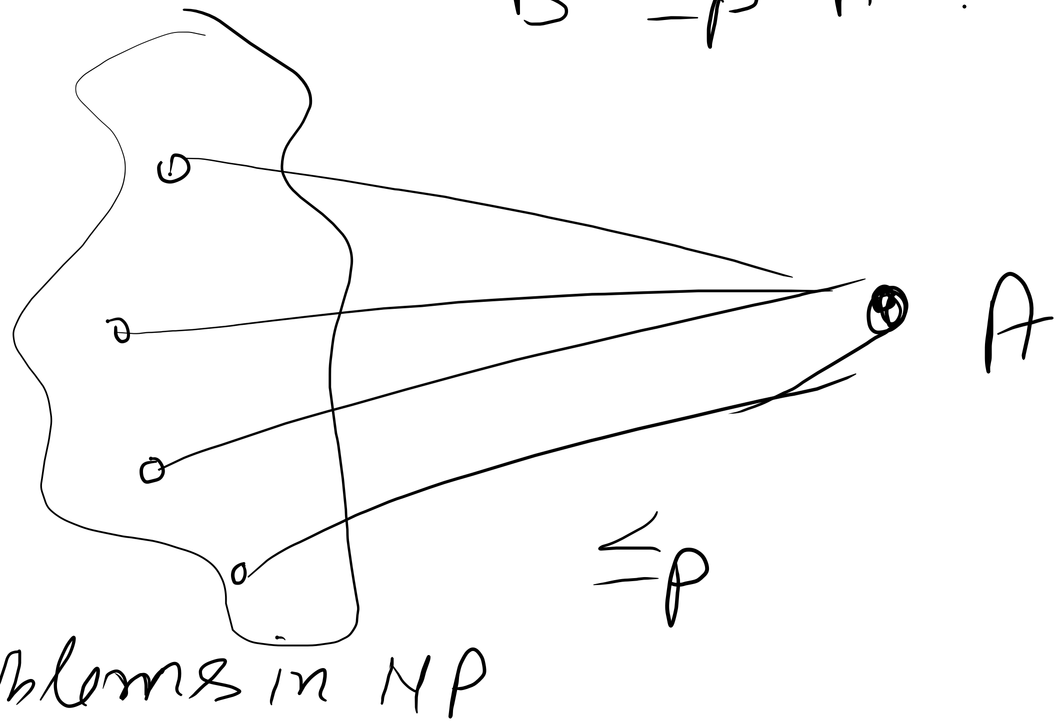
class P
class NP } polynomial time reduction.

<u>class NP-hard</u>

A decision problem A is in class NP-hard if <u>for all</u> problems B in NP

$$B \le_p A.$$

<u>Pictorially</u>



NP-complete

A decision problem A is in class NP-complete if

i) $A \in NP$

ii) A is NP-hard. (all problems $B \in NP$ $B \leq_p A$)

Ex^m satisfiability or circuit satisfiability. (SAT)

1971 → cook and Levin

They ~~proved~~ every problem in NP is polynomial time reducible to SAT.

SAT: ϕ is a formula in CNF form with n variables $x_1, x_2, \ldots, x_n$ and m clauses $c_1, c_2, \ldots, c_m$.

Decide whether ϕ is satisfiable or not.

~~to~~ Assigning 0/1 values to the variables and evaluate ϕ.

Ex^m

$$\phi = c_1 \wedge c_2 \wedge c_3 \wedge \cdots \wedge c_m$$

where each $c_i = (x_{i_1} \vee x_{i_2} \vee \cdots \vee x_{i_k})$

$$\phi = (x_1 \vee x_2 \vee \overline{x_3}) \wedge (x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_4)$$

$$x_1 = 1, \quad x_2 = 1, \quad x_3 = 0, \quad x_4 = 1$$

$\underline{\underline{Th^m}}$ A is a decision problem such that $B \le_p A$, for some NP-complete problem B, then A is NP-hard.

$\underline{\underline{Proof}}$ B is NP-complete.

For all problems C is in NP.

$$C \le_p B \quad \text{——— (1)}$$

we also have $B \le_p A$, — (2)

Transitivity (1) & (2)

$$C \le_p A.$$

that means all problems in NP is polynomial time reducible to A. $\Rightarrow$ A is NP-hard.

