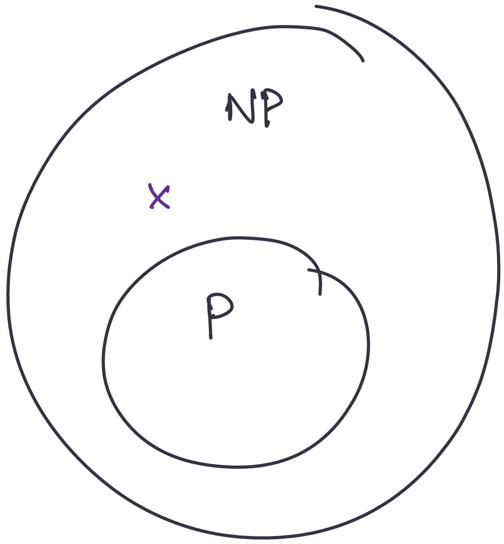
Reductions (Contd.)

- Reduction from CktSAT to 3SAT.



If CktSAT $\leq_p$ 3-SAT

then 3-SAT is NP-hard

→ Every problem in NP can be solved using Ckt-SAT and every instance of CktSAT can be solved using 3-SAT ⇒ Every problem in NP can be solved using 3-SAT.

$$\begin{array}{ccccc} & & N & & N^d \\ \Pi & \xrightarrow{n^{O(1)}} & \text{CktSAT} & \xrightarrow{N^{O(1)}} & \text{3-SAT} \\ n & & n^c & \longrightarrow & n^{c \cdot d} \end{array}$$

$$N^d = \underline{\underline{(n^c)^d}}$$

$$n \xrightarrow[\underline{\underline{n^{O(1)}}}]{} n^{cd}$$

Thm: CktSAT $\leq_p$ 3-SAT

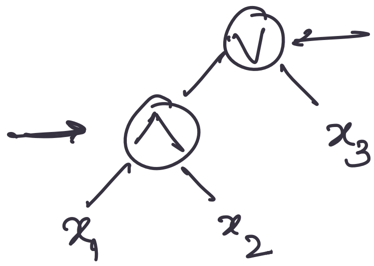
↑ Polynomial time reductions

Turing reduction
Cook reduction

$\rightarrow$ Any Instance of ckt-SAT $\underline{\underline{C}}$

no. of input vars is n

no. of Boolean gates $= n^{O(1)}$



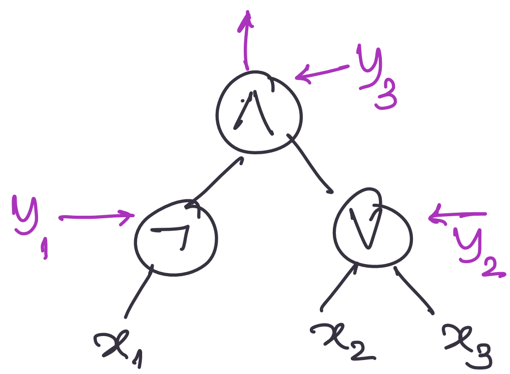
$(x_1 \wedge x_2) \vee x_3$

y_1, y_2

$$\left.\begin{aligned} y_1 &= x_1 \wedge x_2 \\ y_2 &= y_1 \vee x_3 \end{aligned}\right\}$$

$y_2 = (x_1 \wedge x_2) \vee x_3$

Suppose, my circuit has s many gates then we get s many constraints. (with s many new auxilliary variables)



$y_1 = \neg x_1, \overline{x_1}$

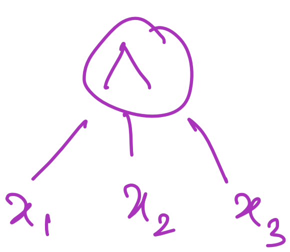
$y_2 = x_2 \vee x_3$

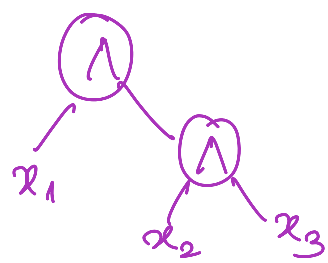
$y_3 = y_1 \wedge y_2$

W.L.O.G, we can assume that no. of inputs to any gate is ≤ 2.

If C is satisfiable, all constraints have to be satisfied simultaneously.

