Formula Satisfiability (SAT)

Given a boolean formula $\Phi$. Is there a way to set $\Phi$'s variable so it evaluates to true?

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

SAT is NP-hard (so SAT is NP-complete)
Reduce Circuit SAT to SAT.

Given a circuit \( K \).

Assign each wire a variable.

Write an equation describing each gate.

\[ a \land b = c \]

\( (a \land b = c) \land \text{the equations of the variable } z \text{ for the output} \)

\[ (y_1 = x_1 \land x_4) \land (y_2 = \overline{x_4}) \land (y_3 = x_3 \land y_2) \land (y_4 = y_1 \lor x_2) \land (y_5 = \overline{x_2}) \land (y_6 = \overline{x_5}) \land (y_7 = y_3 \lor y_5) \land (z = y_4 \land y_7 \land y_6) \land z \]
Lemma: \( K \) is sat. iff \( \Phi \) is.

\( \Rightarrow \): Use \( K \)'s good setting for \( x \) values. Set other variables by what their gates do.

All equations are sat, \( \exists z \) is set true, so \( \Phi \) is true.

\( \Leftarrow \): Use \( \Theta \)'s setting of \( x \) values.

\[
\text{CircuitSat}(K):
\text{transcribe } K \text{ into a boolean formula } \Phi
\text{return } \text{Sat}(\Phi) \quad (**MAGIC***)
\]
So we have poly time reduction from Circuit SAT to SAT.

So any problem \( A \in \text{NP} \) reduces to Circuit SAT which reduces to SAT.

\( \Rightarrow \) SAT is NP-hard.
A literal is a boolean variable \( a \) or its negation \( \overline{a} \).

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.

\[
\begin{align*}
\text{clause} & : (a \lor b \lor c \lor d) \land (b \lor \overline{c} \lor \overline{d}) \land (\overline{a} \lor c \lor d) \land (a \lor \overline{b})
\end{align*}
\]
A 3CNF formula is a CNF formula with exactly three literals per clause.

3SAT (3CNF SAT): Given a Boolean formula \( \phi \) in 3CNF, can we set variables to make \( \phi \) true?

3SAT \( \in \text{NP} \).

3SAT \( \in \text{NP-complete} \).
Reduce from Circuit SAT.

Given circuit $C$. 

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

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

3) Change equations into 2 or 3 clauses each:

\[
\begin{align*}
  a = b \land c & \quad \rightarrow \quad (a \lor \bar{b} \lor \bar{c}) \land (\bar{a} \lor b) \land (\bar{a} \lor c) \\
  a = b \lor c & \quad \rightarrow \quad (\bar{a} \lor b \lor c) \land (a \lor \bar{b}) \land (a \lor \bar{c}) \\
  a = \bar{b} & \quad \rightarrow \quad (a \lor b) \land (\bar{a} \lor \bar{b})
\end{align*}
\]
4) Replace clauses with 1 or 2 literals with 4 or 2 clauses with 3 literals. (Use a new x or y for each of these transforms)

\[
a \lor b \longrightarrow (a \lor b \lor x) \land (a \lor b \lor \bar{x})
\]

\[
a \longrightarrow (a \lor x \lor y) \land (a \lor \bar{x} \lor y) \land (a \lor x \lor \bar{y}) \land (a \lor \bar{x} \lor \bar{y})
\]

\[
(y_1 \lor \bar{x}_1 \lor x_4) \land (y_1 \lor x_1 \lor \bar{z}_1) \land (y_1 \lor x_1 \lor \bar{z}_2) \land (y_1 \lor x_4 \lor \bar{z}_2) \\
\land (y_2 \lor x_4 \lor \bar{z}_3) \land (y_2 \lor x_4 \lor \bar{z}_3) \land (y_2 \lor \bar{x}_4 \lor \bar{z}_4) \land (y_2 \lor \bar{x}_4 \lor \bar{z}_4)
\]

