CMSC 28100

Introduction to Complexity Theory

Autumn 2025
Instructor: William Hoza

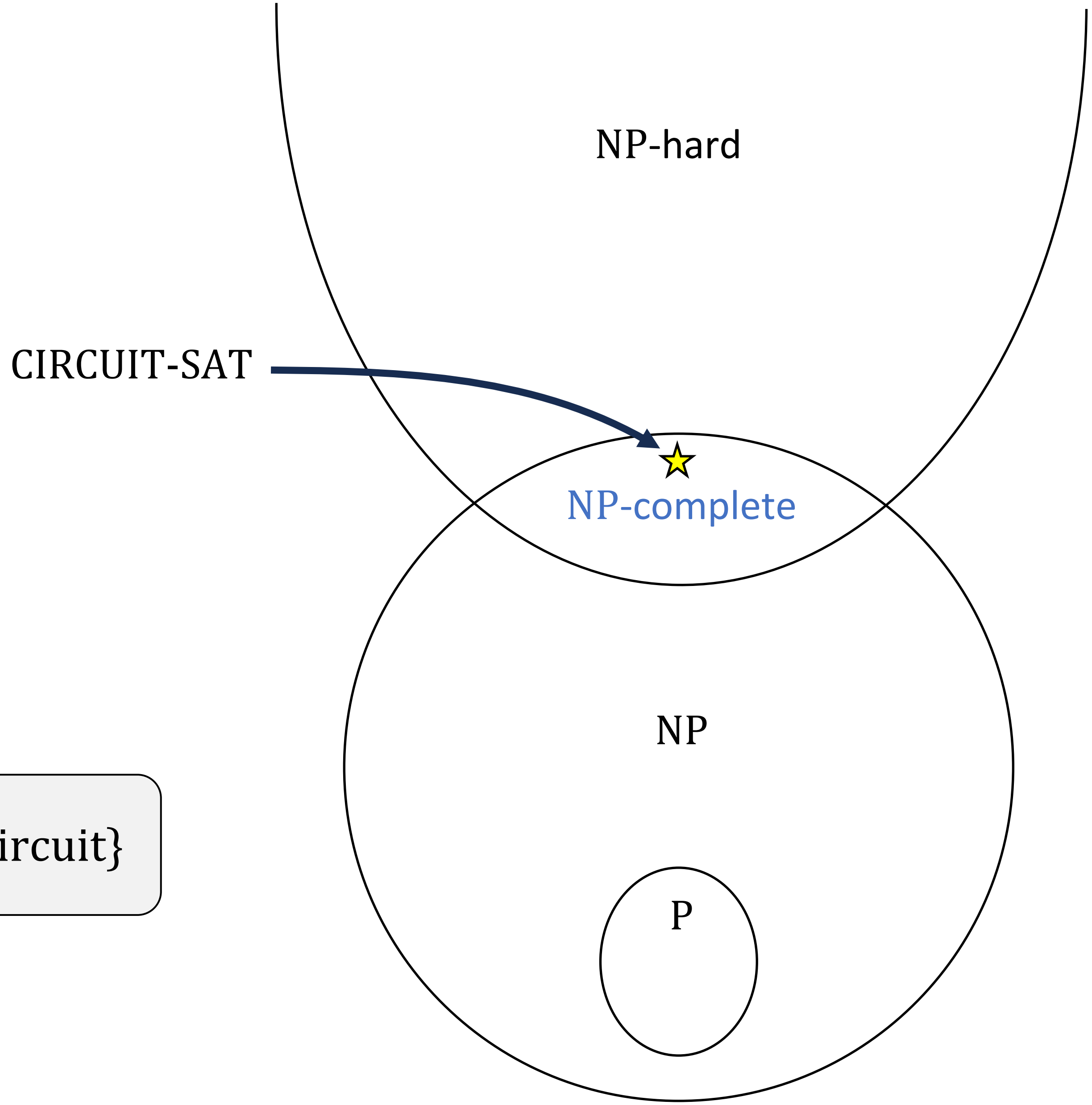
NP-hardness

- Let $Y \subseteq \{0,1\}^*$
- **Definition:** Y is NP-hard if, for every $L \in \mathrm{NP}$, we have $L \leq_{\mathrm{P}} Y$
- Interpretation:
  - Y is at least as hard as any language in NP
  - Every problem in NP is basically a special case of Y

