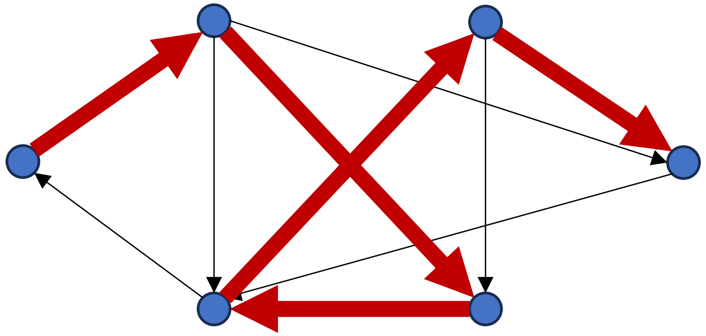
CMSC 28100

Introduction to Complexity Theory

Spring 2025
Instructor: William Hoza

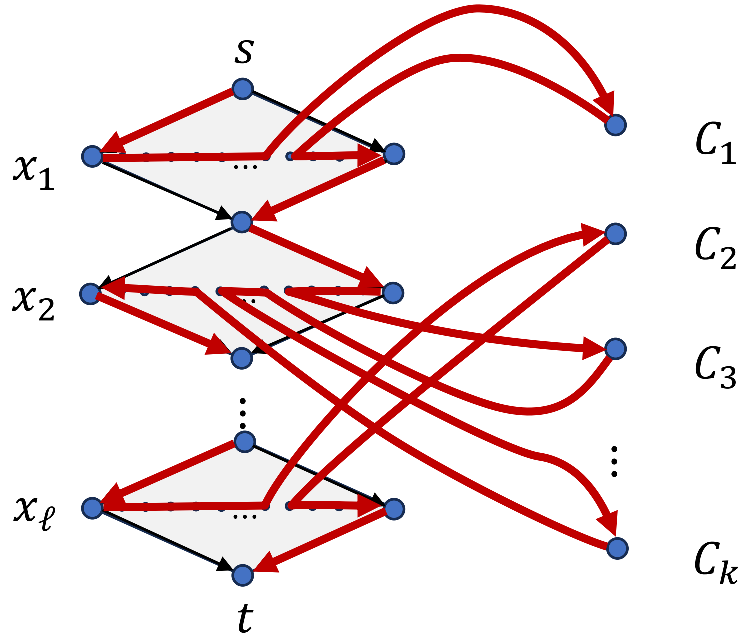
Hamiltonian paths



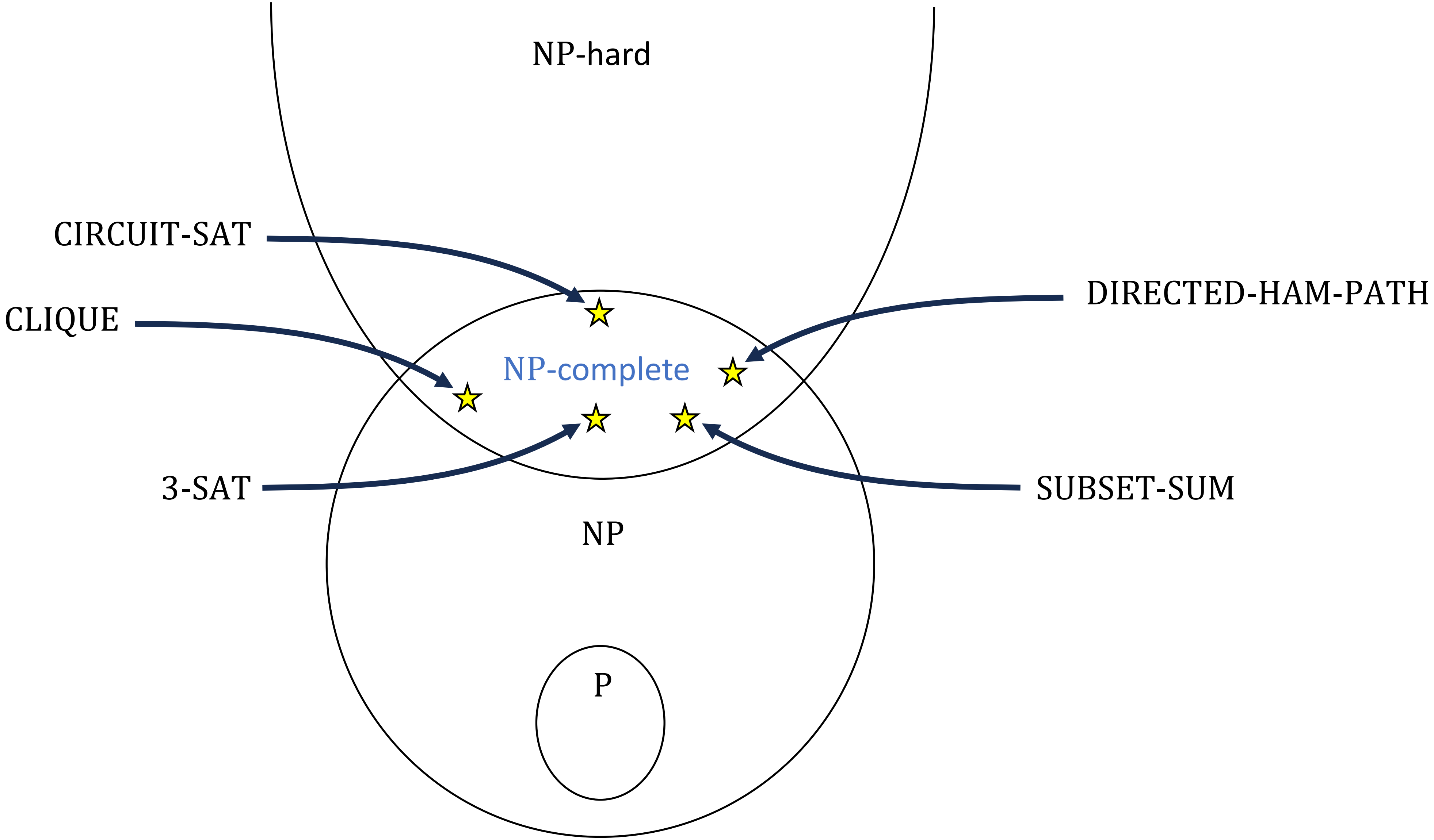
- Let G be a directed graph
- **Definition:** A Hamiltonian path is a directed path that visits every vertex exactly once

DIRECTED-HAM-PATH is NP-complete

- Let DIRECTED-HAM-PATH $= \{\langle G, s, t\rangle :$ G is a digraph, s and t are vertices, and there exists a Hamiltonian path from s to $t\}$



Theorem: DIRECTED-HAM-PATH is NP-complete


NP-hard
CIRCUIT-SAT
DIRECTED-HAM-PATH
CLIQUE
NP-complete
3-SAT
SUBSET-SUM
NP
P

Undirected Hamiltonian path

- Let G be an undirected graph
- A Hamiltonian path in G is a path that visits every vertex exactly once
- Let $\text{UNDIRECTED-HAM-PATH} = \{\langle G, s, t \rangle : G \text{ is an undirected graph, } s \text{ and } t \text{ are vertices, and there exists a Hamiltonian path from } s \text{ to } t\}$

Theorem: UNDIRECTED-HAM-PATH is NP-complete

UNDIRECTED-HAM-PATH is NP-complete

- First, note that UNDIRECTED-HAM-PATH ∈ NP (why?)
- To prove that UNDIRECTED-HAM-PATH is NP-hard, we will do a reduction from DIRECTED-HAM-PATH

Attempted reduction: "Replace each directed edge (u, v) with an undirected edge $\{u, v\}$." Does this work?

