S)]$
  \[ \leq \sum_z \Pr_{\text{over } P}[z \notin P(S)] = \sum_z \Pr_{\text{over } u_1..u_k}[\forall i \ z \oplus u_i \notin S] \]
  \[ = \sum_z \prod_i \Pr_{\text{over } u_i}[z \oplus u_i \notin S] = \sum_z \prod_i \Pr_{\text{over } u_i}[u_i \notin z \oplus S] \]
\( \text{BPP} \subseteq \Sigma_2^P \)

- Probabilistic method to find \( P = \{u_1, u_2, \ldots, u_k\} \), s.t. for all large \( S \), \( P(S) = \{0,1\}^m \)

- Distribution over \( P \)'s: randomized experiment to generate \( P \)

- Pick each \( u_i \) independently, and uniformly at random from \( \{0,1\}^m \)

\[
\Pr_{\text{over } P}[P(S) \neq \{0,1\}^m] = \Pr_{\text{over } P}[\exists z \; z \notin P(S)] \\
\leq \sum_z \Pr_{\text{over } P}[z \notin P(S)] = \sum_z \Pr_{\text{over } u_1..u_k}[\forall i \; z \oplus u_i \notin S] \\
= \sum_z \prod_i \Pr_{\text{over } u_i}[z \oplus u_i \notin S] = \sum_z \prod_i \Pr_{\text{over } u_i}[u_i \notin z \oplus S] \\
= \sum_z \prod_i (|S^c|/2^m)
\]
\[ \text{BPP} \subseteq \Sigma_2^P \]

- Probabilistic method to find \( P = \{ u_1, u_2, ..., u_k \} \), s.t. for all large \( S \), \( P(S) = \{0,1\}^m \)

- Distribution over \( P \)'s: randomized experiment to generate \( P \)

- Pick each \( u_i \) independently, and uniformly at random from \( \{0,1\}^m \)

- \( \Pr_{\text{over } P}[P(S) \neq \{0,1\}^m] = \Pr_{\text{over } P}[\exists z \ z \notin P(S)] \)

\[
\leq \sum_z \Pr_{\text{over } P}[z \notin P(S)] = \sum_z \Pr_{\text{over } u_1...u_k}[\forall i \ z \oplus u_i \notin S]
\]

\[
= \sum_z \Pi_i \Pr_{\text{over } u_i}[z \oplus u_i \notin S] = \sum_z \Pi_i \Pr_{\text{over } u_i}[u_i \notin z \oplus S]
\]

\[
= \sum_z \Pi_i \left( |S^c|/2^m \right) < \sum_z \Pi_i 2^{-n}
\]
\[ \mathsf{BPP} \subseteq \Sigma_2^P \]

- Probabilistic method to find \( P = \{u_1, u_2, \ldots, u_k\} \), s.t. for all large \( S \), \( P(S) = \{0,1\}^m \)

- Distribution over \( P \)'s: randomized experiment to generate \( P \)

- Pick each \( u_i \) independently, and uniformly at random from \( \{0,1\}^m \)

\[ \Pr_{\text{over } P}[P(S) \neq \{0,1\}^m] = \Pr_{\text{over } P}[\exists z \ z \notin P(S)] \leq \sum_z \Pr_{\text{over } P}[z \notin P(S)] = \sum_z \Pr_{\text{over } u_1 \ldots u_k}[\forall i \ z \oplus u_i \notin S] = \sum_z \prod_i \Pr_{\text{over } u_i}[z \oplus u_i \notin S] = \sum_z \prod_i \Pr_{\text{over } u_i}[u_i \notin z \oplus S] = \sum_z \prod_i (|S^c|/2^m) < \sum_z \prod_i 2^{-n} = 2^m.(2^{-n})^k = 1 \]
\[ \text{BPP} \subseteq \Sigma_2^P \]

- Probabilistic method to find \( P = \{u_1, u_2, \ldots, u_k\} \), s.t. for all large \( S \), \( P(S) = \{0,1\}^m \)

- Distribution over \( P \)'s: randomized experiment to generate \( P \)

