Take

G' = complement of G (edge (x,y) in G' iff no edge (x,y) in G)

$k' = k$.

① Inverting the presence of every edge takes $O(|V|^2)$ time.

② $\Rightarrow$: If G' has an independent set of size k, those same vertices form a clique in G, since edges not present in G' are in G.

$\Leftarrow$: If G has a clique of size k, those same vertices form an indep. set in G'.

$\square$

Teaching Evals!

L19

Example: Vertex Cover

Input: Graph G and integer k.

Question: Does G contain a vertex cover of size k?

subset $V' \subseteq V$ such that every edge has at least 1 endpoint in V'

Claim: Clique $\leq_P$ Vertex Cover

Proof: We need a graph G' and integer k', s.t.:

① We can build G' in polynomial time

② G' has a vertex cover of size k'

$\Longleftrightarrow$

G has a clique of size k

Take G' = complement of G, $k' = |V| - k$.

① Holds, as before.

② $\Rightarrow$ Suppose G' has a vertex cover V' of size $k' = |V| - k$. Let u, v be two vertices in $V - V'$. Since V' is a vertex cover, (u,v) is not an edge of G'. Thus, (u,v) is an edge of G. Therefore $V - V'$ is a clique in G of size $|V| - (|V| - k) = k$. ✓

$\Leftarrow$ Suppose G has a clique V' of size k. Consider an edge $(u,v) \in G'$. Since G' is the complement of G, u and v cannot both be in V'. Thus, $V - V'$ is a vertex cover ~~of~~ for G' of size $|V| - k = k'$. ✓

NP-Completeness

[Teaching Evals!]

We're looking for the "most difficult" problems in NP.

= Problems at least as hard as <u>all</u> others.

= Problems Y such that $X \leq_p Y$ for all $X \in NP$.

Such a problem is called <u>NP-hard</u>.

An NP-hard problem <u>in NP</u> is called <u>NP-complete</u>.

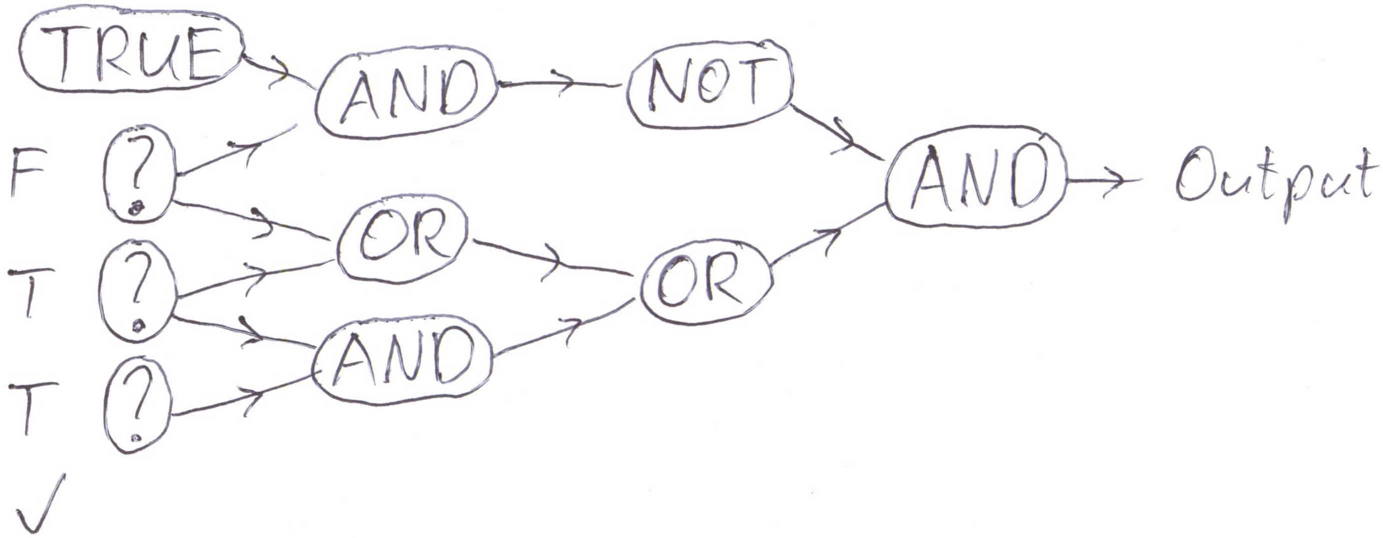
Circuit-SAT (satisfiability)

<u>Input</u>: Boolean circuit

- DAG where vertices are gates
- AND / OR / NOT gates
- Literal TRUE / FALSE gates
- <u>Unknown</u> input gates
- One output gate

<u>Question</u>: Is there an assignment of truth values to the input gates that makes the output gate TRUE?

