Formula Satisfiability (SAT)

Given a boolean formula Φ.
Is there a way to set Φ's variable so it evals to true?

SAT $\in$ NP: How do I set the variables?

SAT is NP-hard (so SAT $\in$ NP-complete)

Reduce CircuitSAT <u>to</u> SAT.

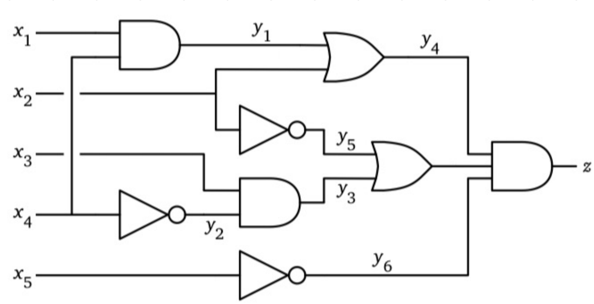
Given a circuit K.

Assign each wire a variable.

Write an equation Φ describing each gate.

$(a \wedge b = c)$

$\wedge$ the equations & the variable z for the output



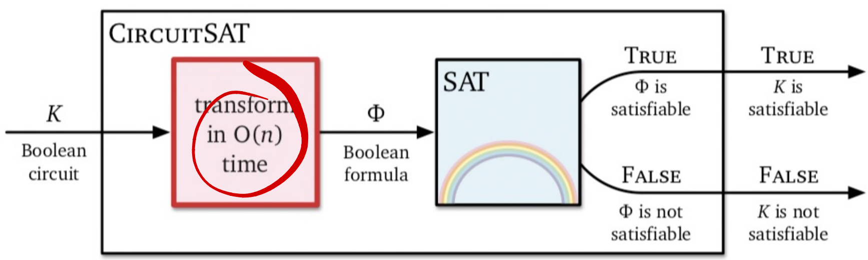
$$(y_1 = x_1 \wedge x_4) \wedge (y_2 = \overline{x_4}) \wedge (y_3 = x_3 \wedge y_2) \wedge (y_4 = y_1 \vee x_2) \wedge$$
$$(y_5 = \overline{x_2}) \wedge (y_6 = \overline{x_5}) \wedge (y_7 = y_3 \vee y_5) \wedge (z = y_4 \wedge y_7 \wedge y_6) \wedge z$$

Lemma: K is sat. iff ϕ is.

$\Rightarrow$: Use K's good setting for x values & set other variables by what their gates do.

All equations are sat, & z is set true, so ϕ is true.

$\Leftarrow$ Use Φ's setting of x values.



CircuitSat(K):
transcribe K into a boolean formula Φ
return Sat(Φ) ⟨⟨★★★MAGIC★★★⟩⟩

So we have <u>poly time</u> reduction from CircuitSAT to SAT.

So any problem $A \in NP$ reduces to CircuitSAT which reduces to SAT.

$\Rightarrow$ SAT is NP-hard.

A **literal** is a boolean variable a or its negation $\bar{a}$.

A **clause** is a disjunction (OR) of several literals.

A boolean formula is in **conjunctive normal form** if it is the **conjunction** of several clauses.

$$\overbrace{(a \vee b \vee c \vee d)}^{\text{clause}} \wedge (b \vee \bar{c} \vee \bar{d}) \wedge (\bar{a} \vee c \vee d) \wedge (a \vee \bar{b})$$

A <u>3CNF formula</u> is a CNF formula with <u>exactly</u> three literals per clause.

3SAT(3CNF SAT):
Given boolean formula Φ in 3CNF. Can we set variables to make Φ true?

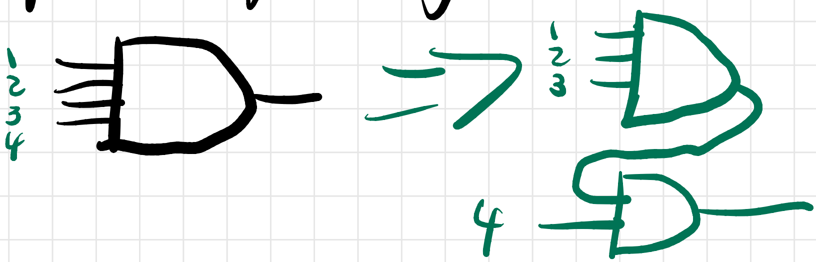
$3SAT \in NP$.

$3SAT \in NP$-complete.

Reduce from CircuitSAT.

Given cirwit K.

1) Change all AND & OR gates to a tree taking exactly two inputs per gate.



2) Assign variables to wires & write equations for gates.