A: Yes

B: It doesn't work, because it doesn't run in polynomial time

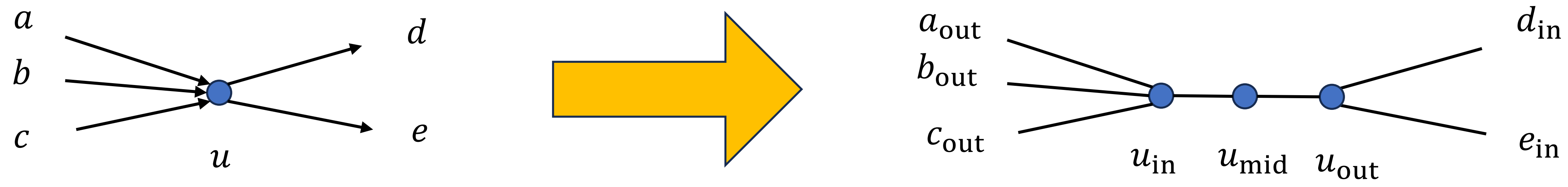
C: It doesn't work, because it doesn't always map YES to YES

D: It doesn't work, because it doesn't always map NO to NO

Respond at PollEv.com/whoza or text "whoza" to 22333

From directed to undirected

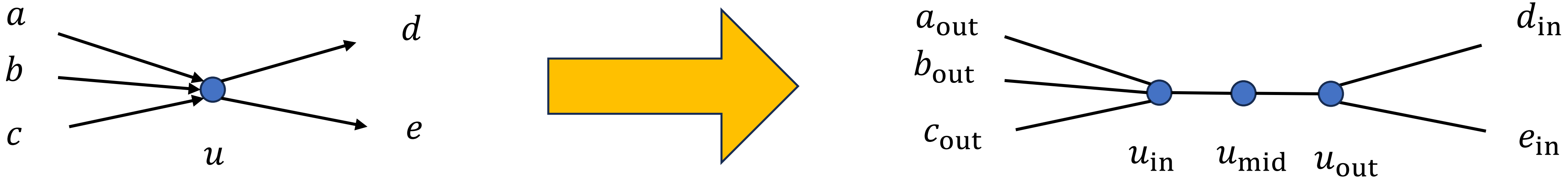
- Reduction: Given $\langle G, s, t\rangle$, produce $\langle G', s_{\text{in}}, t_{\text{out}}\rangle$, constructed as follows:

1. Delete all edges going into s or coming out of t
2. 

- YES maps to YES: Suppose $s \to u_1 \to u_2 \to \cdots \to u_k \to t$ is a Hamiltonian path

- New Hamiltonian path in G':

