Dec 1

SAT formula

AND / conjunction of clauses

$\hookrightarrow$ OR / disjunction of literals $\hookrightarrow x_i, \overline{x_i}$

E.g. $(\underset{\text{OR}}{x_1 \vee} \overline{x_2}) \wedge (\overline{x_1} \vee \overline{x_3}) \underset{\text{AND}}{\wedge} (x_2 \vee \overline{x_3})$ —— (*)

generally: $C_1 \wedge C_2 \wedge \dots \wedge C_m$ $\leftarrow$ m clauses

$\equiv \quad C_1, C_2, \dots, C_m$ $\quad X = \{x_1, \dots, x_n\}$ $\leftarrow$ set of variables

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

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

Assignment: $v: X \to \{\underset{F}{0}, \underset{T}{1}\}$ $\quad n=3$

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

$\boxed{\{F,T\} \to \{0,1\}}$

assignments $= 2^n$ $\leftarrow$ 2 choices for each x_i

An assignment v satisfies a SAT formula Φ, if Φ evaluates to true on the assignment

$\equiv$ assignment satisfies ALL ~~cla~~ clauses

Ex: $(0,0,0)$ (x_1, x_2, x_3)

$x_1 \vee \overline{x_2} = 0 \vee \overline{0} = 0 \vee 1 = 1$

$\overline{x_1} \vee \overline{x_3} = \overline{0} \vee \overline{0} = 1 \vee 1 = 1$

$x_2 \vee \overline{x_3} = 0 \vee \overline{0} = 0 \vee 1 = 1$

$\Rightarrow (0,0,0)$ is a satisfying assignment for (*)

Ex: $(1,1,1)$

$x_1 \vee \overline{x_2} = 1 \vee \overline{T} = 1 \vee 0 = 1$

$\overline{x_1} \vee \overline{x_3} = \overline{T} \vee \overline{T} = 0 \vee 0 = 0$

$\Big\}$ $(1,1,1)$ is NOT a satisfying assignm[ent]

Ex: $(0,0,1)$

$x_1 \vee \overline{x_2} = 0 \vee \overline{0} = 0 \vee 1 = 1$ $\quad x_2 \vee \overline{x_3} = 0 \vee \overline{T} = 0 \vee 0 = 0$

$\overline{x_1} \vee \overline{x_2} = \overline{0} \vee \overline{T} = 1 \vee 0 = 1$ $\Rightarrow (0,0,1)$ is NOT a satisfying assignment

Q: Given a SAT formula Φ, $\exists$ exist a satisfying assignment to Φ?

← Is Φ satisfiable?

3-SAT formula: A SAT formula $\Phi = C_1, \ldots, C_m$ s.t. each clause C_i has EXACTLY 3 literals.

3-SAT problem

Input: 3-SAT formula Φ

Output: True / 1 if Φ is satisfiable?

Trivial ~~alo~~ algo: Try out all 2^n possible assignments & check if any of them satisfy Φ $(O(m \cdot 2^n))$

Many hardness reduction: 3-SAT $\leq_p$ your problems

THM: 3-SAT $\leq_p$ IS → Input: $G; k$
output: 1/True if G has an IS of size $\geq k$.
↑
$S \subseteq V$ s.t no edges are in S

We'll show: Given any 3-SAT formula $\Phi \xrightarrow[\text{in poly time}]{} G; k$

s.t Φ is satisfiable $\iff$ G has IS of size $\geq k$.

$\Phi \to \boxed{\text{Pre-proc}} \xrightarrow{G;k} \boxed{\text{Algo for IS}} \xrightarrow{0/1}$ output same 0, 1

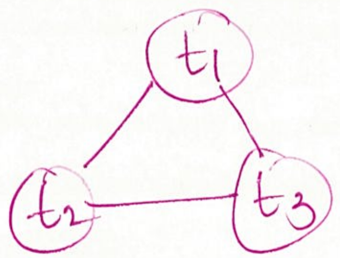
Reduction idea: Use a "gadget"

2 equiv. ways of looking at 3-SAT

→ Make 0/1 choices for $x_1, \dots, x_n$ s.t. it satisfies ≥ 1 literal in each clause.

→ Pick one literal from each clause s.t. you do NOT pick two literals that conflict ($x_i, \bar{x}_i$ for some i)

Gadget: $C = t_1 \vee t_2 \vee t_3$



Only IS are $\{t_1\}, \{t_2\}, \{t_3\}$

Idea: Each choice of an IS in this gadget $\equiv$ picking a literal from C.

Redux: Given $\Phi = C_1, \dots, C_m$ ⟶ (G, m)

3-SAT → Φ

Φ is satisfiable $\Longleftrightarrow$ G has an IS of size $\geq m$

Step 1: Replace each clause C_i by its triangle

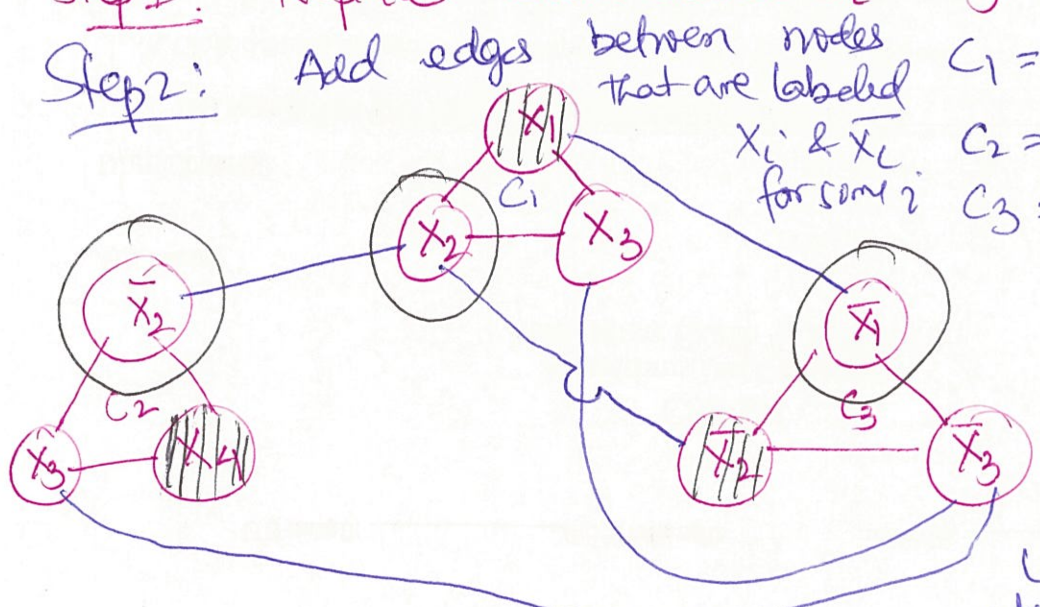
Step 2: Add edges between nodes that are labeled x_i & $\bar{x}_i$ for some i

$n = 4$

$C_1 = x_1 \vee x_2 \vee x_3$ ✓

$C_2 = \bar{x}_2 \vee x_3 \vee x_4$ ✓

$C_3 = \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3$ ✓



Pick one vertex / triangle → all IS of size 3.

IS $= \{x_1, x_4, \bar{x}_2\}$

satisfy ↳ $(1, 0, ?, 1)$

THM: Φ is satisfiable $\Longleftrightarrow$ G (as constructed above) has IS of size $\geq m$.