3) Chang equations into 2 or 3 clauses each

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

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

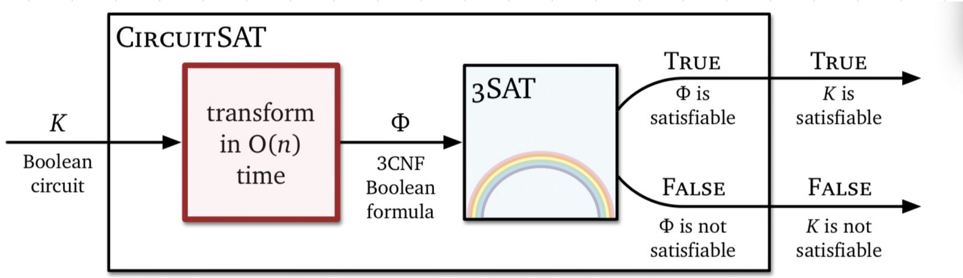
$$a = \bar{b} \quad \longmapsto \quad (a \vee b) \wedge (\bar{a} \vee \bar{b})$$

4) Replace clauses with 1 or 2 literals with 4 or 2 clauses with 3 literals. (Use a new x & y for each of these transforms)

$$a \vee b \longmapsto (a \vee b \vee x) \wedge (a \vee b \vee \bar{x})$$
$$a \longmapsto (a \vee x \vee y) \wedge (a \vee \bar{x} \vee y) \wedge (a \vee x \vee \bar{y}) \wedge (a \vee \bar{x} \vee \bar{y})$$

$$\begin{gathered}
(y_1 \vee \overline{x_1} \vee \overline{x_4}) \wedge (\overline{y_1} \vee x_1 \vee z_1) \wedge (\overline{y_1} \vee x_1 \vee \overline{z_1}) \wedge (\overline{y_1} \vee x_4 \vee z_2) \wedge (\overline{y_1} \vee x_4 \vee \overline{z_2}) \\
\wedge (y_2 \vee x_4 \vee z_3) \wedge (y_2 \vee x_4 \vee \overline{z_3}) \wedge (\overline{y_2} \vee \overline{x_4} \vee z_4) \wedge (\overline{y_2} \vee \overline{x_4} \vee \overline{z_4}) \\
\wedge (y_3 \vee \overline{x_3} \vee \overline{y_2}) \wedge (\overline{y_3} \vee x_3 \vee z_5) \wedge (\overline{y_3} \vee x_3 \vee \overline{z_5}) \wedge (\overline{y_3} \vee y_2 \vee z_6) \wedge (\overline{y_3} \vee y_2 \vee \overline{z_6}) \\
\wedge (\overline{y_4} \vee y_1 \vee x_2) \wedge (y_4 \vee \overline{x_2} \vee z_7) \wedge (y_4 \vee \overline{x_2} \vee \overline{z_7}) \wedge (y_4 \vee \overline{y_1} \vee z_8) \wedge (y_4 \vee \overline{y_1} \vee \overline{z_8}) \\
\wedge (y_5 \vee x_2 \vee z_9) \wedge (y_5 \vee x_2 \vee \overline{z_9}) \wedge (\overline{y_5} \vee \overline{x_2} \vee z_{10}) \wedge (\overline{y_5} \vee \overline{x_2} \vee \overline{z_{10}}) \\
\wedge (y_6 \vee x_5 \vee z_{11}) \wedge (y_6 \vee x_5 \vee \overline{z_{11}}) \wedge (\overline{y_6} \vee \overline{x_5} \vee z_{12}) \wedge (\overline{y_6} \vee \overline{x_5} \vee \overline{z_{12}}) \\
\wedge (\overline{y_7} \vee y_3 \vee y_5) \wedge (y_7 \vee \overline{y_3} \vee z_{13}) \wedge (y_7 \vee \overline{y_3} \vee \overline{z_{13}}) \wedge (y_7 \vee \overline{y_5} \vee z_{14}) \wedge (y_7 \vee \overline{y_5} \vee \overline{z_{14}}) \\
\wedge (y_8 \vee \overline{y_4} \vee \overline{y_7}) \wedge (\overline{y_8} \vee y_4 \vee z_{15}) \wedge (\overline{y_8} \vee y_4 \vee \overline{z_{15}}) \wedge (\overline{y_8} \vee y_7 \vee z_{16}) \wedge (\overline{y_8} \vee y_7 \vee \overline{z_{16}}) \\
\wedge (y_9 \vee \overline{y_8} \vee \overline{y_6}) \wedge (\overline{y_9} \vee y_8 \vee z_{17}) \wedge (\overline{y_9} \vee y_6 \vee z_{18}) \wedge (\overline{y_9} \vee y_6 \vee \overline{z_{18}}) \wedge (\overline{y_9} \vee y_8 \vee \overline{z_{17}}) \\
\wedge (y_9 \vee z_{19} \vee z_{20}) \wedge (y_9 \vee \overline{z_{19}} \vee z_{20}) \wedge (y_9 \vee z_{19} \vee \overline{z_{20}}) \wedge (y_9 \vee \overline{z_{19}} \vee \overline{z_{20}})
\end{gathered}$$

Reduction takes O(n) time.



So it's NP-hard.

Given undirected graph $G = (V, E)$.

