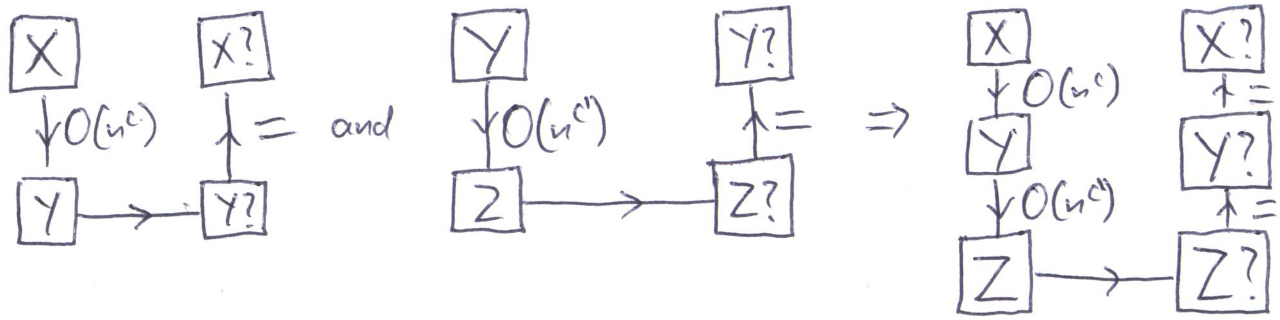
Claim: If $X \leq_p Y$ and $Y \leq_p Z$, then $X \leq_p Z$.

Proof:



□

Claim: If Y is NP-hard, and $Y \leq_p Z$, then Z is also NP-hard.

Proof: To show that Z is NP-hard, we need to prove that $X \leq_p Z$, for each $X \in NP$. Consider a problem $X \in NP$. Since Y is NP-hard, we know that $X \leq_p Y$. We also know that $Y \leq_p Z$ (given), so by transitivity, $X \leq_p Z$. Thus, Z is NP-hard. □

⇒ To show that a problem Z is NP-complete:

1. Show that $Z \in NP$
2. Look for a problem Y that is "similar" to Z and that is <u>known to be</u> NP-complete.
3. Show that $Y \leq_p Z$.

Example: 3-SAT

Input: Boolean formula φ with variables $x_1, x_2, \ldots, x_n$ of the form

$$\varphi = C_1 \wedge C_2 \wedge C_3 \wedge \ldots \wedge C_m,$$

where each clause C_i is of the form

$$C_i = (l_1^i \vee l_2^i \vee l_3^i)$$

and each $\underbrace{l_j^i}_{\text{literal}}$ is either a variable x_k or the negation $\neg x_k$ of a variable.

Question: Does there exist an assignment of truth values to the variables such that φ is true?

Example:

$$\varphi = (x_1 \vee \neg x_1 \vee \neg x_2) \wedge (x_2 \vee x_3 \vee x_4) \wedge (x_1 \vee x_3 \vee \neg x_4)$$

x_1	x_2	x_3	x_4
F	T	F	F

makes φ true.

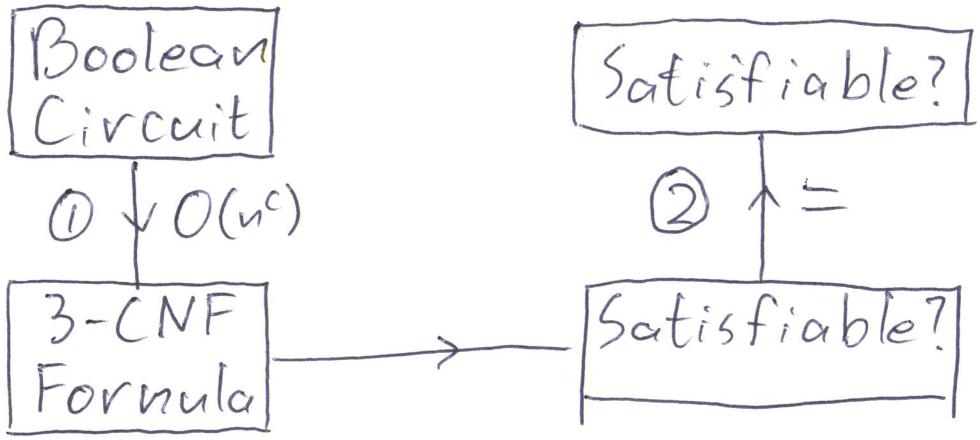
Claim: 3-SAT is NP-complete.

Proof:

* 3-SAT is in NP: Certificate = truth assignment that makes φ true.
  Verification:
  - check that every clause has at least one true literal ✓

* 3-SAT is NP-hard: by reduction ~~to~~ from Circuit-SAT.

We need to show Circuit-SAT $\leq_p$ 3-SAT.



Given a boolean circuit B:

- Introduce a variable for the output of each gate.
- Use clauses to describe the effect of each gate.
- Connect all clauses by $\wedge$.

Gates:

TRUE $\xrightarrow{x}$ $\longrightarrow$ clause: x, same as $(x \vee x \vee x)$

FALSE $\xrightarrow{x}$ $\longrightarrow$ clause: $\neg x$, same as $(\neg x \vee \neg x \vee \neg x)$

? $\xrightarrow{x}$ $\longrightarrow$ no clause.

$\xrightarrow{y}$ NOT $\xrightarrow{x}$ $\longrightarrow$ clause: $(x \Leftrightarrow \neg y)$, same as

$(\neg x \vee \neg y \vee \neg y)$
$\wedge (x \vee y \vee y)$

$\rightarrow$ clause: $(x \Leftrightarrow (y \vee z))$, same as

$$(\neg x \vee y \vee z)$$
$$\wedge (x \vee \neg y \vee \neg y)$$
$$\wedge (x \vee \neg z \vee \neg z)$$

$\rightarrow$ clause: $(x \Leftrightarrow (y \wedge z))$, same as

$$(\neg x \vee y \vee y)$$
$$\wedge (\neg x \vee z \vee z)$$
$$\wedge (x \vee \neg y \vee \neg z)$$

Output gate $\xrightarrow{x}$ $\rightarrow$ clause: x, same as

$$(x \vee x \vee x)$$

Let φ be the conjunction of all these clauses. By construction, φ is satisfiable if and only if B is satisfiable. (2)

(1) Size of φ:

- #variables = #gates in B
- each clause has 3 literals
- #clauses $\leq 3 \times$ #gates in B

Thus, φ ~~has polynomial~~ can be computed in time polynomial in the size of B. $\square$

List of NP-complete problems

- Circuit-SAT (L 19, p. 69)
- 3-SAT (L 20, p. 73)
- Independent Set (L 18, p. 66)
- Clique (L 18, p. 66)
- Vertex Cover (L 19, p. 67)
- Hamiltonian Cycle (L 17, p. 61)
- TSP

  Input: Complete weighted graph G, number k.

  Question: Does G contain a Hamiltonian cycle with total weight $\leq k$?

- Knapsack (L 12, p. 42)
- Subset sum

  Input: Set $S = \{x_1, x_2, \dots, x_n\}$ of numbers. Target number t.

  Question: Is there a subset $S' \subseteq S$ such that $\sum_{x \in S'} x = t$?