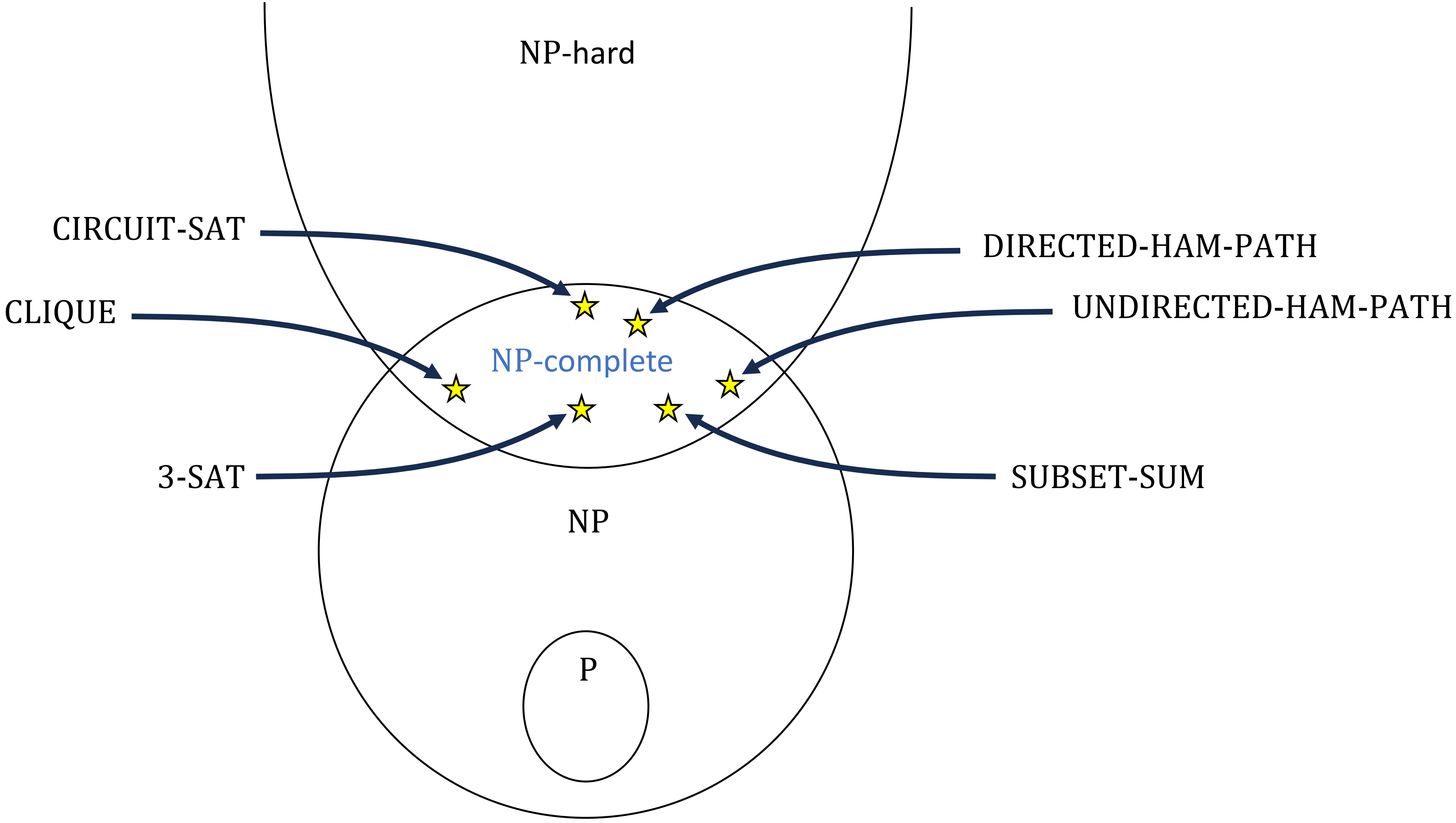
$$s_{\text{in}} \sim s_{\text{mid}} \sim s_{\text{out}} \sim (u_1)_{\text{in}} \sim (u_1)_{\text{mid}} \sim (u_1)_{\text{out}} \sim (u_2)_{\text{in}} \sim \cdots \sim (u_k)_{\text{out}} \sim t_{\text{in}} \sim t_{\text{mid}} \sim t_{\text{out}}$$



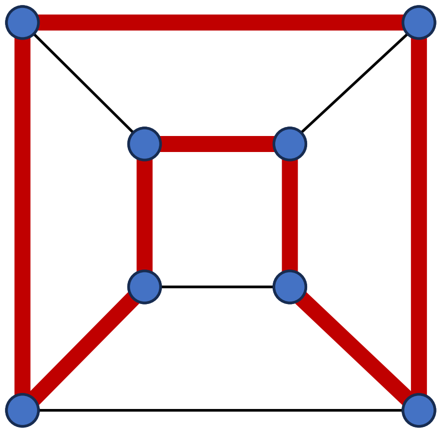
- NO maps to NO: Suppose G' has a Hamiltonian path from s_{in} to t_{out}
- We deleted edges going into s, so the path begins $s_{\text{in}} \sim s_{\text{mid}} \sim s_{\text{out}}$
- For every u, path must eventually use edges $u_{\text{in}} \sim u_{\text{mid}} \sim u_{\text{out}}$
- Therefore, the path has the form

$$s_{\text{in}} \sim s_{\text{mid}} \sim s_{\text{out}} \sim (u_1)_{\text{in}} \sim (u_1)_{\text{mid}} \sim (u_1)_{\text{out}} \sim (u_2)_{\text{in}} \sim \cdots \sim (u_k)_{\text{out}} \sim t_{\text{in}} \sim t_{\text{mid}} \sim t_{\text{out}}$$