- Pick each \( u_i \) independently, and uniformly at random from \( \{0,1\}^m \)

- \( \Pr \text{(over } P) [P(S) \neq \{0,1\}^m] = \Pr \text{(over } P) [\exists z \ z \notin P(S)] \)

= \( \sum z \Pr \text{(over } P) [z \notin P(S)] = \sum z \Pr \text{(over } u_1 \ldots u_k) [\forall i \ z \oplus u_i \notin S] \)

= \( \sum z \prod_i \Pr \text{(over } u_i) [z \oplus u_i \notin S] = \sum z \prod_i \Pr \text{(over } u_i) [u_i \notin z \oplus S] \)

= \( \sum z \prod_i (|S^c|/2^m) < \sum z \prod_i 2^{-n} = 2^m(2^{-n})^k = 1 \)

- So (with \( |S| > (1-2^{-n})2^m \) and \( k=m/n \), \( \exists P, P(S) = \{0,1\}^m \)
\[ \text{BPP} \subseteq \Sigma_2^p \]

\[ x \in L: |\text{Yes}_x| > (1 - 2^{-n})2^m \]

\[ x \notin L: |\text{Yes}_x| < 2^{-n}2^m \]

Space of random strings = \{0,1\}^m

\[ \text{Yes}_x = \{ r \mid M(x,r) = \text{yes} \} \]
For each $x \in L$, $\exists P$ (of size $k=m/n$) s.t. $P(\text{Yes}_x) = \{0,1\}^m$.

$\text{BPP} \subseteq \Sigma_2^P$

$x \in L$: $|\text{Yes}_x| > (1 - 2^{-n})2^m$

$x \notin L$: $|\text{Yes}_x| < 2^{-n}2^m$

Space of random strings = $\{0,1\}^m$

$\text{Yes}_x = \{r | M(x,r) = \text{yes} \}$

For each $x \in L$, $\exists P$ (of size $k=m/n$) s.t. $P(\text{Yes}_x) = \{0,1\}^m$. 
For each $x \in L$, $|\text{Yes}_x| > (1-2^{-n})2^m$

For each $x \notin L$, $P(\text{Yes}_x) \subseteq \{0,1\}^m$

$\text{BPP} \subseteq \Sigma_2^P$

Space of random strings = $\{0,1\}^m$

$\text{Yes}_x = \{r | M(x,r) = \text{yes}\}$

For each $x \in L$, $\exists P$ (of size $k=m/n$) s.t. $P(\text{Yes}_x) = \{0,1\}^m$

For each $x \notin L$, $P(\text{Yes}_x) \subseteq \{0,1\}^m$
\[ \text{BPP} \subseteq \Sigma_2^P \]

For each \( x \in L \), \( \exists P \) (of size \( k = m/n \)) s.t. \( P(\text{Yes}_x) = \{0,1\}^m \)

For each \( x \not\in L \), \( P(\text{Yes}_x) \subset \{0,1\}^m \)

Space of random strings = \( \{0,1\}^m \)

\( \text{Yes}_x = \{ r \mid M(x,r) = \text{yes} \} \)

\[ |P(\text{Yes}_x)| \leq k|\text{Yes}_x| = (m/n) 2^{-n}2^m < 2^m \]
\[ \text{BPP} \subseteq \Sigma_2^p \]

For each \( x \in L \), \( \exists P \) (of size \( k = \frac{m}{n} \)) s.t. \( P(\text{Yes}_x) = \{0, 1\}^m \)

For each \( x \notin L \), \( P(\text{Yes}_x) \subseteq \{0, 1\}^m \)

\[ | P(\text{Yes}_x) | \leq k | \text{Yes}_x | = \left( \frac{m}{n} \right) 2^{-n} 2^m < 2^m \]

\[ L = \{ x | \exists P \forall z \text{ for some } r \in P^{-1}(z) \text{ M}(x,r) = \text{yes} \} \]

Space of random strings = \{0,1\}^m

\[ \text{Yes}_x = \{ r | \text{M}(x,r) = \text{yes} \} \]
BPP-Complete Problem?
BPP-Complete Problem?