Alternative definition of NP-complete

A decision problem A is in class NP-complete if

i) A is in NP

ii) $\exists$ an NP-complete problem B such that $B \leq_p A$, $\Big\}$ NP-hard.

Methods to prove a problem A is NP-complete

1) Prove $A \in NP$.

2) select a known suitable NP-complete problem B

3) Describe an algorithm that computes a function f that maps each instance x of B to an instance f(x) of A.

4) Prove that the function satisfies $x \in B$ iff $f(x) \in A$ $\forall x \in B$

5) Prove that f can be computed in polynomial time.

$\underline{\underline{Ex^m}}$ Prove that 3-SAT is NP-complete.

3-SAT: $\phi = (\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge (x_2 \vee \bar{x}_3 \vee x_4) \wedge (x_1 \vee \bar{x}_2 \vee x_4)$

Each clause contains exactly 3 literals.

1) 3-SAT $\in$ NP.

Consider a certificate: Given an assignment to the variables. $x_1 = 0,\ x_2 = 1,\ x_3 = 0,\ x_4 = 1$

In ϕ, use this assignment, evaluat ϕ.

It takes poly-time (in n and m).

2) choose a suitable NP-complete problem say SAT.

$$SAT \leq_p 3\text{-}SAT$$

3) ϕ be an instance of SAT, F_ϕ be an instance of 3SAT

&

4)

$$\phi \xrightarrow{f} F_\phi$$

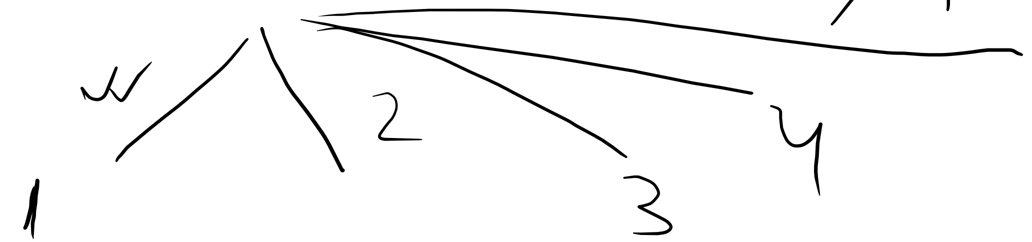
$$\phi = C_1 \wedge C_2 \wedge C_3 \wedge \cdots \wedge C_m$$

C_i's have arbitrary sizes.

$$F_\phi = C_1' \wedge C_2' \wedge \cdots \wedge C_{m'}'$$

c_i''s are exactly size-3.

consider a clause C_i: it has literals ≥ 4.



Assume $C_i = x$

replace C_i by 4 clauses as follows.

$$F_{C_i} = (x \vee z_1 \vee z_2) \wedge (x \vee z_1 \vee \bar{z}_2) \wedge (x \vee \bar{z}_1 \vee z_2) \wedge (x \vee \bar{z}_1 \vee \bar{z}_2)$$

C_i is evaluate to 1 then take any value of z_1 and z_2

then F_{C_i} is satisfiable.

$C_i = (x \vee y)$

~~re~~place by $F_{C_i} = (x \vee y \vee z) \wedge (x \vee y \vee \bar{z})$

C_i is evaluate to 1 then any value of z makes F_{C_i} satisfiable.

$C_i = (x \vee y \vee z)$

not doing anything

$c_i = (x \vee y \vee z \vee w)$

replace c_i by

$f_{c_i} = (x \vee y \vee t) \wedge (z \vee w \vee \bar{t})$

if c_i evaluates to 1 then (if x or y evaluate to 1 then take $t = 0$)

→ else if z or w evaluate to 1 then take $t = 1$)

$C_i = (x_1 \vee x_2 \vee x_3 \vee \cdots \vee x_k)$

$F_{C_i} = (x_1 \vee x_2 \vee z_1) \wedge (\bar{z}_1 \vee x_3 \vee z_2) \wedge (\bar{z}_2 \vee x_4 \vee z_3) \wedge \cdots$

$\cdots \wedge (\bar{z}_{k-3} \vee x_{k-1} \vee x_k)$

Assume C_i is satisfiable, let x_t be 1

we consider $z_1, z_2, \ldots, z_{t-2} = 1$

$z_{t-1}, z_t, \ldots, z_{k-3} = 0$

5) <u>Polynomial time</u>

F_ϕ contains polynomial in $(n \& m)$ number of clauses and variables.