- Hamiltonian path in G: $s \to u_1 \to u_2 \to \cdots \to u_k \to t$


NP-hard
CIRCUIT-SAT
DIRECTED-HAM-PATH
CLIQUE
UNDIRECTED-HAM-PATH
NP-complete
3-SAT
SUBSET-SUM
NP
P

Hamiltonian cycles



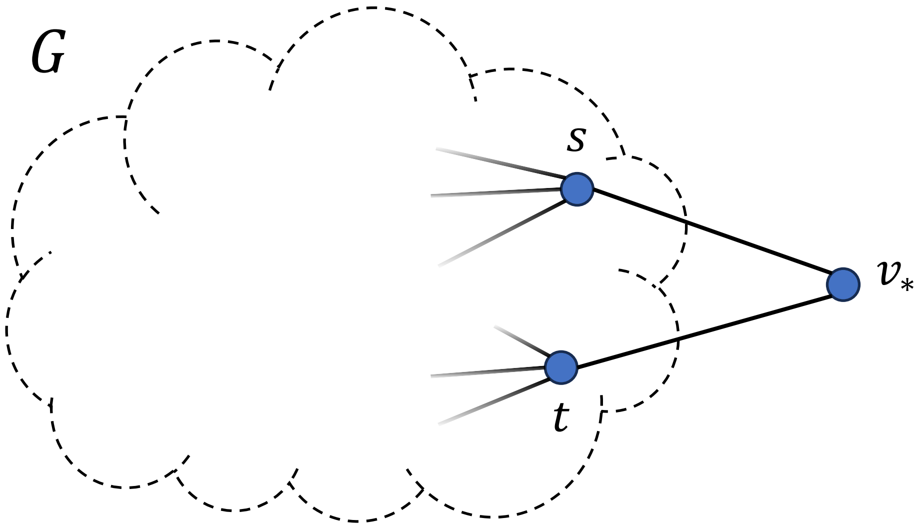
- Let G be an undirected graph
- A Hamiltonian cycle is a cycle that visits every vertex exactly once
- Let $\text{UNDIRECTED-HAM-CYCLE} = \{\langle G \rangle : G \text{ is an undirected graph with at least one Hamiltonian cycle}\}$

Theorem: UNDIRECTED-HAM-CYCLE is NP-complete

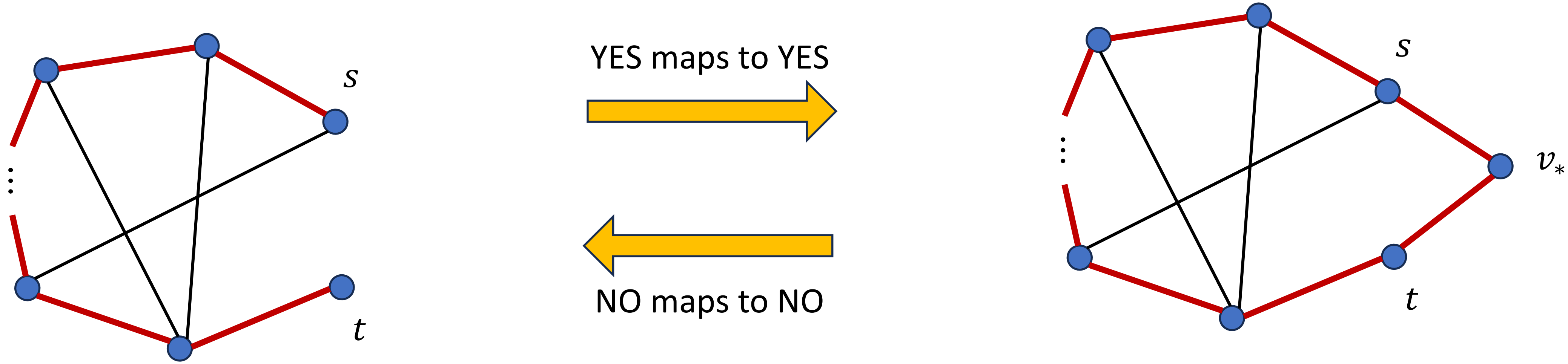
$\mathrm{UNDIRECTED\text{-}HAM\text{-}CYCLE}$ is NP-complete