NP-completeness

- Let $Y \subseteq \{0, 1\}^*$
- **Definition:** Y is NP-complete if Y is NP-hard and $Y \in \mathrm{NP}$
- The NP-complete languages are the hardest languages in NP
- If Y is NP-complete, then the language Y can be said to "capture" / "express" the entire complexity class NP

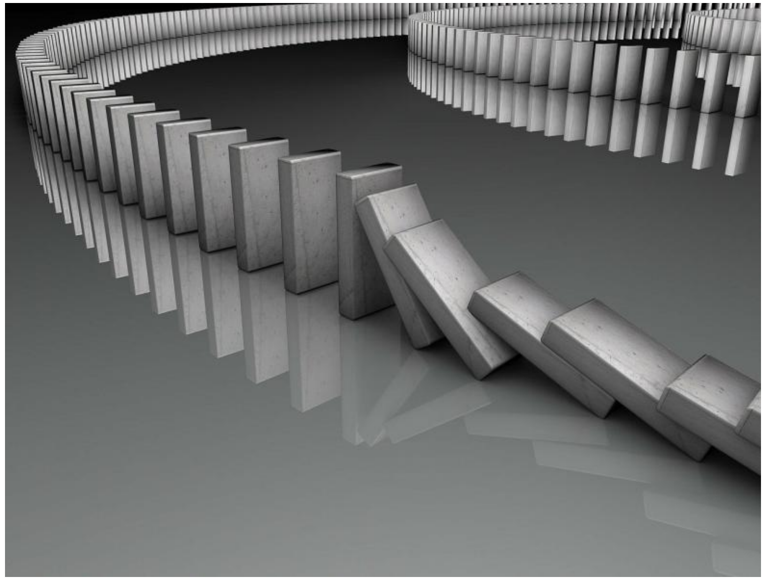
NP-completeness



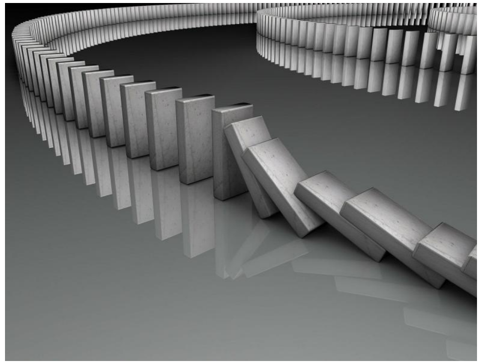
CIRCUIT-SAT $= \{\langle C \rangle : C$ is a satisfiable circuit$\}$

What else is NP-complete?

