P — solvable in polynomial time

NP — yes instances can be confirmed in poly time

A problem X is NP-hard

Formally: If X can be solved in poly time then P=NP
Practical: X cannot be solved in poly time

Cook-Levin: CircuitSAT is NP-hard

To prove X is NP-hard
Reduce a known NP-hard problem to X in poly time

CircuitSAT, SAT, MaxIndSet, 3SAT

3SAT input: \((a \lor b \lor c) \land (\overline{a} \lor \overline{b} \lor d) \land (b \lor \overline{c} \lor \overline{e}) \land \ldots\)

Max Independent Set
Input: Graph \(G=(V,E)\)

Question: Size of largest independent set of vertices?
\[
(\overline{a} \lor b \lor c) \land (\overline{b} \lor \overline{c} \lor d) \land (\overline{a} \lor c \lor d) \land (a \lor \overline{b} \lor \overline{d})
\]

**MaxIndSet ≤ k**

If MIS = k
then formula is solvable
If formula is satisfiable
then MIS = k

\[V = \text{literals} \quad E = \text{same clause} + \text{contradictions} \quad k = \# \text{clauses} \quad \text{poly time} \checkmark\]

\[a = F \quad b = T \quad c = F \quad d = F\]

\[\Rightarrow a = T \quad b = T \quad c = T \quad d = \text{whatever}\]
NP-hardness proof: Reduce \( Z \) to \( X \):

1. Describe a polytime algo that transforms an arbitrary input to \( Z \) into a special input to \( X \).

2. Prove: if input to \( Z \) is good, then input to \( X \) is also good.

3. Prove: if input to \( X \) is good, then input to \( Z \) is also good.
Given $G = (V, E)$ build new graph $G' = (V, E')$

$$E' = \{ uv \mid uv \notin E \}$$

Reduction from 3SAT:

1. Truth gadget
   - Truth gadget for a variable $v$.
2. Variable gadget
   - Variable gadget for a variable $v$. 

$$v \lor \overline{v} \lor c$$
(a \lor b \lor c) \land (b \lor \bar{c} \lor \bar{d}) \land (\bar{a} \lor c \lor d) \land (a \lor \bar{b} \lor \bar{d})

One of these must be \text{T}.