BPP:

Suppose a $L \in BPP$ then

$\forall x \in L \quad Pr_r[M(x,r)=1] \geq \frac{2}{3}$

$\forall x \notin L \quad Pr_r[M(x,r)=1] \leq \frac{1}{3}$

$\forall x \notin \bar{L} \quad Pr[M(x,r)=1] \geq \frac{2}{3}$

$\forall x \in \bar{L} \quad Pr[M(x,r)=1] \leq \frac{1}{3}$

Thm: $\forall d > 0, \exists M'$ s.t — "Error reduction"

$\forall x \in L \quad Pr_{r'}[M'(x,r')=1] \geq 1 - 2^{-|x|^d}$

$\forall x \notin L \quad Pr_{r'}[M'(x,r')=1] \leq 2^{-|x|^d}$

$Pr[M'(x,r') \neq 1^L(x)] \leq 2^{-|x|^d}$.

M' is constr. from M by swapping accept and rej states.

coBPP: $\forall L \in BPP$;

$\forall x \in \bar{L} \quad Pr_r[M'(x,r)=1] \geq \frac{2}{3}$

$\forall x \notin \bar{L} \quad Pr_r[M'(x,r)=1] \leq \frac{1}{3}$.

Given a $\mathcal{C}$ what is co$\mathcal{C}$?

$L \in BPP$

$\bar{L} \in coBPP$

Claim: BPP is closed under complement.

$BPP = coBPP$.

(1) $BPP \subseteq coBPP$ (2) $coBPP \subseteq BPP$

↓

L has a BPP machine

$\forall x \in L \quad Pr_r[M(x,r)=1] \geq \frac{2}{3}$

$\forall x \notin L \quad Pr_r[M(x,r)=1] \leq \frac{1}{3}$

M' is obtained from M by swapping acc and rej. states

$\forall x \notin \bar{L} \quad Pr[M'(x,r) \neq 1] \geq \frac{2}{3}$

$\forall x \in \bar{L} \quad Pr[M'(x,r) \neq 1] \leq \frac{1}{3}$.

Rephrased as

$\forall x \notin L \quad \Pr[M'(x,r)=1] \leq \frac{1}{3}$

$\forall x \in L \quad \Pr[M'(x,r)=1] \geq \frac{2}{3}$.

} $\frac{1-2^{-m}}{\frac{2}{3}}$ vs $\frac{2^{-m}}{\frac{1}{3}}$ instead of and

(2) follows from "similar" arguments.

Theorem [Adleman 1978]

$BPP \subseteq P/poly$

$P/poly$:= Class of polynomial sized ckts.

$= \{ \bigcup_{c>0} SIZE(n^c) \}_{n \geq 0}$

(c is a const; $SIZE(n^c)$: Class of ckts of size n^c.)

Remark: Circuits in $P/poly$ can be "defined" or "constructed" w/ polynomial size advice from $\sim$.

Pf: Goal is to construct a $P/poly$ circuit for a language $L \in BPP$.

$\forall x \in \{0,1\}^n$, $\Pr_{r \sim \{0,1\}^m}[M(x,r) \neq \mathbb{1}^L(x)] \leq 2^{-(n+1)}$

} Pick d s.t $|x|^d = n+1$.

$\mathbb{1}^L$: Indicator function

$\mathbb{1}^L(x) = \begin{cases} 1 & \text{if } x \in L \\ 0 & \text{o/w} \end{cases}$

