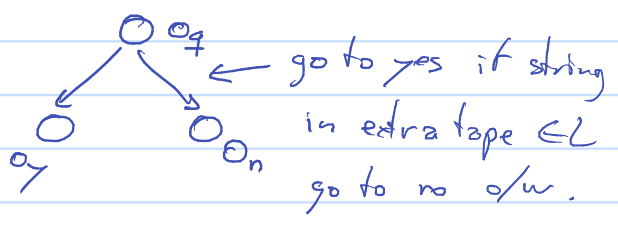
Oracles. Have a language $L$, TM with oracle $L$:

→ add extra tape

→ 3 extra states $O_q$, $O_y$, $O_n$.




if we add $L \in P$.

$M^L$ can be simulated with $N$ w/poly overhead: each time $M$ reaches state $O_q$. then $N$ can solve $L$, go to yes/no.

if we add $L \in NP$.

$M$ is a nondet. TM. $M^{SAT}$. can solve problems in NP and in coNP.

if we add EXP-complete language: e.g. $L = \{ \langle M, x, 1^n \rangle$ s.t. $M(x)$ runs for at most $2^n$ steps and accepts $\}$.

claim: both $P^L$ and $NP^L = EXP$.

$K \in EXP$. want to solve $K$ with $P^L$.

$\exists M$ that decides $K$ in time $2^{n^c}$.

can build TM $N^L$: input $x$: writes to oracle tape $\langle M, x, 1^{n^c} \rangle$

queries oracle.

gives same answer.

$NP^L \subseteq EXP$.

$M^L(x, y)$ ← certificate

TM $N$ will run in exp time.

for all possible $y$ of poly size:

simulate $M^L(x,y)$

$MP \subseteq EXP$

$NP = coNP$? $P = coP$

$EXP = coEXP$

$\not\subseteq P$ $\not\subseteq NP$-comp.

th $\exists$ NP-intermediate languages.

th $NTIME(n^a) \subsetneq NTIME(n^b)$ if $a < b$

$DTIME$ $DTIME$

→ bijection TH & integers.
→ simulate w/o large overhead.
] "black-box" proof. "relativizes"

proofs would also work with oracles. if $P^L \neq NP^L$ then $\exists$ int. $NP^L$ languages.

th: $\exists A, B$. s.t. $P^A = NP^A$ $P^B \neq NP^B$.

proof. (a) ✓ $P^A = NP^A = EXP$.

(bi).

$UB = \{ 1^n$ s.t. $\exists x \in B$ $|x| = n \}$.

$UB \in NP^B$. check $x = 1^n$ for some $n$.

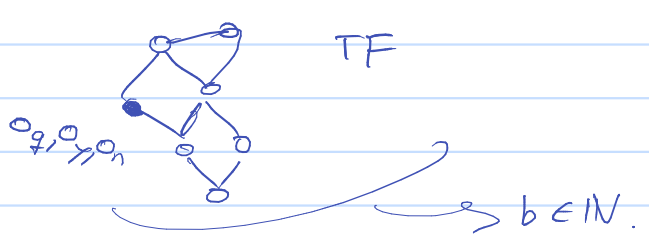
guess $y$ $|y| = |x|$.

query oracle $y \in B$?

choose $B$ s.t. $UB \not\subseteq P^B$. $\emptyset = B_0 \subseteq B_1 \subseteq B_2 \cdots \subseteq B$.

assume have $B_{i-1}$, have $TM_i^B$

i is indep. of B.




$B_{i-1}$ will have finitely many strings.

pick $n$ s.t. $n > |x|$ for any $x \in B_{i-1}$.

Run $TM_i^B$ on $1^n$ and make sure it answers incorrectly.

if $TM_i^B(1^n) = 1$ then $1^n \notin UB$ $TM_i$ incorrect on $1^{n_i}$

if 0 $\in UB$.

$TM_i^B$ make queries to strings: if query $x \in B^{i-1}$ then answer 1.

if query $x \notin B^{i-1}$ then answer 0

and record that $x \notin B$.

each TM makes poly. many queries. → at step $i$: $\leq i \cdot poly(n)$ strings known $\notin B$.

$\sum_{j=1}^{i} poly(n_j) \leq \sum poly(n) \leq i \cdot poly(n)$.

if $TM_i^B$ said 1: then $B^i = B^{i-1}$ // no string of length $n$ in $B$.

$1^n \notin UB$ ✗.

if $TM_i^B$ said 0: $2^n$ possible strings. poly(n) strings ruled out.

pick some string not yet ruled out. $B^i = B^{i-1} \cup \{z\}$.

$z$

$1^n \in UB$ ✗.

by construction: result of our simulation is same as if had really run TM with B.

therefore all $TM^B$ answer incorrectly. ⁂.

$NP^{SAT} = NP^{NP}$ $NP^{(NP^{NP})}$.

Polynomial Hierarchy: $NP$: $\{ x : \exists y \; M(x,y) = 1 \}$. $\Sigma^1 P$

$coNP$: $\{ x : \forall y \; M(x,y) = 0 \}$. $\Pi^1 P$

$\Sigma^2 P$: $\{ x : \exists y \forall z \; M(x,y,z) = 1 \}$.

$\Pi^2 P$: $\{ \forall y \exists z \quad \}$.

$\Sigma^i P$: $\{ x : \exists y^1 \forall y^2 \exists y^3 \cdots Q y^i \; M(x, y^1 \cdots y^i) = 1 \}$.

$PH = \bigcup_i \Sigma^i P$. $PH \subseteq PSPACE$.

SAT $NP_c$ $\overline{SAT}$ $coNP_c$ $\Sigma^i SAT$: $\{ \varphi : \exists y^1 \forall y^2 \cdots Q y^i \; \varphi(y^1 \cdots y^i)$ true $\}$.

$L \in coNP$ iff $\bar{L} \in NP$. $\Sigma^1 SAT$: $\{ \varphi : \exists y \; \varphi(y)$ true $\}$.

$\Pi^1 SAT$: $\{ \varphi : \forall y \; \varphi(y)$ true $\}$.

$\Sigma^i SAT$ is $\Sigma^i P$-complete.

$\Pi^i SAT$ is $\Pi^i P$-complete.

PSPACE-complete problem: QBF: $\{ \varphi : \exists y^1 \forall y^2 \cdots \underline{Q} y^n \; \varphi(y^1 \cdots y^n)$ true $\}$. (Quantified Boolean Formula)

$\exists / \forall$.

PH ≠ PSPACE bc no PH-complete problem but QBF is PSPACE complete.

$P \neq NP$, $coNP \neq NP$.

conj. $\Sigma^i \neq \Sigma^{i+1}$ $\forall i$. "PH does not collapse".

prop if $\Sigma^i = \Sigma^{i+1}$ then $PH = \Sigma^i$.

prop if $\Sigma^i \neq \Sigma^{i+1}$ $\forall i$ then $PH \neq PSPACE$.

if $\exists$ PH-complete problem: $L$ $PH = \bigcup_i \Sigma^i$ $\exists i$ s.t. $L \in \Sigma^i$.

$L$ PH-hard ⇒

can reduce all $\Sigma^j$ $j > i$ to $L$. //