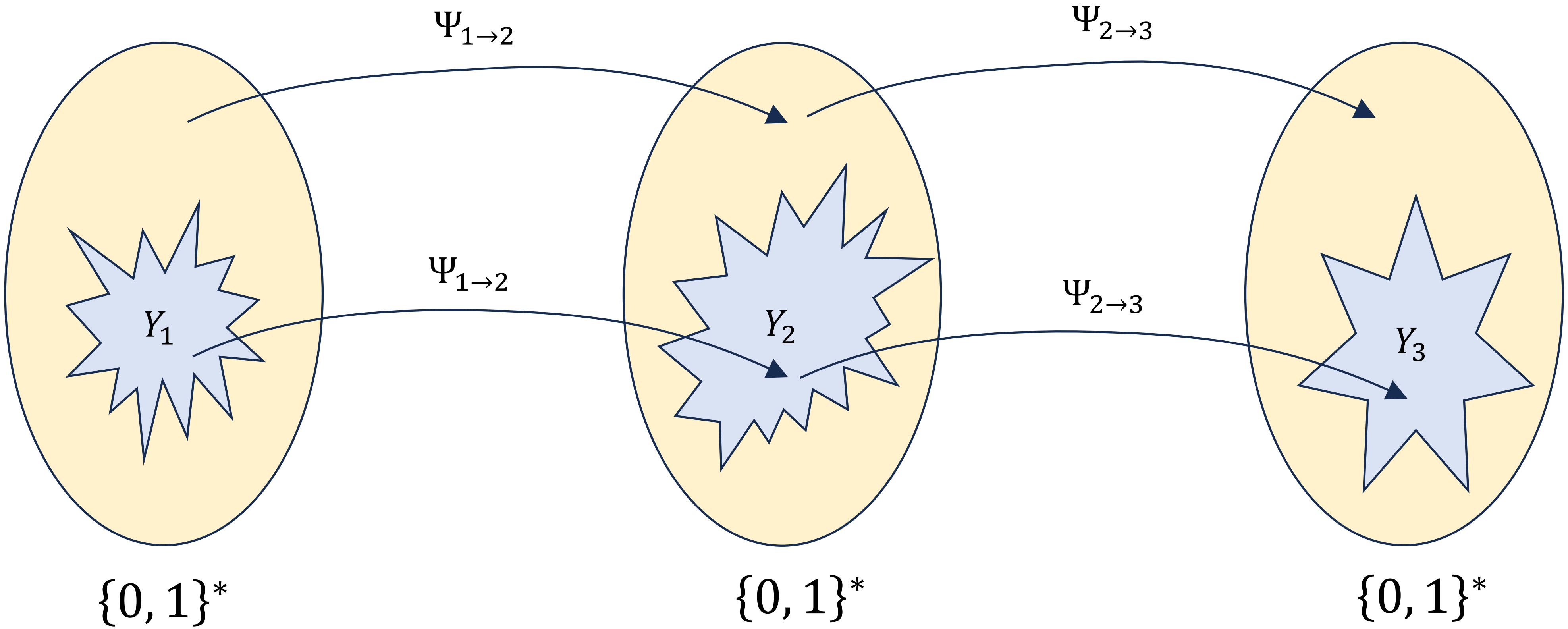
- We showed that CIRCUIT-SAT is NP-complete
- This will help us to prove that other problems, such as CLIQUE, are also NP-complete
- Idea: Chain reductions together



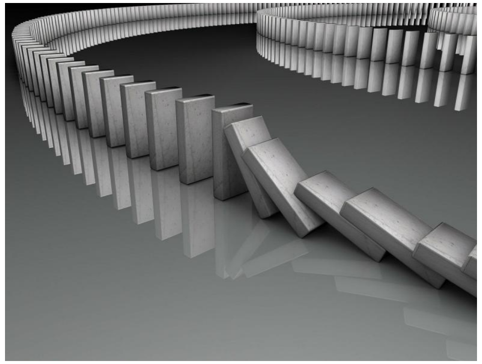
Chaining reductions together



- If $Y_1 \leq_{\text{P}} Y_2 \leq_{\text{P}} Y_3$, then $Y_1 \leq_{\text{P}} Y_3$

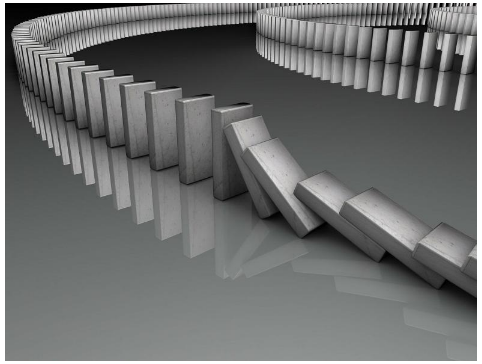


Chaining reductions together



- Let $Y_{\mathrm{OLD}}, Y_{\mathrm{NEW}} \subseteq \{0,1\}^*$
- **Claim:** If Y_{OLD} is NP-hard and $Y_{\mathrm{OLD}} \leq_{\mathrm{P}} Y_{\mathrm{NEW}}$, then Y_{NEW} is NP-hard
- **Proof:** Let $L \in \mathrm{NP}$
- Then $L \leq_{\mathrm{P}} Y_{\mathrm{OLD}} \leq_{\mathrm{P}} Y_{\mathrm{NEW}}$
- Therefore, $L \leq_{\mathrm{P}} Y_{\mathrm{NEW}}$