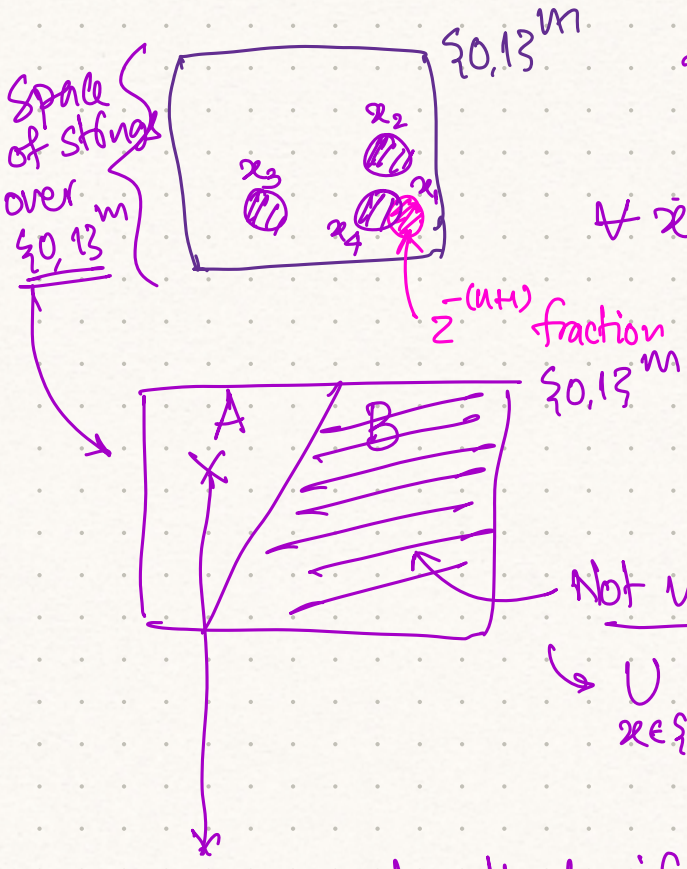
↳ Rephrasing it:

For a string $x \in \{0,1\}^n$, # of random strings r s.t $M(x,r) \neq \mathbb{1}^L(x)$ is at most $\frac{2^m}{2^{(n+1)}}$. | $m = poly(n)$

We say a random string r is **not useful** if $M(x,r) \neq \mathbb{1}^L(x)$.

By union bound, we can say that at most $2^n \cdot \frac{2^m}{2^{n+1}}$

random strings are not useful.



$\forall x$, there are $\leq \frac{1}{2^{n+1}}$ fraction of non useful strings

$\forall x$, there are at most $\frac{2^n \cdot 2^m}{2^{n+1}}$ non useful strings $= \frac{2^m}{2}$

Not useful random

$\hookrightarrow \bigcup_{x \in \{0,1\}^n}$ (Not useful strings for x)

$\hookrightarrow x \in L$, $M(x,r) \neq 1^L(x)$.

r_0 has property that if $x \in L$ then $M(x,r_0) = 1 = \underbrace{1^L(x)}_{=1}$

From the arguments above, $|B| \leq \frac{2^m}{2}$ and $|A| = 2^m - |B| \geq \frac{2^m}{2}$.

$\Rightarrow$ There are a lot of r_0's s.t $\forall x$; $M(x,r_0) = 1^L(x)$.

This means that fixing the random string to r_0, the turing machine M will output correctly.

$\forall x \in L \cap \{0,1\}^n$, $M^{r_0}(x) = 1$

Circuit for length n input

Cook Levin $\longrightarrow$ Ckt reduction

$\forall x \in L \cap \{0,1\}^n$, $C_n^{r_0}(x) = 1$

$r_0 \in \{0,1\}^m$ and $m = \text{poly}(n)$

r_0 is dependent on n.

Given r_0 as advise, we can construct $C^{r_0}(x)$ ckt.

$\therefore C^{r_0}(x)$ is of poly size, $L \in P/\text{poly}$ as well.