$\wedge \left(y_i = \sim \right)$





$z \sim y$ vars or x vars.

$$\overbrace{y_i}^{a} = \overbrace{z_k \wedge z_l}^{b} \text{ (or) } y_i = z_k \vee z_l \text{ (or) } y_i = \neg z_j$$

$$(y_i \vee \overline{(z_k \wedge z_l)}) \wedge (\bar{y}_i \vee (z_k \wedge z_l))$$

$$= (y_i \vee \bar{z}_k \vee \bar{z}_l) \wedge (\bar{y}_i \vee z_k) \wedge (\bar{y}_i \vee z_l).$$

$\underline{\underline{a = b}}$

a	b	a=b
0	0	1
0	1	0
1	0	0
1	1	1

$$(a \wedge b) \vee (\bar{a} \wedge \bar{b}) \checkmark$$

$$= (a \vee \bar{a}) \wedge (a \vee \bar{b}) \wedge (\bar{a} \vee b) \wedge (b \vee \bar{b})$$

$$= (a \vee \bar{b}) \wedge (\bar{a} \vee b)$$

$$\overset{n}{C} \longleftrightarrow \bigwedge_{g \in gates(C)} \overset{n^{O(1)}}{\text{Constraint}(g)}$$

Solution ≡ Satisfiable assignment

If C has solution then $\bigwedge_{g \in gates(C)} (\text{Constraints}(g))$ has a solution.

If $\bigwedge_{g \in gates(C)} (\text{Constraints}(g))$ has a solution so does C.

$\bigwedge_{g \in gates(C)} (\text{Constraints}(g))$ can be expressed as a CNF.

iff.

$$(a) \longrightarrow (a \vee z) \wedge (a \vee \bar{z}).$$

$(a \vee b) \longrightarrow (a \vee b \vee z) \wedge (a \vee b \vee \bar{z})$.

Ckt of size $n^{O(1)}$ $\longrightarrow$ 3-SAT with at most $n^{O(1)}$ many clauses.

Ckt C is sat iff 3-SAT instance is sat.

<u>Thm</u>: 1. 3-SAT is NP-hard
2. 3-SAT is in NP

} 3-SAT is NP-complete.

Loosely speaking, CktSAT is believed not to have algorithms smaller than $2^{\delta n}$ for a constant $\delta \in (0,1]$.

↑ Exponential time hypothesis.

<u>MaxSAT</u>: Output the no. of satisfiable clauses in a given CNF.

$$C_1 \wedge C_2 \wedge \cdots \wedge C_m$$

$$\max_{\substack{\vec{a} \in \{0,1\}^n \\ \{T,F\}^n}} \{ \text{no. of clauses satisfied by } \vec{a} \}$$

3-SAT can be solved using MAXSAT.

↳ if algo for MAXSAT outputs m then 3-SAT is sat.
else not.

⇒ MaxSAT is NP-hard.

Polynomial time algorithm for MaxSAT ⇒ P=NP.

→ One way to deal with hard problems

↳ Approximation.

c-approximation for maximization problems:

An algorithm is c-approximation if in polytime we can output a value that is $\frac{OPT}{c}$ $(c \geq 1)$.

Randomized algorithms

↳ Algorithms that also use randomness to make decisions.

PRAS

Polynomial time randomized approximation schemes

$C_1 \wedge C_2 \wedge \cdots \wedge C_m$ ← Working with 3-CNF.

$i \in [1, m] : C_i = x_{i_1} \vee x_{i_2} \vee x_{i_3}$

$$X_i := \begin{cases} 1 & \text{if a random assignment satisfies } C_i \\ 0 & \text{o/w}. \end{cases}$$

$$\Pr[X_i = 1] = \frac{7}{8}$$

$$\mathbb{E}[X_i] = 1 \cdot \Pr[X_i = 1] + 0\,[\Pr[X_i = 0]] = \frac{7}{8}.$$

$\sum_{i=1}^{m} X_i \leftarrow$ no. of clauses that are sat by a random assignment.

$\mathbb{E}\left[\sum_{i=1}^{m} X_i\right] \leftarrow$ Exp no. of clauses that are sat by a random assignment.

$$\parallel$$

$$\sum_{i=1}^{m} \mathbb{E}[X_i] = \sum_{i=1}^{m} \frac{7}{8} = \frac{7}{8} m.$$

Markov's inequality

$$\Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}$$

$$\Pr\left[\sum_{i=1}^{m} X_i < \frac{m}{2}\right]$$

$$\underline{\underline{Y}} = m - \sum_{i=1}^{m} X_i$$