3SAT (3CNF-SAT)

**Literal**: a boolean variable or its negation \((a, \bar{a})\) \((-a)\)

**Clause**: a disjunction of one or more literals \((b \lor \bar{c} \lor \bar{d})\)

**Conjunctive normal form (CNF)**: the conjunction (and) of clauses

\[\frac{\text{clause}}{(a \lor b \lor c \lor d) \land (b \lor \bar{c} \lor \bar{d}) \land (\bar{a} \lor c \lor d) \land (a \lor \bar{b})}\]

**3CNF**: CNF formula with exactly three literals per clause
3SAT: Given a 3CNF formula, is there an assignment to variable so formula evaluates to True?

Reduction from Circuit SAT:

Given a boolean circuit:

1) Change circuit so all AND & OR gates have two inputs:

2) \(( y_1 = x_1 \land x_2 ) \land ( y_2 = y_1 \land x_3 ) \land ( y_3 = y_2 \land x_4 )\)

Write an equation for each gate like in SAT.
3) Change each gate equation into a conjunction of clauses.

\[
\begin{align*}
  a = b \land c & \quad \rightarrow \quad (a \lor \bar{b} \lor \bar{c}) \land (\bar{a} \lor b) \land (\bar{a} \lor c) \\
  a = b \lor c & \quad \rightarrow \quad (\bar{a} \lor b \lor c) \land (a \lor \bar{b}) \land (a \lor \bar{c}) \\
  a = \bar{b} & \quad \rightarrow \quad (a \lor b) \land (\bar{a} \lor \bar{b})
\end{align*}
\]

(now the circuit’s formula is in CNF)

4) Change each two or one literal clause into a couple three literal clauses.

\[
\begin{align*}
  a \lor b & \quad \rightarrow \quad (a \lor b \lor x) \land (a \lor b \lor \bar{x}) \\
  a & \quad \rightarrow \quad (a \lor x \lor y) \land (a \lor \bar{x} \lor y) \land (a \lor x \lor \bar{y}) \land (a \lor \bar{x} \lor \bar{y})
\end{align*}
\]
(circuit to formula example)
gross but linear time!
So 3SAT is NP-hard.

3SAT ∈ NP. (Proof is how is set variables.)

⇒ 3SAT is NP-complete.
Given simple, unweighted graph \( G = (V,E) \).

Independent set: \( S \subseteq V \), No pair \( u,v \in S \) share an edge.

Maximum Independent Set (Max Ind Set)
Find largest ind. set in \( G \).

Max Ind Set is NP-hard!
(from 3SAT)
Given 3CNF formula $\Phi$.

$k \leftarrow$ number of clauses in $\Phi$

Make a graph $G$ with $3k$ vertices, one for each literal of $\Phi$ (one per literal-clause pair).

Any two literals in same clause share an edge. Call these "triangle" edges. (a $\lor$ $b$ $\lor$ c) $\Rightarrow$

Any pair of literals $a$ and $\bar{a}$ share a "negation" edge.
Claim: \( \Phi \) contains an independent set of size \( k \) iff \( \Phi \) is satisfiable.

Proof:

Suppose \( \Phi \) is satisfiable. Fix a satisfiable assignment. Each clause has at least one true literal. Take the corresponding vertex of one true literal in each clause to make a set \( S \). No triangle edge has both end vertices in \( S \). No negation edge connects a vertex in \( S \) to a non-vertex in \( S \).
\( S \) is an ind. set of size \( k \)

Suppose \( \exists \) ind. set \( S \) of size \( k \). Use any assignment that makes \( S \)'s literals True.

Triangle edges limit \( S \) to one literal per clause \( \Rightarrow \) it must use exactly one literal per clause.

Negation edges guarantee at most one of \( \bar{a} \) or \( a \) need be set to True.

We sat. every clause!
We reduced known NP-hard problem 3SAT to MaxIndSet in poly time \( \Rightarrow \) MaxIndSet is NP-hard.

This was the optimization version of the problem MaxIndSet. The decision version: Given \( G \) and an integer \( k \), is there an independent set of size \( k \)? Still hard, but now in \( NP \Rightarrow NP \)-complete.
The General Pattern:
To prove B is NP-hard, take known NP-hard problem A and reduce A to B.

1) Show a poly time algorithm to reduce a arbitrary instance of A to a special instance of B.
2) Prove that if a is "good" then b is "good".
3) Prove that if b is "good" then a is "good".
May help to imagine each stage as an algorithm!
The "proof" are often called certificates.

1) The algorithm to turn a into 6.
2) Alg to turn a certificate for a into one for 6.
3) Alg to turn a certificate for 6 into a certificate for a.
Still have $G = (V, E)$.

clique: a complete subgraph

Max Clique: Find the largest clique in $G$.

vertex cover: a subset of vertices touching every edge at least once

Min Vertex Cover: Find a smallest vertex cover.

Both are NP-hard!
Max Clique from Max Ind Set.

Given $G$ for which we want an ind. set.

edge-complement: $\bar{G}$ of $G$: opposite set of edges $e$ in $\bar{G}$ if $e$ not in $G$.

$S \subseteq V$ is independent in $G$ if $S$ is a clique in $\bar{G}$.

\[
\begin{array}{ccc}
\text{MaxIndependentSet} & \xrightarrow{\text{complement in } O(V^2) \text{ time}} & \text{MaxClique} \\
G \text{ graph} & \xrightarrow{\bar{G} \text{ graph}} & k \text{ size of largest clique in } \bar{G} \\
& & k \text{ size of largest independent set in } \bar{G}
\end{array}
\]
Min Vertex Cover from Max Ind Set:

Obs: $S \subseteq V$ is independent if and only if $V \setminus S$ is a vertex cover.

Decision versions (given $G$ and $k$, is there a $S \subseteq V$ of size $k$?) are NP-hard also. And in NP.

$\Rightarrow$ NP-complete.