\[
(y_3 \lor \bar{x}_3 \lor \bar{y}_2) \land (y_3 \lor x_3 \lor \bar{z}_5) \land (y_3 \lor x_3 \lor \bar{z}_5) \land (y_3 \lor \bar{y}_2 \lor \bar{z}_6) \land (y_3 \lor \bar{y}_2 \lor \bar{z}_6) \\
\land (y_4 \lor \bar{y}_1 \lor x_2) \land (y_4 \lor \bar{x}_2 \lor \bar{z}_7) \land (y_4 \lor \bar{x}_2 \lor \bar{z}_7) \land (y_4 \lor \bar{y}_1 \lor \bar{z}_8) \land (y_4 \lor \bar{y}_1 \lor \bar{z}_8)
\]

\[
(y_5 \lor x_2 \lor \bar{z}_9) \land (y_5 \lor x_2 \lor \bar{z}_9) \land (y_5 \lor \bar{x}_2 \lor \bar{z}_{10}) \land (y_5 \lor \bar{x}_2 \lor \bar{z}_{10})
\]

\[
(y_6 \lor x_5 \lor \bar{z}_{11}) \land (y_6 \lor x_5 \lor \bar{z}_{11}) \land (y_6 \lor \bar{x}_5 \lor \bar{z}_{12}) \land (y_6 \lor \bar{x}_5 \lor \bar{z}_{12})
\]

\[
(y_7 \lor y_3 \lor y_5) \land (y_7 \lor \bar{y}_3 \lor \bar{z}_{13}) \land (y_7 \lor \bar{y}_3 \lor \bar{z}_{13}) \land (y_7 \lor y_5 \lor \bar{z}_{14}) \land (y_7 \lor y_5 \lor \bar{z}_{14})
\]

\[
(y_8 \lor \bar{y}_4 \lor \bar{y}_7) \land (y_8 \lor y_4 \lor z_{15}) \land (y_8 \lor y_4 \lor z_{15}) \land (y_8 \lor \bar{y}_7 \lor z_{16}) \land (y_8 \lor \bar{y}_7 \lor z_{16})
\]

\[
(y_9 \lor \bar{y}_8 \lor \bar{y}_6) \land (y_9 \lor y_8 \lor z_{17}) \land (y_9 \lor y_8 \lor z_{18}) \land (y_9 \lor \bar{y}_6 \lor \bar{z}_{18}) \land (y_9 \lor \bar{y}_6 \lor \bar{z}_{18})
\]

\[
(y_9 \lor z_{19} \lor z_{20}) \land (y_9 \lor \bar{z}_{19} \lor \bar{z}_{20}) \land (y_9 \lor z_{19} \lor \bar{z}_{20}) \land (y_9 \lor \bar{z}_{19} \lor \bar{z}_{20})
\]

**Reduction takes O(n) time.**
So it's NP-hard.
Given undirected graph 

\[
G = (V, E)
\]

An independent set \( S \subseteq V \)

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 \( \Phi \).

Build a graph \( G \).

For each clause in \( \Phi \), \( k \) vertices are created, one per literal of \( \Phi \).

For any pair of literals in a clause, they are connected with an “edge”.

Any two literals \( \text{a} \) and \( \text{a} \) are connected with a “negation” edge.
Claim: $\Phi$ is sat. if $G$ has an ind. set of size $k$.

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

$|S| = k$

$\subseteq$ Set variable so each ind set literal is true.
Decision Version: Ind Set

Given a graph $G$ and an integer $k$. Does $G$ have an independent set of size $k$?

$\text{Ind Set} \in \text{NP}$, $\in \text{NP-complete}$. 
A clique is another name for a complete graph.

Max Clique: Given G, what is the largest clique subgraph?

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

Min Vertex Cover: Find a min size vertex cover.
Claim: Both are NP-hard.

From MaxIndSet:

1) Given $G$. Take complement $\overline{G} = (V, \overline{E})$. $\overline{E} = \{uv : uv \notin E\}$
2) \( I \) is an ind. set of \( G = (V, E) \) iff \( V \setminus I \) is a vertex cover.

\[
\text{MaxIndependentSet} \leftarrow \text{MinVertexCover} \leftarrow n - k
\]

Decisions versions hard too

\( \Rightarrow \) \( \text{Clique} \) \& \( \text{Vertex Cover} \in \text{NP-complete} \)