Roadmap



- We will define a language called "3-SAT"
- We will prove $\text{CIRCUIT-SAT} \leq_{\text{P}} \text{3-SAT} \leq_{\text{P}} \text{CLIQUE}$
- This will show that CLIQUE is NP-hard

k-CNF formulas

- Recall: A CNF formula is an “AND of ORs of literals”
- **Definition:** A k-CNF formula is a CNF formula in which every clause has at most k literals
- Example of a 3-CNF formula with two clauses:

$$\phi = (x_1 \vee \bar{x}_2 \vee \bar{x}_6) \wedge (x_5 \vee x_1 \vee x_2)$$

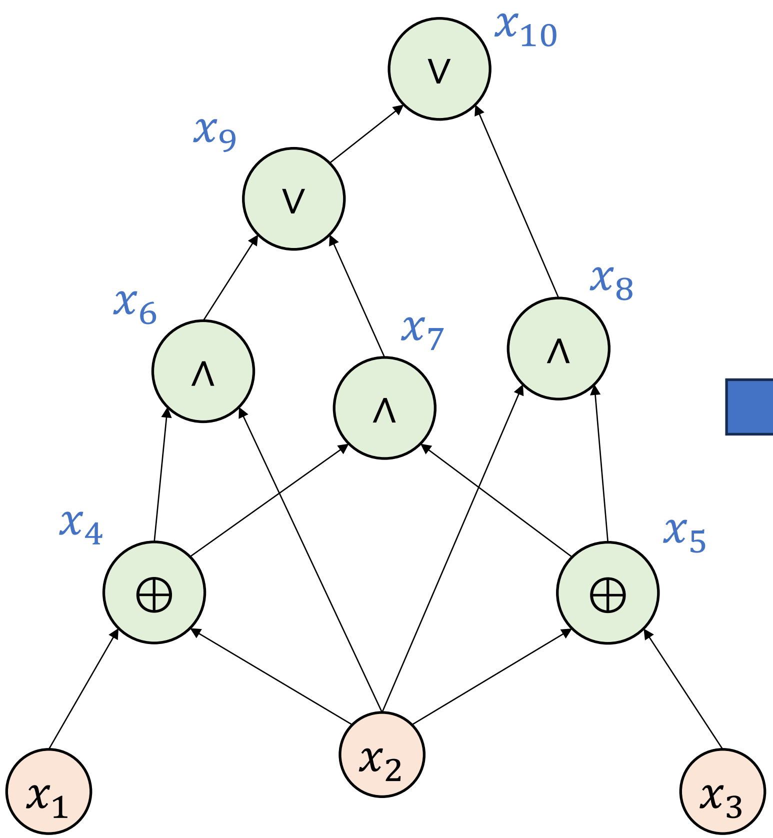
The Cook-Levin Theorem

- Define $k\text{-SAT} = \{\langle\phi\rangle : \phi \text{ is a satisfiable } k\text{-CNF formula}\}$

> **The Cook-Levin Theorem:** 3-SAT is NP-complete

- **Proof step 1:** $3\text{-SAT} \in \text{NP}$. (What is the certificate?)
- **Proof step 2:** We need to show that 3-SAT is NP-hard
  - Reduction from CIRCUIT-SAT

Step 1: Circuit ⇒ Program



Circuit (3 input variables)

- $x_4 \leftarrow x_1 \oplus x_2$
- $x_5 \leftarrow x_2 \oplus x_3$
- $x_6 \leftarrow x_4 \wedge x_2$
- $x_7 \leftarrow x_4 \wedge x_5$
- $x_8 \leftarrow x_2 \wedge x_5$
- $x_9 \leftarrow x_6 \vee x_7$
- $x_{10} \leftarrow x_9 \vee x_8$
- Return x_{10}

Program (3 input variables)