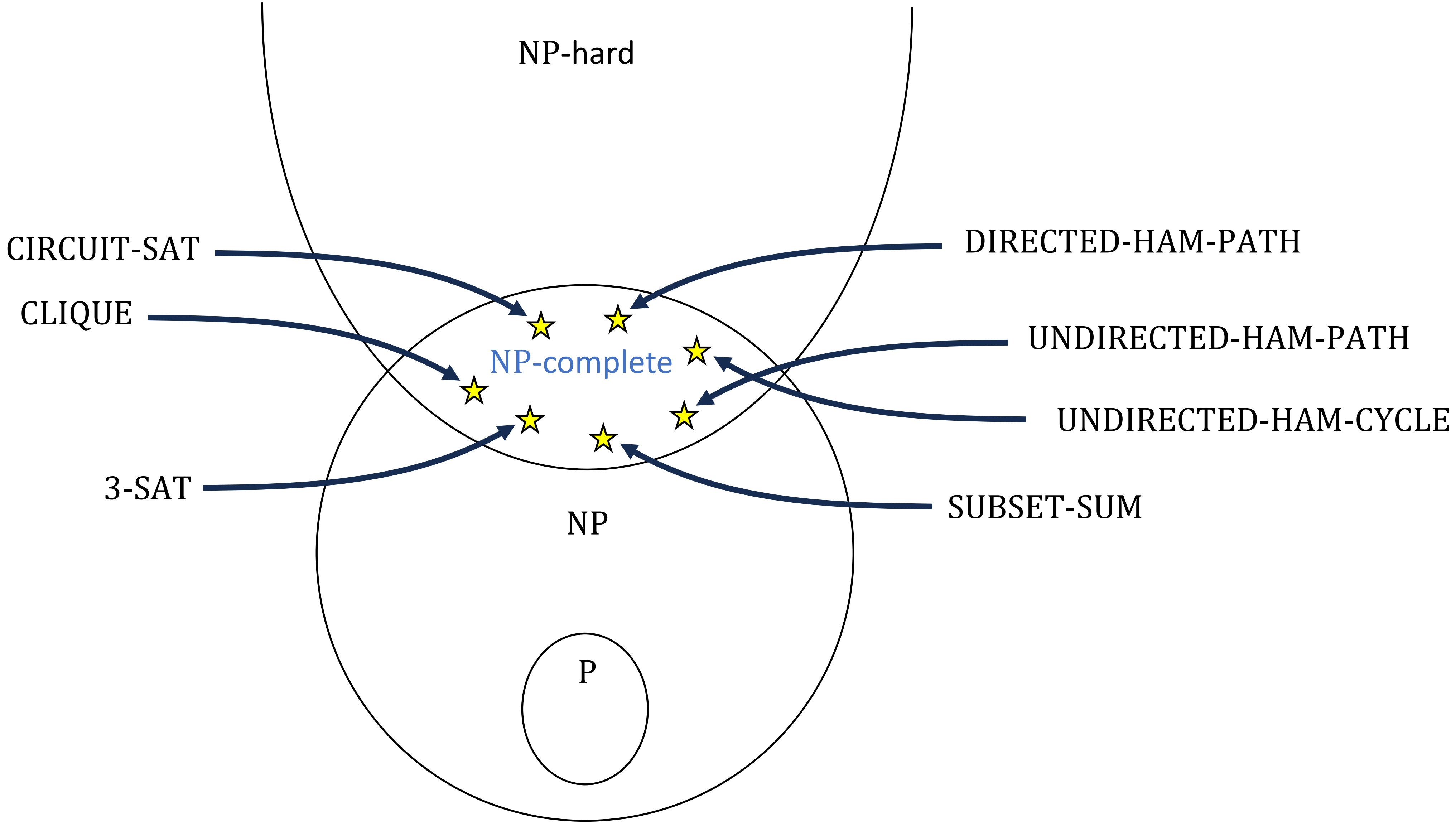
- **Proof:** First note that $\mathrm{UNDIRECTED\text{-}HAM\text{-}CYCLE} \in \mathrm{NP}$ (why?)
- To prove NP-hardness, we do a reduction from $\mathrm{UNDIRECTED\text{-}HAM\text{-}PATH}$

