Dec 4 RECAP

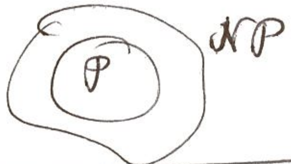
→ **Def:** $\mathcal{P}$: set of all problems that can be solved by a poly time algo.

→ **Def:** $\mathcal{NP}$: $Y \in \mathcal{NP}$ iff ∃ an efficient verifier B_Y s.t. ∀ inputs w:

(i) $w \in Y \Rightarrow \exists$ a witness t s.t. $|t| \le \text{poly}(|w|)$ and $B_Y(w,t) = 1$

(ii) $w \notin Y \Rightarrow \forall$ witness t s.t. $|t| \le \text{poly}(|w|)$, $B_Y(w,t) = 0$

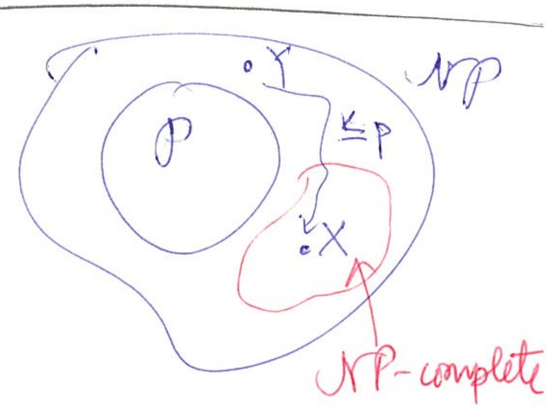
→ $\mathcal{P} \subseteq \mathcal{NP}$



Def: X is $\mathcal{NP}$-complete if

(i) $X \in \mathcal{NP}$

(ii) $\forall\ Y \in \mathcal{NP}$, $Y \le_p X$

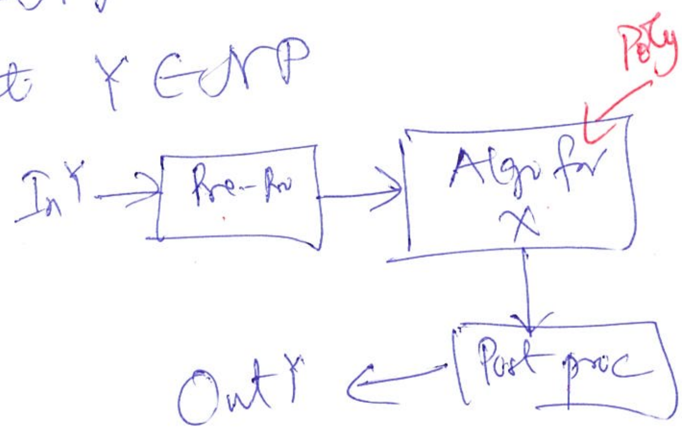


Claim 1: Let X be $\mathcal{NP}$-Complete

If $X \in \mathcal{P} \Rightarrow \mathcal{P} = \mathcal{NP}$

Pf idea: If $X \in \mathcal{P}$, Let $Y \in \mathcal{NP}$

$Y \le_p X$



$\Rightarrow$ Since $X \in \mathcal{P} \Rightarrow \exists$ a poly time algo for X

$\Rightarrow Y \in \mathcal{P}$

$\Rightarrow \mathcal{NP} \subseteq \mathcal{P}$ ▫

$\Rightarrow \mathcal{P} = \mathcal{NP}$.

$\because \mathcal{P} \subseteq \mathcal{NP}$

THM: IS is $\mathcal{NP}$-complete

THM2: (1) $IS \le_p VC$ (2) $VC \le_p IS$

Lemma 1: Let $G = (V, E)$.

S is an IS $\iff$ $V \setminus S$ is a VC

Pf (1) of THM2: $IS \le_p VC$

Pf: Given G, k for IS

Prec: compute $n-k$ from k — no post-proc

$(G,k) \xrightarrow{G,\ n-k}$ Algo for VC $\longrightarrow$ same output for IS

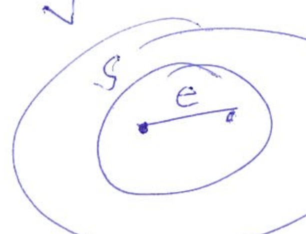
G has an IS of size $\ge k$ $\iff$ G has a VC of size $\le n-k$.

Redux for (2) is same.

Pf (idea) of Lemma 1:

S is IS $\Longrightarrow$ $V \setminus S$ is VC

Pf by contradiction



S is an IS (contradict)

by

$V \setminus S$ is not a VC.

$\Rightarrow \exists$ an edge e that does not have either end point in $V \setminus S$

$\Rightarrow e$ is "contained" in S $\Rightarrow$ S is not an IS

$\Longleftarrow$ is similar

Satisfiability / SAT problem

General: SAT formula on variables $X = \{x_1, \ldots, x_n\}$ (Boolean)

↳ AND clauses

↳ OR of literals

↳ literal $\in \{x_i, \overline{x_i}\}$

E.g. $\Phi_0 = (x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (x_2 \vee \overline{x_3})$

Generally: $c_1 \wedge c_2 \wedge \cdots \wedge c_m$ $\quad c_i$: clauses

$\equiv c_1, c_2, \ldots, c_m$

Clause c_i: OR of literals $t_1 \vee t_2 \vee \cdots \vee t_\ell$

each $t_i \in \{x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}\}$

Assignment: $a : X \to \{0, 1\}$ (0 = F, 1 = T)

$n = 3$

$x_1 = 0$	1	0
$x_2 = 0$	1	0
$x_3 = 0$	1	1

① $\Phi_0(\overset{x_1}{0}, \overset{x_2}{0}, \overset{x_3}{0}) = (0 \vee \overline{0}) \wedge (\overline{0} \vee \overline{0}) \wedge (0 \vee \overline{0})$

$= (0 \vee 1) \wedge (1 \vee 1) \wedge (0 \vee 1)$

$= 1 \wedge 1 \wedge 1 = 1$

$\Rightarrow (0, 0, 0)$ is a satisfy assignment for Φ_0

② $\Phi_0(1, 1, 1) = (1 \vee \overline{1}) \wedge (\overline{1} \vee \overline{1}) \wedge (1 \vee \overline{1})$

$= (1 \vee 0) \wedge (0 \vee 0) \wedge (1 \vee 0)$

$= 1 \wedge 0 \wedge 1 = 0$