Diagonalization and P vs NP

Oracle machines

$\mathcal{O} : \{0,1\}^* \to \{0,1\}$

$M^{\mathcal{O}}$: Turing machine with access to oracle $\mathcal{O}$.

↳ query $\mathcal{O}(x)$.

$P^{\mathcal{O}}$: poly time computable using $\mathcal{O}$.

$NP^{\mathcal{O}}$: nondet. poly time using $\mathcal{O}$.

Thm: $\exists$ A, B s.t

$P^A = NP^A$

$P^B \neq NP^B$

Pf: $A(\underbrace{\alpha}_{\text{T.M code}}, x) = 1$ if $M_\alpha(x) = 1$ in $2^{|x|}$ steps.

$= 0$ o.w

1) $P^A = EXP$

$P^A \supseteq EXP$

$P^A \subseteq EXP$

2) $NP^A = EXP$

$NP^A \supseteq P^A \supseteq EXP$

$NP^A \subseteq EXP$

$B : \{0,1\}^* \to \{0,1\}$

$\exists f \in NP^B, f \notin P^B$

$$f(x) = \begin{cases} 1 & \text{if } \exists y,\ |y| = |x| \text{ and } B(y) = 1 \\ 0 & \text{o.w.} \end{cases}$$

Defining B

Attempt #2.

Let $M_1, M_2, M_3, \ldots$ be an enumeration of all machines.

Every machine occurs infinite # of times

Phase i: Consider machine M_i, inputs of length t_i, where t_i is so large that $B(x)$ undef. for $x \in \{0,1\}^{t_i}$

Run $M_i(1^{t_i})$ for $2^{t_i/10}$ steps

~~Whenever M_i queries $x \in \{0,1\}^{t_i}$, set $B(x) = 0$~~

Whenever M_i queries x if $B(x)$ is defined, return $B(x)$ o.w set $B(x) = 0$.

If M_i halts with output 1, set $B(x) = 0$ for all x

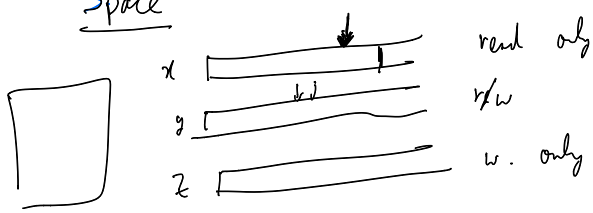
If M_i halts with output 0, set $B(y) = 1$ for some $y \in \{0,1\}^{t_i}$.

Claim: $f \notin P^B$.

Suppose M_{1000} computes f in time $O(n^2) = cn^2$

$\Rightarrow \exists j$ s.t $M_j = M_{1000}$ and $cn^2 \leq 2^{j/10}$

Space



Space : max value of j.

$L = DSPACE(\log(n))$

$NL = NSPACE(\log(n))$.

$PSPACE = \bigcup_c PSPACE(n^c)$.

$\underbrace{NPSPACE = PSPACE}_{\text{will show.}}$

Claim: Suppose $f \to s_1(n) \geq \log n$ space
$g \to s_2(n) \geq \log n$ space

$\Rightarrow f(g(x))$ can be computed in $O(s_1(n)+s_2(n))$ space.