Step 2: Program ⇒ Formula

- $x_4 \leftarrow x_1 \oplus x_2$
- $x_5 \leftarrow x_2 \oplus x_3$
- $x_6 \leftarrow x_4 \wedge x_2$
- $x_7 \leftarrow x_4 \wedge x_5$
- $x_8 \leftarrow x_2 \wedge x_5$
- $x_9 \leftarrow x_6 \vee x_7$
- $x_{10} \leftarrow x_9 \vee x_8$
- Return x_{10}

Program (3 input variables)

$$
\begin{aligned}
&\left(x_4 == (x_1 \oplus x_2)\right)\\
\wedge &\left(x_5 == (x_2 \oplus x_3)\right)\\
\wedge &\left(x_6 == (x_4 \wedge x_2)\right)\\
\wedge &\left(x_7 == (x_4 \wedge x_5)\right)\\
\wedge &\left(x_8 == (x_2 \wedge x_5)\right)\\
\wedge &\left(x_9 == (x_6 \vee x_7)\right)\\
\wedge &\left(x_{10} == (x_9 \vee x_8)\right)\\
\wedge &\,(x_{10})
\end{aligned}
$$

Formula (10 input variables)

Let C be the initial circuit and let ϕ be the final 3-CNF formula.
Which of the following is <u>false</u>?

A: C is satisfiable if and only if ϕ is satisfiable

B: $|\langle\phi\rangle| \leq \mathrm{poly}(|\langle C\rangle|)$

C: The number of clauses in ϕ is $\Theta(\text{size of } C)$

D: C and ϕ compute the same Boolean function

Respond at PollEv.com/whoza or text “whoza” to 22333

Step 3

$(x_4 == (x$

$\wedge (x_5 == (x$

$\wedge (x_6 == (x_4 \wedge x_2))$

$\wedge (x_7 == (x_4 \wedge x_5))$

$\wedge (x_8 == (x_2 \wedge x_5))$

$\wedge (x_9 == (x_6 \vee x_7))$

$\wedge (x_{10} == (x_9 \vee x_8))$

$\wedge (x_{10})$

Formula (10 input variables)

$\vee x_2) \wedge (\bar{x}_4 \vee \bar{x}_1 \vee \bar{x}_2)$

$\vee x_3) \wedge (\bar{x}_5 \vee \bar{x}_2 \vee \bar{x}_3)$

$\wedge (\bar{x}_6 \vee x_4) \wedge (\bar{x}_6 \vee x_2) \wedge (\bar{x}_4 \vee \bar{x}_2 \vee x_6)$

$\wedge (\bar{x}_7 \vee x_4) \wedge (\bar{x}_7 \vee x_5) \wedge (\bar{x}_4 \vee \bar{x}_5 \vee x_7)$

$\wedge (\bar{x}_8 \vee x_2) \wedge (\bar{x}_8 \vee x_5) \wedge (\bar{x}_2 \vee \bar{x}_5 \vee x_8)$

$\wedge (\bar{x}_9 \vee x_6 \vee x_7) \wedge (x_9 \vee \bar{x}_6) \wedge (x_9 \vee \bar{x}_7)$

$\wedge (\bar{x}_{10} \vee x_9 \vee x_8) \wedge (x_{10} \vee \bar{x}_9) \wedge (x_{10} \vee \bar{x}_8)$

$\wedge (x_{10})$

Every Boolean function has a CNF representation!

