May 8

SAT formula

↳ AND/conjunction of clauses
↳ OR/disjunction of literals. ↳ $x_i, \bar{x}_i$

Ex: $(x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_3) \wedge (x_2 \vee \bar{x}_3)$ → (*)

AND   OR

In general: $C_1 \wedge C_2 \wedge \cdots \wedge C_m$
$\equiv C_1, C_2, \cdots, C_m$ } m clauses ↓ use variable literal ↓ $X = \{x_1, \ldots, x_n\}$

clause: OR of literals: $t_1 \vee t_2 \vee \cdots \vee t_k$

$t_i \in \left\{ \begin{matrix} x_1, \ldots, x_n, \\ \bar{x}_1, \ldots, \bar{x}_n \end{matrix} \right\}$

Assignment: $v: X \to \{0, 1\}$

assignments: 2^n ← 2 choices for each x_i

Ex: $n = 3$

$x_1 = 0$	1	0	⋮	$\{T/F\}$
$x_2 = 0$	1	0	⋮	$\to \{1/0\}$
$x_3 = 0$	1	1	⋮	

An assignment satisfies a SAT formula φ, if φ evaluates to true/1 on that assignment.

$\equiv$ assignment satisfies ALL clauses.

Ex: $(0,0,0)$

$x_1 \vee \bar{x}_2 = 0 \vee \bar{0} = 0 \vee 1 = 1$
$\bar{x}_1 \vee \bar{x}_3 = \bar{0} \vee \bar{0} = 1 \vee 1 = 1$
$x_2 \vee \bar{x}_3 = 0 \vee \bar{0} = 0 \vee 1 = 1$

} $1 \wedge 1 \wedge 1 = 1$
$(0,0,0)$ is a satisfying assignment for (*).

$(1,1,1)$:

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

$1 \wedge 0 \wedge 1 = 0$

$(1,1,1)$ is not a satisfying assignment for (*).

$(0,0,1)$:

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

$1 \wedge 1 \wedge 0 = 0$

$(1,1,0)$ is NOT a satisfying assignment for (*)

Q: Given a SAT formula φ, $\exists$ a satisfying assignment for φ?

$\equiv$ is φ satisfiable?

3-SAT formula: A SAT formula where each clause C_i has EXACTLY 3 literals.

3-SAT Problem:

Input: A 3-SAT formula φ

Output: ~~1/T~~ if φ is satisfiable.

Brute-force/Naive algo: Try all possible 2^n assignments and check if any of them satisfies φ. $\left(O(m \cdot 2^n)\right)$

To prove your problem is Hard:

3-SAT $\le_P$ your problem.

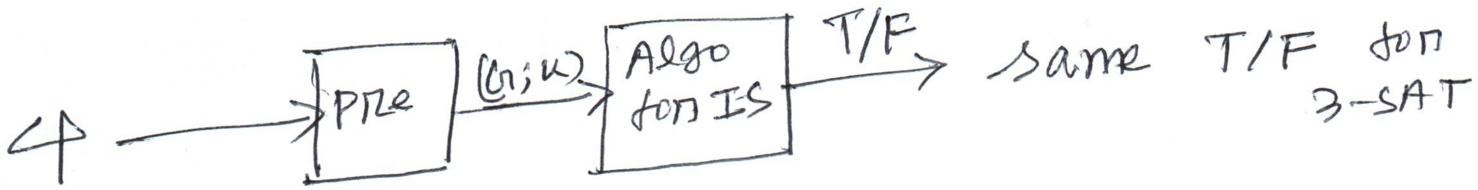
↳ " is as hard as 3-SAT.

THM: 3SAT $\le_p$ IS $\to$ (i/P: $(G; K)$
O/P: 1/T if G has an IS of size $\ge K$).

We will show that given a 3-SAT formula

$$\varphi \xrightarrow[\text{in poly-time}]{\text{reduce}} (G; K) \text{ s.t.}$$

φ is satisfiable $\iff$ G has an IS of size $\ge K$.



---

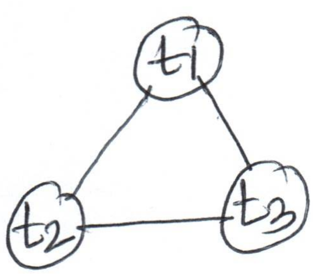
* 2-ways to look at 3-SAT.

① Make 0/1 choices for literals $x_1, \ldots, x_n$ s.t. it satisfies ≥ 1 literals in every clause.

② Pick one literal from each clause s.t. you do not pick two literals that conflict.
$\hookrightarrow x_i, \overline{x_i}$ conflict

Reduction IDea: Use a "gadget".

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



IS: $\{t_1\}, \{t_2\}, \{t_3\}$

IDea: Each choice of an IS $\equiv$ picking a literal from clause C.