Not known!
BPP-Complete Problem?

Not known!

$L = \{ (M,x,1^t) \mid M(x) = \text{yes in time } t \text{ with probability } > 2/3 \}$?
BPP-Complete Problem?

Not known!

$L = \{ (M,x,1^t) \mid \text{M(x)=yes in time } t \text{ with probability } > \frac{2}{3} \}$?

Is indeed BPP-Hard
BPP-Complete Problem?

Not known!

\[ L = \{ (M,x,1^t) \mid M(x) = \text{yes in time } t \text{ with probability } > 2/3 \} \]

Is indeed BPP-Hard

But in BPP?
BPP-Complete Problem?

Not known!

\[ L = \{ (M,x,1^t) \mid M(x) = \text{yes in time } t \text{ with probability } > 2/3 \} \]

Is indeed BPP-Hard

But in BPP?

Just run \( M(x) \) for \( t \) steps and accept if it accepts?
BPP-Complete Problem?

- Not known!
- \( L = \{ (M,x,1^t) \ | \ M(x) = \text{yes in time } t \text{ with probability } > 2/3 \} \) ?
- Is indeed BPP-Hard
- But in BPP?
  - Just run \( M(x) \) for \( t \) steps and accept if it accepts?
  - If \((M,x,1^t)\) in \( L \), we will indeed accept with prob. > 2/3
BPP-Complete Problem?

Not known!

\[ L = \{ (M, x, 1^t) \mid M(x) = \text{yes in time } t \text{ with probability } > 2/3 \} \]

Is indeed BPP-Hard

But in BPP?

Just run \( M(x) \) for \( t \) steps and accept if it accepts?

If \((M, x, 1^t)\) in \( L \), we will indeed accept with prob. \( > 2/3 \)

But \( M \) may not have a bounded gap. Then, if \((M, x, 1^t)\) not in \( L \), we may accept with prob. very close to \( 2/3 \).
BPTIME-Hierarchy Theorem?
BPTIME-Hierarchy Theorem?

BPTIME(n) ⊊ BPTIME(n^{100})?
BPTIME-Hierarchy Theorem?

- $\text{BPTIME}(n) \nsubseteq \text{BPTIME}(n^{100})$?
- Not known!
BPTIME-Hierarchy Theorem?

- $\text{BPTIME}(n) \subseteq \text{BPTIME}(n^{100})$?
- Not known!
- But is true for $\text{BPTIME}(T)/1$
Today
Today

Probabilistic computation
Today

- Probabilistic computation
- $\mathsf{BPP} \subseteq \mathsf{P/poly}$ (so if $\mathsf{NP} \subseteq \mathsf{BPP}$, then $\mathsf{PH}=\Sigma_2^P$)
Today

- Probabilistic computation
- \( \text{BPP} \subseteq \text{P/poly} \) (so if \( \text{NP} \subseteq \text{BPP} \), then \( \text{PH} = \Sigma_2^P \))
- \( \text{BPP} \subseteq \Sigma_2^P \cap \Pi_2^P \)
Today

- Probabilistic computation
- $\text{BPP} \subseteq \text{P/poly}$ (so if $\text{NP} \subseteq \text{BPP}$, then $\text{PH}=\Sigma_2^P$)
- $\text{BPP} \subseteq \Sigma_2^P \cap \Pi_2^P$
- Coming up
Today

- Probabilistic computation
- \( \text{BPP} \subseteq \text{P/poly} \) (so if \( \text{NP} \subseteq \text{BPP} \), then \( \text{PH}=\Sigma_2^P \))
- \( \text{BPP} \subseteq \Sigma_2^P \cap \Pi_2^P \)
- Coming up
  - Basic randomized algorithmic techniques
Today

- Probabilistic computation
- $\text{BPP} \subseteq \text{P/poly}$ (so if $\text{NP} \subseteq \text{BPP}$, then $\text{PH}=\Sigma_2^P$)
- $\text{BPP} \subseteq \Sigma_2^P \cap \Pi_2^P$
- Coming up
  - Basic randomized algorithmic techniques
  - Saving on randomness