Goals
- Prove facts 3 and 4

Questions
- If we know that diagonalization and simulation fails, does that mean the only way to prove if P = NP is to invent a new proof technique?

Announcements
- Pet photos
- Seniors: Fill out 701 form

Thm: $\exists O : P^O = NP^O$ ←

Pf: $\forall O$, $P^O \subseteq NP^O$ b/c $P \subseteq NP$ using simulation.

We need to find an O s.t. $NP^O \subseteq P^O$

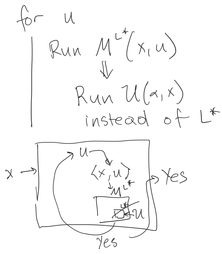
Let $L^* = \{\langle \alpha, x, 1^n \rangle : M_\alpha(x) \text{ outputs } 1 \text{ in } 2^n \text{ steps}\}$ ← ($1^3 = 111$)

We will prove $NP^{L^*} \subseteq EXP$ and ① $EXP \subseteq P^{L^*}$ ②

Then $NP^{L^*} \subseteq P^{L^*}$

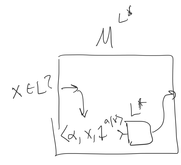
Fact: To decide if $x \in L^*$, can run $U(\alpha, x)$ for $n \cdot 2^n$ steps

$NP^{L^*} \subseteq EXP$ ① Let $L \in NP^{L^*}$. Then there exist a polynomial time TM M^{L^*} s.t. if $x \in L$, $\exists\, u : M^{L^*}(x,u) = 1$. Let M' be the TM that iterate through every possible u and it will run M^{L^*} except, whenever M^{L^*} queries L^*, M' runs U to determine the result of the query. If any u causes an accept, M' accepts. M' runs in exponential time so $L \in EXP$.



$EXP \subseteq P^{L^*}$ ② Let $L \in EXP$, then there is a TM M_α that decides L in $2^{a|x|}$ steps on input x, where $a \in \mathbb{N}$. Let M^{L^*} be the machine that on input x, queries $\langle \alpha, x, 1^{a|x|} \rangle$ to L^* and outputs the result. M^{L^*} runs in polynomial time, so $L \in P^{L^*}$

$O(i)$



Result: $P^{L^*} = NP^{L^*}$ ⇒ Can not use diagonalization to prove $P \neq NP$

Thm: $\exists O : P^O \neq NP^O$

Create our oracle B

- Enumerate all oracular TM's M_1^O, M_2^O, M_3^O
- Starting at $i = 1$ (and repeating for each $i = 2, 3, 4, \ldots$) pick a number n_i (will tell how to pick later) and run M_i^B on input 1^{n_i}.
- Pick n_i to be the smallest number s.t.
  - $(n_i)^i < 2^{n_i}$
  - no string of length n_i has been assigned

y	$y \in B$?
$\emptyset$	No
0	Yes
1	No
00	No ✓
10	
⋮	

$n_1 = 1$

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

Query $y \in B$ 00 in B? No

No

Look at list + if y is already assigned, be consistent

If unassigned, Answer No + update list.

- If $M_i^B(1^{n_i})$ doesn't terminate in $(n_i)^i$ steps, or if outputs 1, then set all strings y s.t. $|y| \leq n_i$ that haven't already been assigned to "No"
- If $M_i^B(1^{n_i})$ outputs 0 in $(n_i)^i$ steps, then, assign one string of length n_i to be "Yes".
  - There are 2^{n_i} such strings
  - $M_i^B(1^{n_i})$ only ran $(n_i)^i$ steps $< 2^{n_i}$