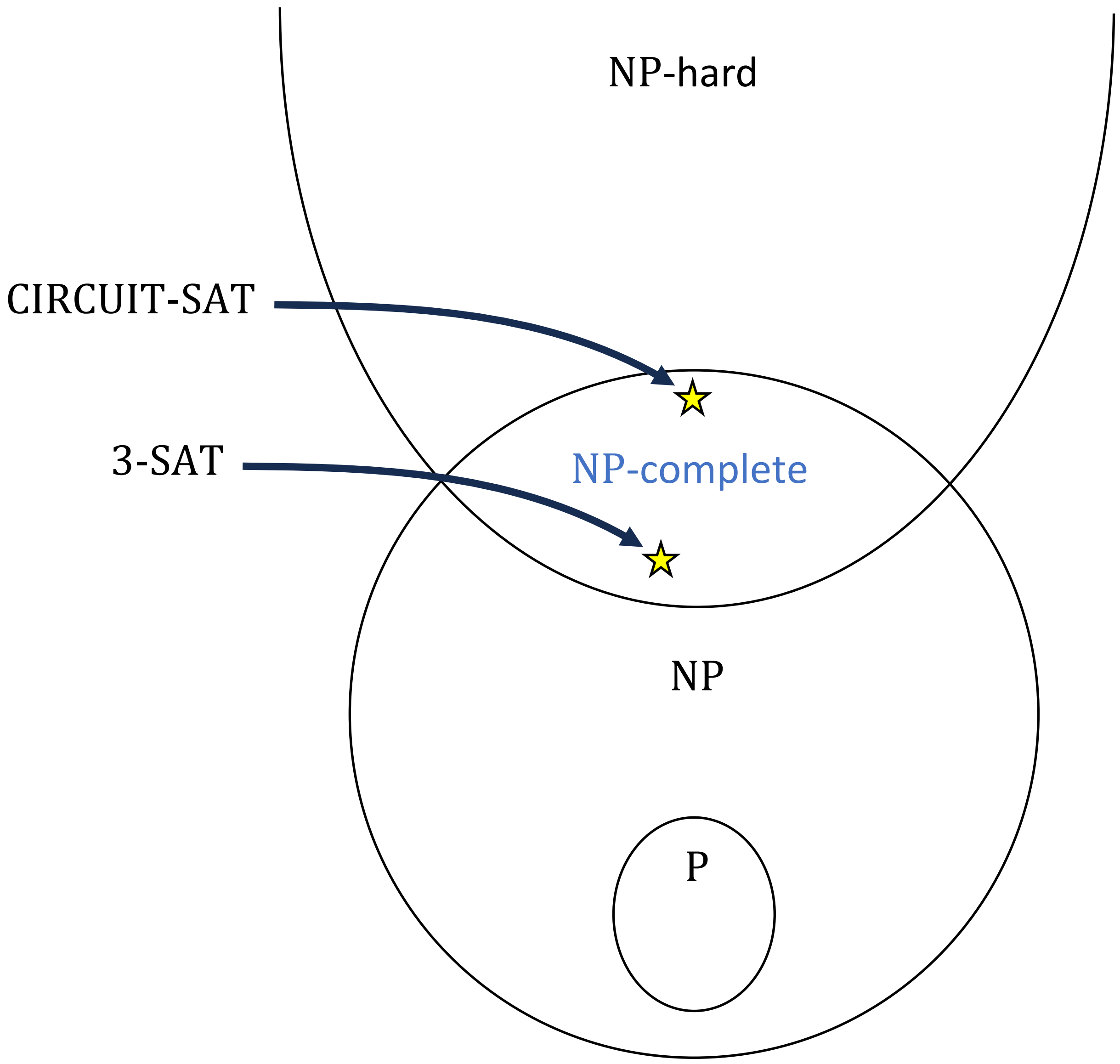
3-CNF Formula (10 input variables)

Reduction correctness

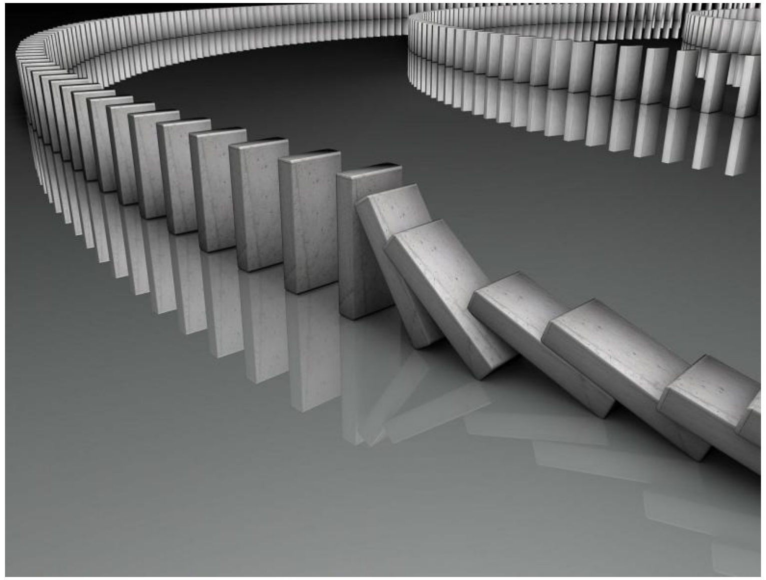
- Let the gates of C be $g_1, \ldots, g_m$ (topological order)
- **Claim:** C is satisfiable if and only if ϕ is satisfiable
- **Proof:** ($\Rightarrow$) Suppose $C(x_1, \ldots, x_n) = 1$
- Let $x_{n+i} = g_i(x_1, \ldots, x_n)$
- Then $\phi(x_1, \ldots, x_{n+m}) = 1$