An <u>independent set</u> $S \subseteq V$ of G has no edge of G between its vertices.

Max Ind Set: Find a max size independent set.

Claim: Max Ind Set is NP-hard.

Reduce from 3SAT.

Given 3CNF Φ.

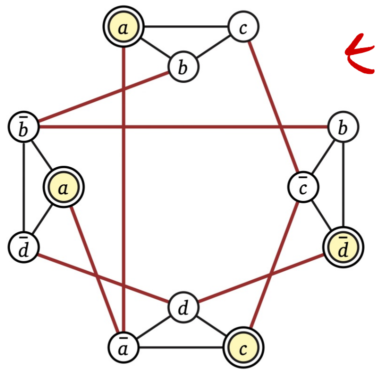
Build a graph G.

$k \leftarrow$ # clause in Φ.

G gets $3k$ vertices, one per literal of Φ.

Any pair of literals in a clause get a "triangle" edge between their vertices.

Any two literals a & $\bar{a}$ get a "negation" edge.



$\leftarrow G$

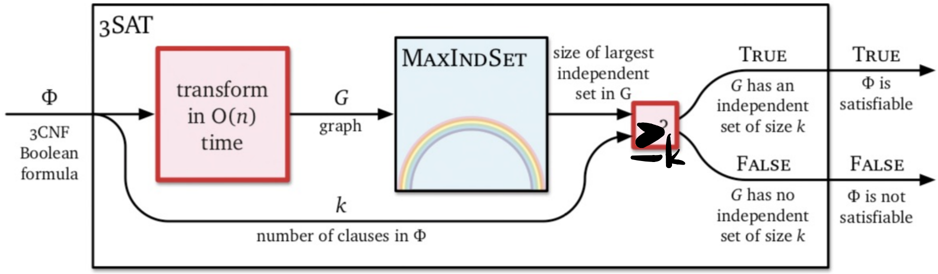
$\Phi = (a \vee b \vee c) \wedge (b \vee \bar{c} \vee \bar{d}) \wedge (\bar{a} \vee c \vee d) \wedge (a \vee \bar{b} \vee \bar{d})$

Claim: Φ is sat. if G has an ind. set of size k.

$\Rightarrow$ Pick a sat. assignment & one true literal (wrt assignment) per clause. S: vertices of those true literals.

$|S| = k$

$\Leftarrow$ Set variable so each ind set literal is true.



Decision version: Ind Set

Given G & an integer k.
Does G have an ind set of
size k?

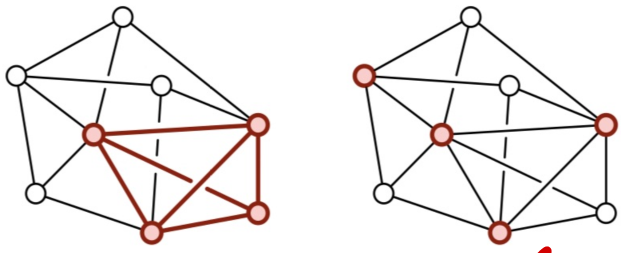
Ind Set $\in$ NP.
$\in$ NP-complete.

A <u>clique</u> is another name for a complete graph.

MaxClique: Given G, what is the largest clique subgraph?

A <u>vertex cover</u> is a subset of vertices where each edge is hit <u>at least</u> once.

Min Vertex Cover: Find a min size vertex cover.



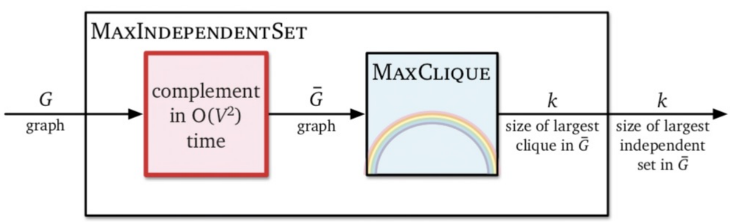
clique

vertex cover

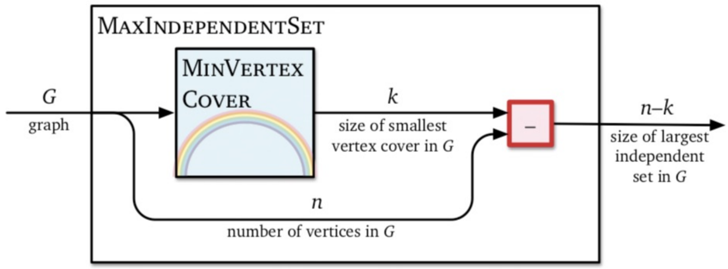
Claim: Both are NP-hard.

From MaxIndSet:

1) Given G. Take complement $\bar{G} = (V, \bar{E})$. $\bar{E} = \{uv : uv \notin E\}$



2) I is an ind. set of $G=(V,E)$ iff $V \setminus I$ is a vertex cover.



Decision versions hard too

$\Rightarrow$ Clique + Vertex Cover $\in$ NP-complete