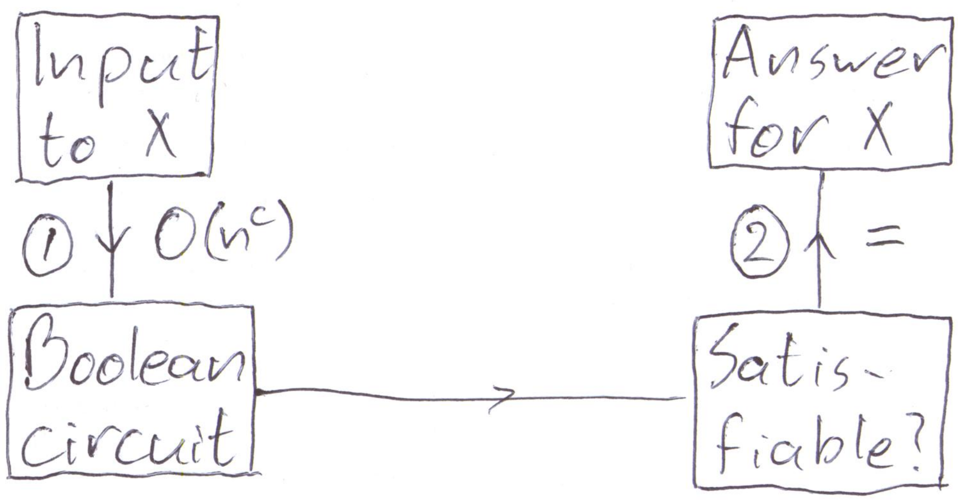
Example:



Claim: Circuit-SAT is NP-complete.

Proof:

* Circuit-SAT is in NP (Exercise!)
* For every problem $X \in NP$, $X \leq_p$ Circuit-SAT.



What do we know about X?

$X \in NP$: there is a verification algorithm V:

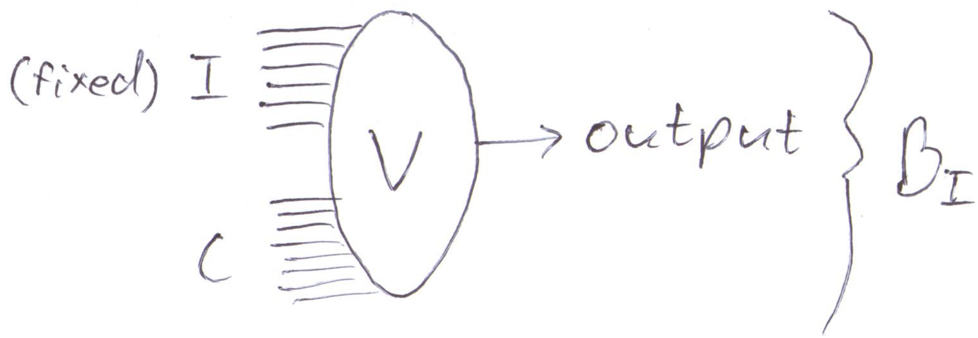
~~V runs in $O(n^k)$ time~~

- $X(I) = YES \iff \exists$ certificate C: $V(I, C) = YES$

— V runs in $O(n^c)$ ~~time~~ steps

① Each step does $O(n^{c'})$ bit-operations.

$\Rightarrow$ We can implement V as a boolean circuit of size $O(n^{c+c'})$.



② $X(I) = \text{YES} \Leftrightarrow \exists$ certificate $C : V(I,C) = \text{YES}$

$\Leftrightarrow \exists$ assignment $C : B_I(C) = \text{TRUE}$

$\Leftrightarrow B_I$ is satisfiable $\square$