Reduction correctness

- Let the gates of C be $g_1, \ldots, g_m$ (topological order)
- **Claim:** C is satisfiable if and only if ϕ is satisfiable
- **Proof:** $(\Leftarrow)$ Suppose $\phi(x_1, \ldots, x_{n+m}) = 1$
- Then $x_{n+i} = g_i(x_1, \ldots, x_n)$ for every i by induction
- Furthermore, $x_{n+m} = 1$
- Therefore, $C(x_1, \ldots, x_n) = 1$


NP-hard
CIRCUIT-SAT
3-SAT
NP-complete
NP
P

Chaining reductions together



- $\mathrm{3\text{-}SAT}$ is the starting point for many NP-hardness proofs
- We are finally ready to prove that CLIQUE is NP-complete

CLIQUE is NP-complete

- Recall $\mathrm{CLIQUE} = \{\langle G, k\rangle : G \text{ contains a } k\text{-clique}\}$

Theorem: CLIQUE is NP-complete

- **Proof:** We showed $\mathrm{CLIQUE} \in \mathrm{NP}$ in a previous class

- To prove that CLIQUE is NP-hard, we will do a reduction from 3-SAT

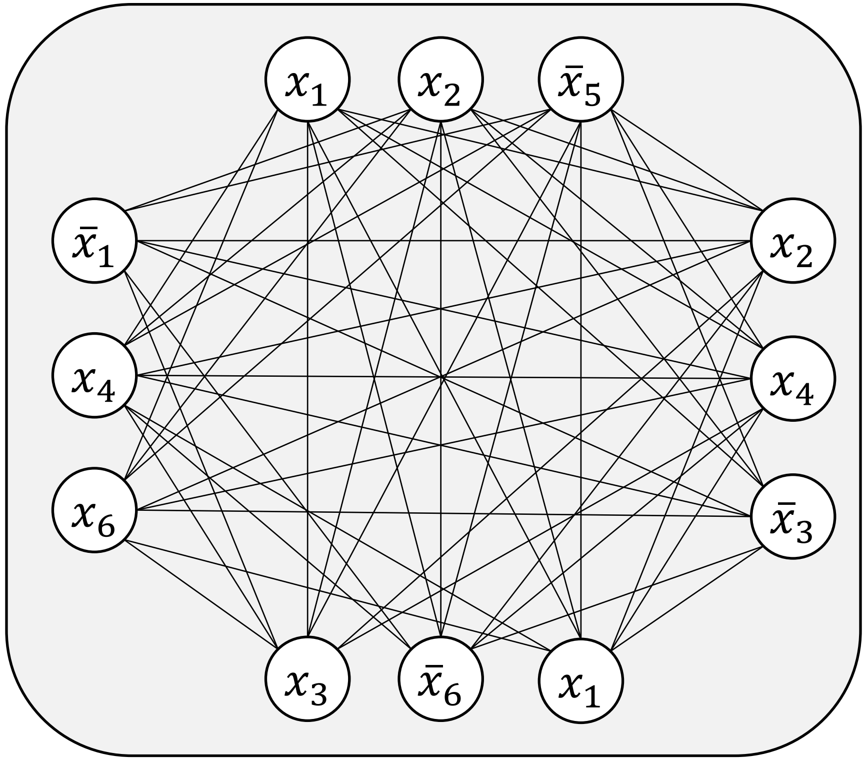
Proof that $3\text{-SAT} \leq_{\mathrm{P}} \mathrm{CLIQUE}$