Theorem [Sipser-Gacs]

$$BPP \subseteq \Sigma_2^P \cap \Pi_2^P$$

Pf: $L \in \Sigma_2^P$ then

$$\forall x \in L,\ \exists u_1 \in \{0,1\}^{p(|x|)}\ \forall u_2 \in \{0,1\}^{q(|x|)}\ M(x,u_1,u_2)=1.$$

Π_2^P is a complement class of Σ_2^P. Say $L' \in \Pi_2^P$

$$\forall y \in L',\ \forall v_1 \in \{0,1\}^{p'(|y|)}\ \exists v_2 \in \{0,1\}^{q'(|y|)}\ M'(y,v_1,v_2)=1.$$

Want: If $L \in BPP$ then $L \in \Sigma_2^P$ and $L \in \Pi_2^P$.

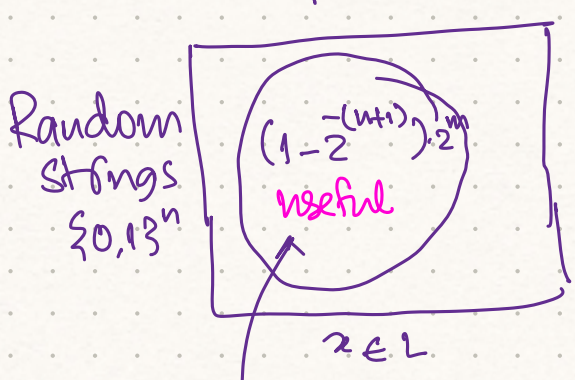
$\forall x \in \{0,1\}^n$; $\Pr_r[M(x,r) \neq \mathbb{1}^L(x)] \leq 2^{-(n+1)}$ ...

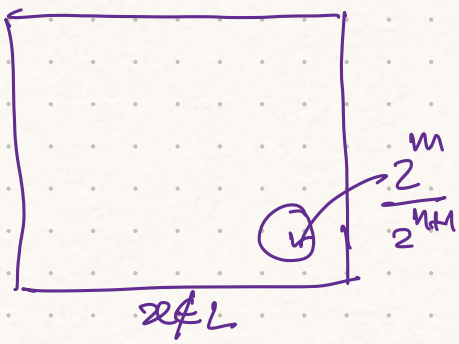
$S_x = \{r \mid M(x,r)=1\}$ (At this point we do not know if $x \in L$)

If $x \in L$, $|S_x| \geq (1-2^{-(n+1)}) \cdot 2^m$,

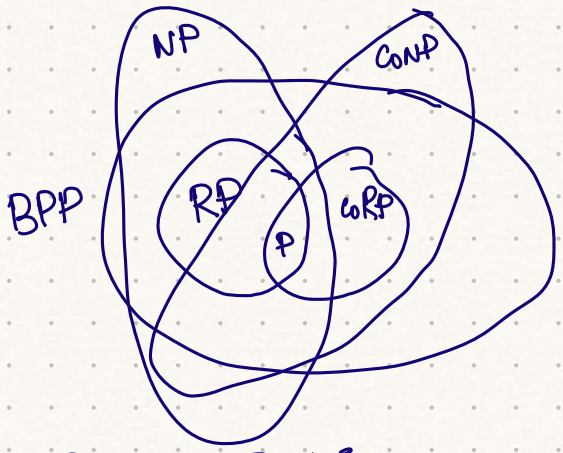
else $|S_x| \leq 2^{-(n+1)} \cdot 2^m$.

We want to give a Σ_2^P statement to show the membership of a string $x \in L \in BPP$.





Thm If $NP \subseteq P/poly$ then $PH \subseteq \Sigma_2^P$. [Karp Lipton]



If $NP \subseteq RP$ then $PH \subseteq \Sigma_2^P$.

$\forall r \in S_x$: $M(x,r) = \mathbb{1}^L(x)$.

Want to say that if $x \in L$, then S_x is "large".
else S_x is "small".

↳ Can be phrased as a Σ_2^P statement. (To be shown)

$\Rightarrow$ $BPP \subseteq \Sigma_2^P$; $coBPP \subseteq \Pi_2^P$

$BPP = coBPP$

$\Downarrow$

$BPP \subseteq \Sigma_2^P \cap \Pi_2^P$.