- Let us study an execution of N on x.
- Say, N makes the bit choices,

$$c_1, c_2, \ldots, c_m \in \{0,1\}.$$

- Say, N queries SAT on the formulas, $\varphi_{\bar{c},1}, \varphi_{\bar{c},2}, \ldots, \varphi_{\bar{c},k}$ & gets answers, $a_1, a_2, \ldots, a_k \in \{0,1\}$.

- Then, $x \in L$ iff $\exists c_1, \ldots, c_m, a_1, \ldots, a_k$, ($N$ accepts x on the path $\langle c_1, \ldots, c_m \rangle$ & $\langle a_1, \ldots, a_k \rangle$ are the correct answers).

iff

$$\exists \bar{c}, \bar{a}\ \overbrace{\exists u_1, \ldots, u_k}^{\text{for YES answers}}\ \overbrace{\forall v_1, \ldots, v_k}^{\text{for NO answers}} \text{ s.t.}$$

N accepts x on the path $\bar{c}$ & the answers $\langle a_1, \ldots, a_k \rangle$ AND

$\forall i \in [k], (a_i = 1 \Rightarrow \varphi_{\bar{c},i}(u_i) = 1)$ AND

$\forall i \in [k], (a_i = 0 \Rightarrow \varphi_{\bar{c},i}(v_i) = 0)$.

$\Rightarrow L \in \Sigma_2$. $\Rightarrow NP^{SAT} \subseteq \Sigma_2$.

$\Rightarrow \Sigma_2 = NP^{SAT} = NP^{NP}$. $\square$

- Exercise: Complete the pf. for $i>2$.

- Thus, $\Sigma_2 = NP^{NP}$, $\Sigma_3 = NP^{NP^{NP}}$, ...

- Note: $P^P = P$ but NP^{NP} is conjectured to be harder than NP.

Between PH & Pspace: Counting

- Define #SAT: $\{\text{boolean formula } \phi\} \to \mathbb{N}$
  $\phi \mapsto$ #sat. assgn. of ϕ.

- Is #SAT eff. computable?

Definition: FP $:= \{ f \mid f$ is a fn. $\{0,1\}^* \to \mathbb{N}$, computable by a poly-time TM $M_f \}$.

Open: #SAT $\notin$ FP?

– #SAT motivates the defn. of #P:

Defn: A fn. $f:\{0,1\}^* \to \mathbb{N}$ is in **#P** if there is a poly-time TM M & a constant $c>0$ s.t. $\forall x$, $f(x) = \#\{y \in \{0,1\}^{|x|^c} \mid M(x,y)=1\}$.

$\Rightarrow$ #P is the collection of functions that **count** the #accepting-paths of an efficient NDTM.

Proposition: (i) $\#SAT \in \#P$.

(ii) $FP \subseteq \#P$. Open: $FP \neq \#P$?

(iii) $NP \subseteq P^{\#P} \subseteq Pspace$.

Using a fn. as an oracle.

prob. poly-time

– #P has an analogous decision version:

Defn: A lang. $L \in$ **PP** if $\exists$ poly-time TM M & a constant $c>0$ s.t. $\forall x$, $x \in L$ iff $\#\{y \in \{0,1\}^{|x|^c} \mid M(x,y)=1\} \geq \frac{1}{2} \cdot 2^{|x|^c}$.

- A function in PP computes the most-significant-bit of the corresponding fn. in #P !

Proposition: (1) $\#SAT \in FP^{PP}$.

(2) $P^{\#P} = P^{PP}$.

Proof:

(1) Let ϕ be a boolean formula in vars. $x_1, \ldots, x_n$.

- Using PP we could know whether #(sat.-assgn. of ϕ) $\geq 2^{n-1}$ or not.
- Wlog we assume it $\geq 2^{n-1}$.

(else we work with $\neg\phi$.)

- Think of $\{0,1\}^n$ to be lexicographically ordered.

For X_1 query PP at the n-variate formula $\phi \wedge (x_1 = 1)$.

- We want to find a string X s.t. there are exactly 2^{n-1} strings $y > X$ with $\phi(y) = 1$.

(X can be found in n calls to PP.)

- Note that for $x < X$, $x_1 = 0$.

$\Rightarrow$ The boolean formula $\psi :=$
$\phi \wedge (x < X) \wedge (x_1 = 0)$
has #(sat. assgn.) = #(sat. assgn. of ϕ) $-$
2^{n-1}.

(n-1)-variate

- Thus, the answer of the PP-oracle on ψ gives us the <u>2nd</u> most-significant-bit of #(sat. assgn. of ϕ).

- By repeating this process we can compute <u>all</u> the bits of #(sat. assgn. of ϕ).

$\Rightarrow$ $\#SAT \in FP^{PP}$.

(2) • We have shown $P^{\#SAT} \subseteq P^{PP}$.
- Later we will show that #SAT is #P-complete.

$\Rightarrow$ $P^{\#P} \subseteq P^{PP}$.

- The other direction is trivial. $\square$