- Let ϕ be a 3-CNF formula (an instance of 3-SAT)
- Reduction: $\Psi(\langle\phi\rangle) = \langle G, k\rangle$
  - k is the number of clauses in ϕ
  - G is a graph on $\leq 3k$ vertices defined as follows

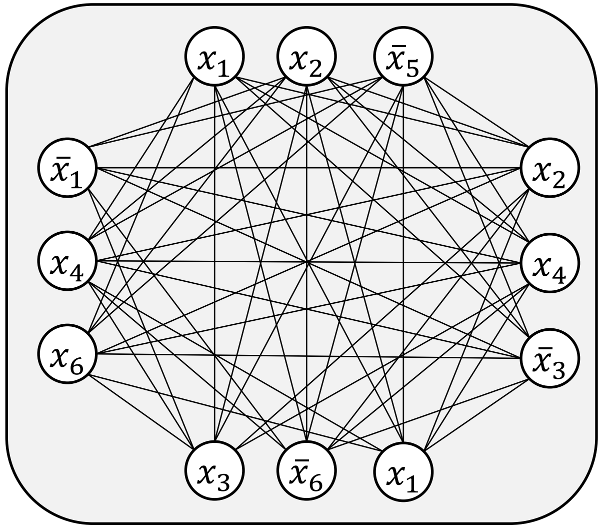
Reduction from 3-SAT to CLIQUE

- For each clause $(\ell_1 \vee \ell_2 \vee \ell_3)$, create a "group" of three vertices labeled ℓ_1, ℓ_2, ℓ_3
  - (If the clause only has one or two literals, then only use one or two vertices)
- Put an edge $\{u, v\}$ if u and v are in different groups and u and v do not have contradictory labels (x_i and $\bar{x}_i$)

- E.g., $\phi = (x_1 \vee x_2 \vee \bar{x}_5) \wedge (\bar{x}_1 \vee x_4 \vee x_6)$
  $\wedge (x_2 \vee x_4 \vee \bar{x}_3) \wedge (x_3 \vee \bar{x}_6 \vee x_1)$

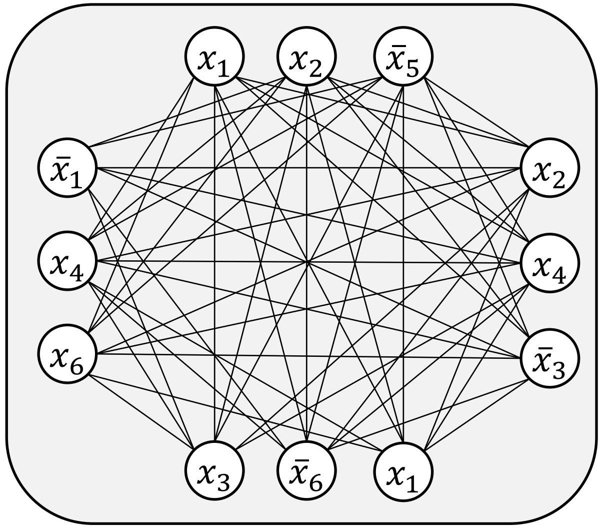


YES maps to YES



- Suppose there exists x such that $\phi(x) = 1$
- In each clause, at least one literal is satisfied by x
- Therefore, in each group, at least one vertex is "satisfied by x," i.e., it is labeled by a literal that is satisfied by x
- Let S be a set consisting of one satisfied vertex from each group
- Then S is a k-clique (vertices in S cannot have contradictory labels)

NO maps to NO



- Suppose G has a k-clique S
- Let x be an assignment that satisfies each vertex in S
  - This exists because no two vertices in S have contradictory labels
- S cannot contain two vertices from a single group, and $|S| = k$, so S must contain one vertex from each group
- Therefore, x satisfies at least one literal in each clause, so $\phi(x) = 1$