Goals

- Prove can't prove P=NP using simulation
- Define Probabilistic computation classes

Announcements

- Pet photos
- 3 weeks!

Thm: $\exists\ \mathcal{O}: P^{\mathcal{O}} \neq NP^{\mathcal{O}}$

Proof Strategy:

Create oracle B by:

- Enumerate all oracular TM's $M_1^{\mathcal{O}}, M_2^{\mathcal{O}}, M_3^{\mathcal{O}} \ldots$
- Starting at $i=1$, (and repeating for each successive $i = 2, 3, 4, \ldots$) pick n_i (will tell how to pick) and run M_i^B on input 1^{n_i} for time $(n_i)^i$. Note $|1^{n_i}| = n_i$.

y	$y \in B$
$\emptyset$	No
0	
1	
00	
10	
$\vdots$	
1001	
$\vdots$	

- Pick n_i to be smallest # s.t.
  - $(n_i)^i < 2^{n_i}$
  - no string of length n_i has be assigned

[Initially, $n_1 = ?$] $n_1 = 1$ b/c - $1^1 < 2^1$ - no strings assigned

- Run $M_i^B(1^{n_i})$ for $(n_i)^i$ steps

$\downarrow$ starts running $\downarrow$

Query $y \in B$?

Look at our list. If y already decided, answer consistently

If y not already decided, decide $y \notin B$. Add to list

- If $M_i^B(1^{n_i})$ doesn't terminate or outputs 1, set all y s.t. $|y| \leq n_i$ that haven't been queried to "No" in B. (All y s.t. $|y| = n_i$ will be No.)
- If $M_i^B(1^{n_i})$ outputs 0, set one string of length n_i to be "Yes" in B, and all other unqueried y, $|y| \leq n_i$ to be "No"
  - There are 2^{n_i} strings of length n_i
  - B/c only $(n_i)^i$ steps, only $(n_i)^i$ of them could be queried and so assign.

Now consider the language

$$L_B = \{1^n : \exists\ y \in B \text{ s.t } |y| = n\}$$

Thm: $L_B \notin P^B$

Suppose for contradiction that $L_B \in P^B$. Then there is TM M^B that decides L_B in Cn^d time, for some constants d, C. Let α be a natural number s.t. $M_\alpha^B = M^B$ s.t. $n_\alpha^\alpha > Cn_\alpha^d$. But then $M_\alpha^B(1^{n_\alpha})$ will be incorrect by our construction of B.

Thm $L_B \in NP^B$

Let $M^B(x,u)$ be the TM that accepts if

- $x = 1^n$ for some u
- $|u| = n$
- M^B queries if $u \in B$ and gets answer yes