From paths to cycles



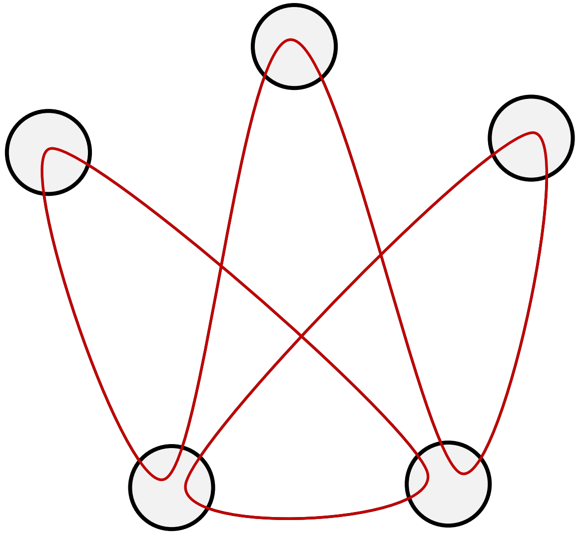
- Reduction: Given $\langle G, s, t\rangle$, add one new vertex v_* and two new edges $\{s, v_*\}$ and $\{t, v_*\}$
- Poly-time computable ✔



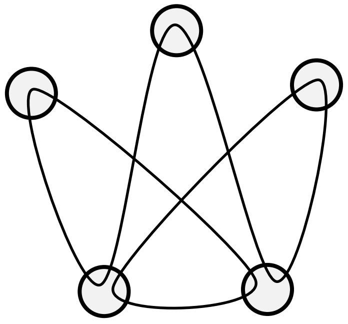

NP-hard
CIRCUIT-SAT
CLIQUE
NP-complete
3-SAT
DIRECTED-HAM-PATH
UNDIRECTED-HAM-PATH
UNDIRECTED-HAM-CYCLE
SUBSET-SUM
NP
P

Comparison: Eulerian cycles

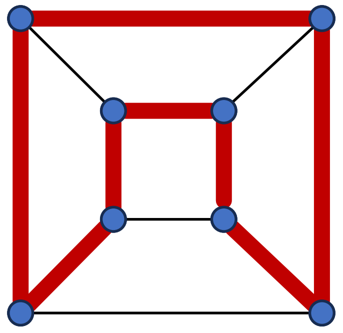
- Let G be an undirected graph
- An Eulerian cycle is a cycle that traverses every edge exactly once (possibly visiting some vertices multiple times)



