$PADSAT = \{ \langle \varphi, \underbrace{1 \dots 1}_{H(|\varphi|)} \rangle : \varphi \in SAT \}$

$H(n) = \min(\log\log n, G(n))$.

$G(n) = \min_i$ st $\forall x$ $\underline{|x| \le \log n}$ TM_i decide if $x \in PADSAT$ in time $\le i \cdot |x|^i$

1. is $H(n)$ computable? what is its time complexity?

for $i \in [\log\log n]$
for $x \in \{0,1\}^{\log n}$ $\leq n$
run TM_i on x for $i \cdot |x|^i$ $\leftarrow$ poly.
run some other TM to compute if $x \in PADSAT$. $\leftarrow$ poly.
if TM_i and TM agree: continue
o/w stop checking.
if all x were ok then return i.
if no TM ok return $\log\log n$.

$x = \langle \varphi, \underbrace{1 \dots 1}_{H(|\varphi|)} \rangle$ SAT $\log n$

$H(n) = \min(\log\log n, G(n))$.

$G(n) = \min_i$ st $\forall x$ $\underline{|x| \le \log n}$ TM_i decide if $x \in PADSAT$ in time $\le i \cdot |x|^i$

2. Assume $PADSAT \notin P$. show that $H(n) \to \infty$.

assume $\exists$ bounded subsequence of $H(n)$.
then $\exists$ constant subsequence of $H(n)$. $\to$ say constant is C.

claim TM_C can decide $PADSAT$ in time $C \cdot |x|^C$.

$\forall x \in PADSAT$ find elem of subseq that is $> 2^{|x|}$.
by def G, TM_C decide x in time $\le C \cdot |x|^C$.

3. Prove PADSAT is not NP-hard.

assume $f: SAT \to PADSAT$
$\varphi \mapsto \langle \psi, 1 \rangle$ f runs in polynomial time.
$n^{H(|\psi|)}$

$PADSAT \notin P$ $H(n) \to \infty$. $\to n^{H(|\varphi|)}$ grows faster than any polynomial.

use f to solve SAT. $|\psi| < |\varphi|$.

$\exists N$, st $\forall \varphi$ $|\varphi| > N$

$f: \varphi \mapsto \underbrace{\langle \psi, \overbrace{1}^{n^{H(|\psi|)}} \rangle}_{poly}$ $n = |\varphi|$

if $P \neq NP$:
$P \subsetneq P_1 \subsetneq P_2 \subsetneq \dots \subsetneq NP$.

start with φ.
apply $f(\varphi)$, get $\langle \varphi', \cancel{1} \rangle$
$f(\varphi')$, get φ''.
$\vdots$
until $\varphi^{(k)}$ is no longer smaller than $\varphi^{(k-1)}$
brute force $\varphi^{(k)}$.

Language is unary if subset of $\{1\}^*$. 11, 1111, ...

claim: if every unary NP language is in P, then $EXP = NEXP$.

take $L \in NEXP$ $x \in L$ $1x$ $\underbrace{1 \dots 1}_{bin(1x)}$

$UnaryL$ / $UL = \{ \underbrace{1 \dots 1}_{bin(x)} : x \in L \}$

$UL \in NP \to UL \in P$.

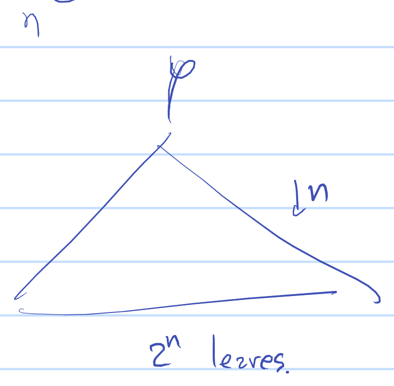
If some unary language L is NP-hard then $P = NP$.

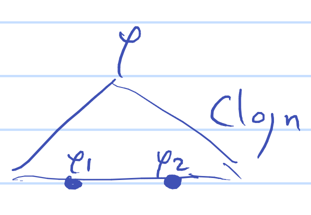
want algorithm for SAT in polynomial time.

$f: SAT \to L$
$\varphi \mapsto 1^m$ $m = poly(|\varphi|)$

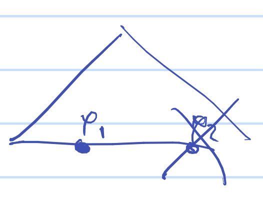
$\varphi \to \varphi|_{x_1=0} \to \varphi|_{x_1=0, x_2=0}$, $\varphi|_{x_1=0, x_2=1}$; $\varphi|_{x_1=1}$

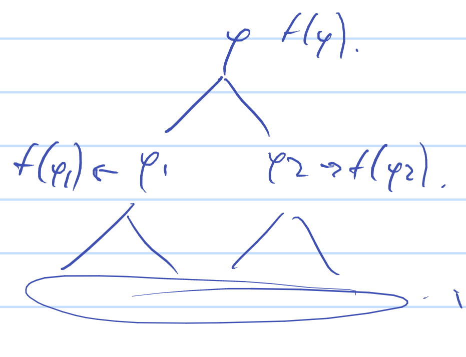
n^c





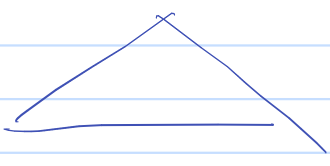
$f(\varphi_1) = f(\varphi_2)$.





$f(\varphi_1)$

if repeated labels, then drop rightmost formula.



because only have n^c different labels
only have layers of width n^c.

$\to \varphi$ on n variables.

have set all variables.

$\varphi|_{x_1 \dots x_n = b} = \langle^T_F$