How to show that A is NP-complete

1. Show $A \in NP$.

2. Select a known NP-complete problem B.

3. Do a **polynomial-time reduction**: Give a polynomial time algorithm which will transform an instance x' of B to an instance of x of A such that x' has a "yes" answer (for B) if and only if x has a "yes" answer (for A).

3-SAT

A boolean formula is in 3-SAT form if it is in CNF form and each clause has exactly three literals per clause.

Theorem 1. *3-SAT is NP-Complete.*

Proof.

3-SAT $\in NP$.

We will give a polynomial-time reduction from SAT to 3-SAT.

Consider an instance F of SAT. We convert it into an instance F' of 3-SAT such that F is satisfiable iff F' is satisfiable.

In each clause of F that has 1 or 2 literals, we replicate one of the literals till the total number is three.

If a clause has more than 3 literals, say, $(a_1 \vee a_2 \vee \cdots \vee a_l)$, then we replace it with the following $l-2$ clauses:

$(a_1 \vee a_2 \vee z_1) \wedge (z_1^c \vee a_3 \vee z_2) \wedge (z_2^c \vee a_4 \vee z_3) \wedge \cdots \wedge (z_{l-3}^c \vee a_{l-1} \vee a_l)$.

The size of F' is only polynomially larger than F.

If a clause in F is satisfiable then all the corresponding clauses in F' can be satisfied and viceversa. $\square$

2SAT

Theorem 2. *2SAT is in P.*

Proof: Let ϕ be an instance of 2SAT.

Define a directed graph $G(\phi)$ as:

(1) the vertices of G are variables of ϕ and their negations;

(2) there is an edge between (α, β) iff there is a clause $(\neg\alpha \vee \beta)$ in ϕ.

Claim: ϕ is satisfiable iff there is a variable x such that there are paths from x to $\neg x$ and from $\neg x$ to x in $G(\phi)$.

Proof of Claim

Suppose such paths exist. Assume that ϕ is satisfiable by truth assignment T.

Let $T(x) = true$. Since there is a path from x to $\neg x$, there must be an edge (α, β) in this path such that $T(\alpha) = true$ and $T(\beta) = false$.

$\implies$ that the clause $(\neg\alpha \vee \beta)$ is not satisfied by T. Contradiction.

Suppose there is no variable with such paths in $G(\phi)$.

Construct the following truth assignment by repeating the following:

Pick a node α whose truth value is not yet defined and such that there is no path from α to $\neg\alpha$.

Assign all nodes reachable from α, $true$; assign $false$ to negations of these nodes.

This satisfies ϕ.

We can check the condition of the Claim in polynomial time.