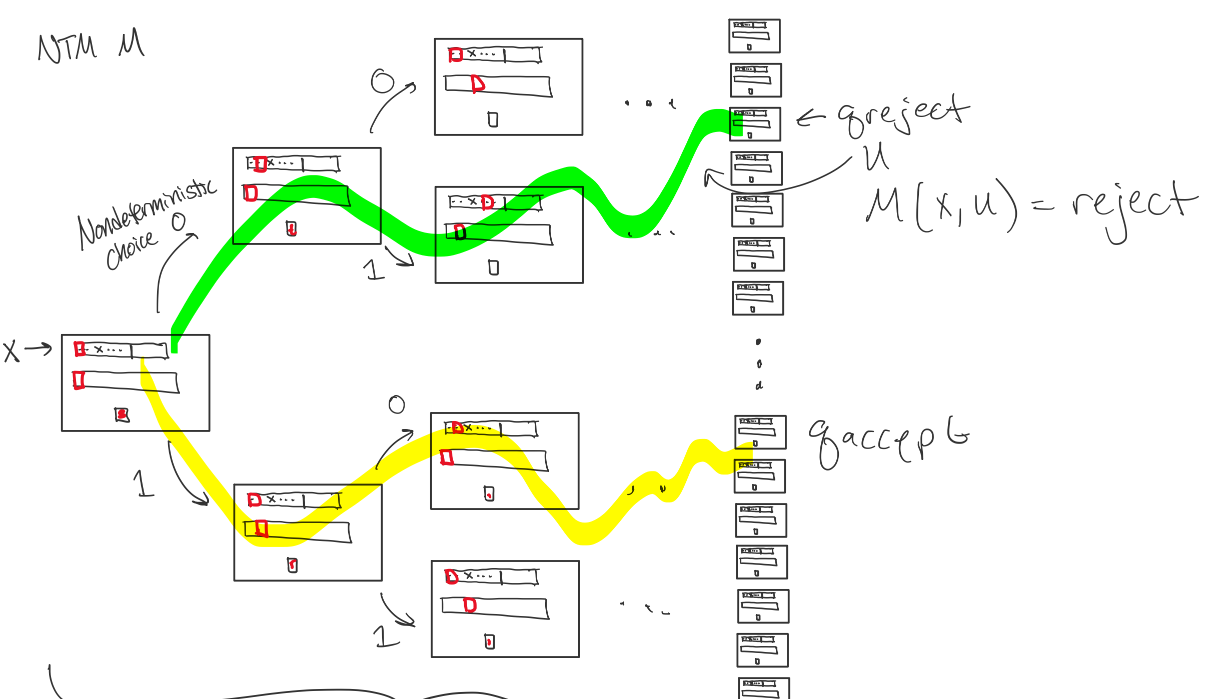
$M^B(x,u)$ runs in polynomial time

Thus: $P^B \neq NP^B$

9s Probabilistic Computation

Wednesday, April 20, 2022
1:19 PM

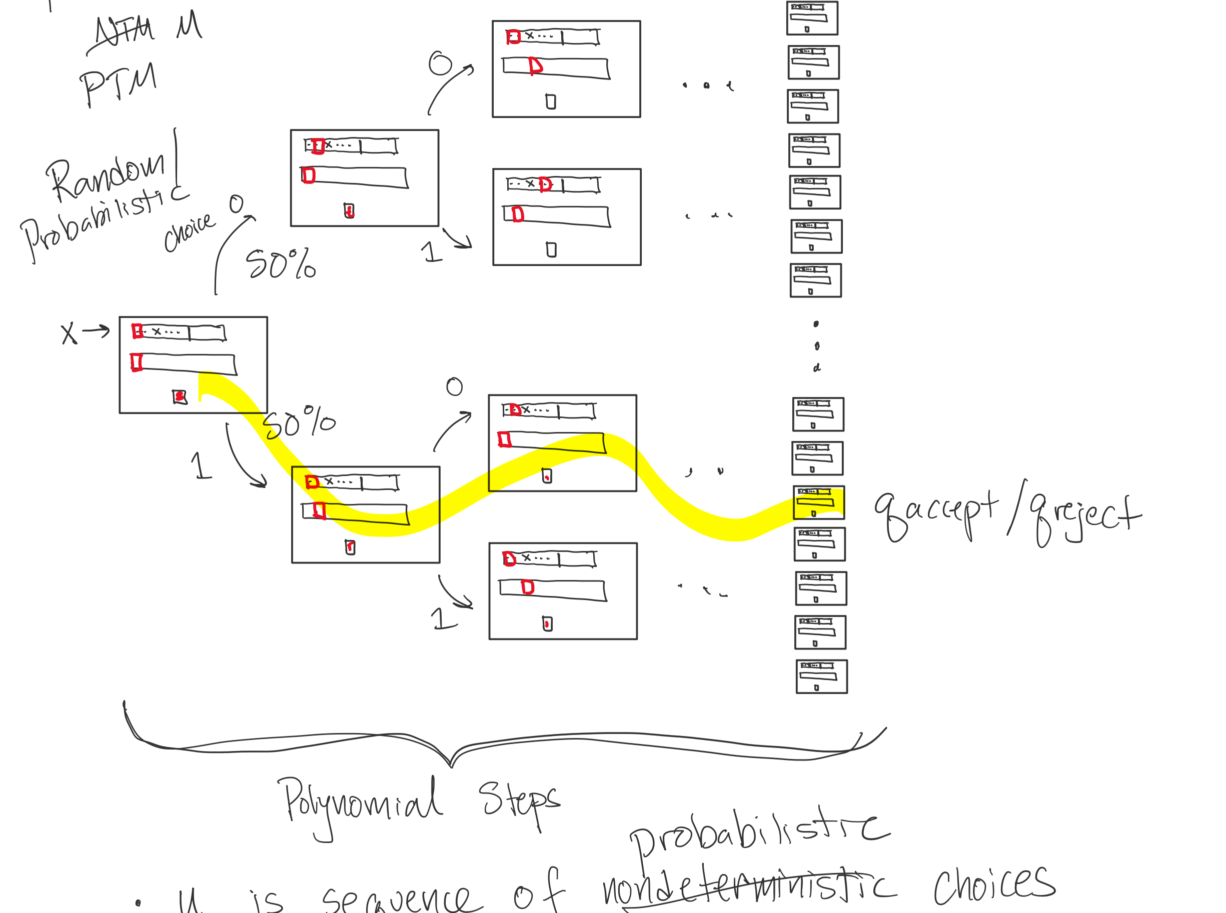
Recall Non-deterministic Polynomial Time Computation:



u a path through NTM
$M(x,u)$ the output of following u

- M accepts if $\exists\ u: M(x,u)$ accepts
- M rejects if $\forall\ u: M(x,u)$ rejects

Probabilistic Polynomial Time Computation:



- u is sequence of ~~nondeterministic~~ probabilistic choices
- $M(x,u)$ is output of following u

M accepts x if $\Pr_u[M(x,u) = \text{accept}] \geq 2/3$

M rejects x if $\Pr_u[M(x,u) = \text{rejects}] \geq 2/3$

BPP (Bounded Probabilistic Polynomial Time)

$L \in BPP$ if $\exists$ a probabilistic TM M, M should halt in polynomial time regardless of its random choices and $\forall x \in \{0,1\}^*$

- If $x \in L \rightarrow M$ accepts
- If $x \notin L \rightarrow M$ rejects