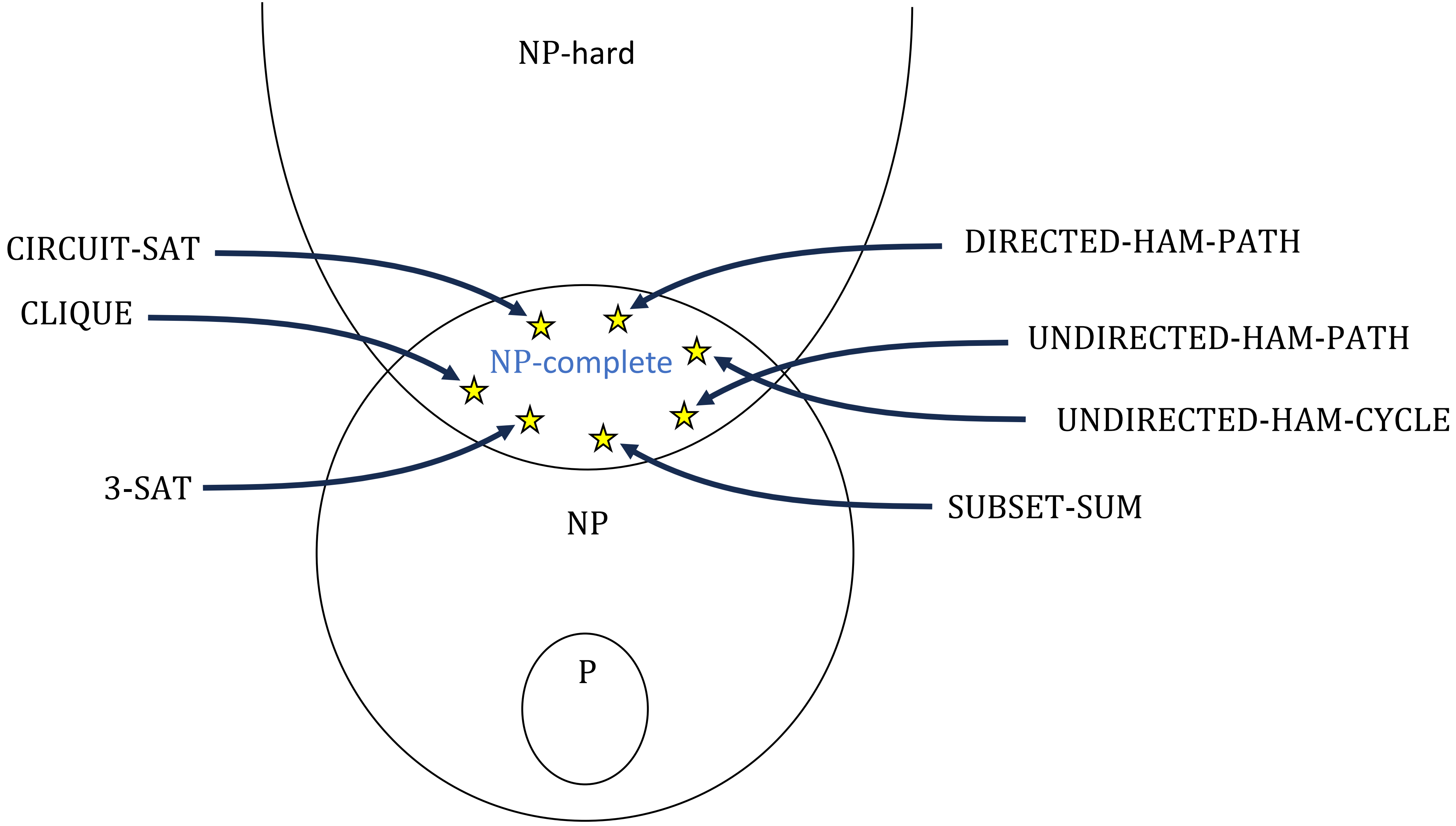
Eulerian cycles vs. Hamiltonian cycles



- Which graphs have Eulerian cycles?
  - Let G be a simple, undirected, connected graph
  - **Euler's Theorem:** G has an Eulerian cycle if and only if every vertex has even degree
  - (Proof omitted)



- Which graphs have Hamiltonian cycles?
  - There is probably no "good" answer to this question!


NP-hard
CIRCUIT-SAT
CLIQUE
NP-complete
3-SAT
DIRECTED-HAM-PATH
UNDIRECTED-HAM-PATH
UNDIRECTED-HAM-CYCLE
SUBSET-SUM
NP
P

NP-completeness is everywhere

- There are thousands of known NP-complete problems!
- These problems come from many different areas of study
  - Logic, graph theory, number theory, scheduling, optimization, economics, physics, chemistry, biology, …

Proving that Y_{NEW} is NP-complete ("cheat sheet")

1. Prove that $Y_{\mathrm{NEW}} \in \mathrm{NP}$
   - What is the certificate? How can you verify a purported certificate in polynomial time?
2. Prove that Y_{NEW} is NP-hard
   - Which NP-complete language Y_{OLD} are you reducing from?
   - What is the reduction? "Given w, construct w'." How is w' defined? Polynomial time?
   - YES maps to YES: Assume there is a certificate x showing $w \in Y_{\mathrm{OLD}}$. In terms of x, describe a certificate y showing that $w' \in Y_{\mathrm{NEW}}$.
   - NO maps to NO: (Contrapositive) Assume there is a certificate y showing $w' \in Y_{\mathrm{NEW}}$. In terms of y, describe a certificate x showing that $w \in Y_{\mathrm{OLD}}$.

NP-complete languages stand or fall together

- If you design a poly-time algorithm for one NP-complete language, then $\mathrm{P} = \mathrm{NP}$, so all NP-complete languages can be decided in poly time!

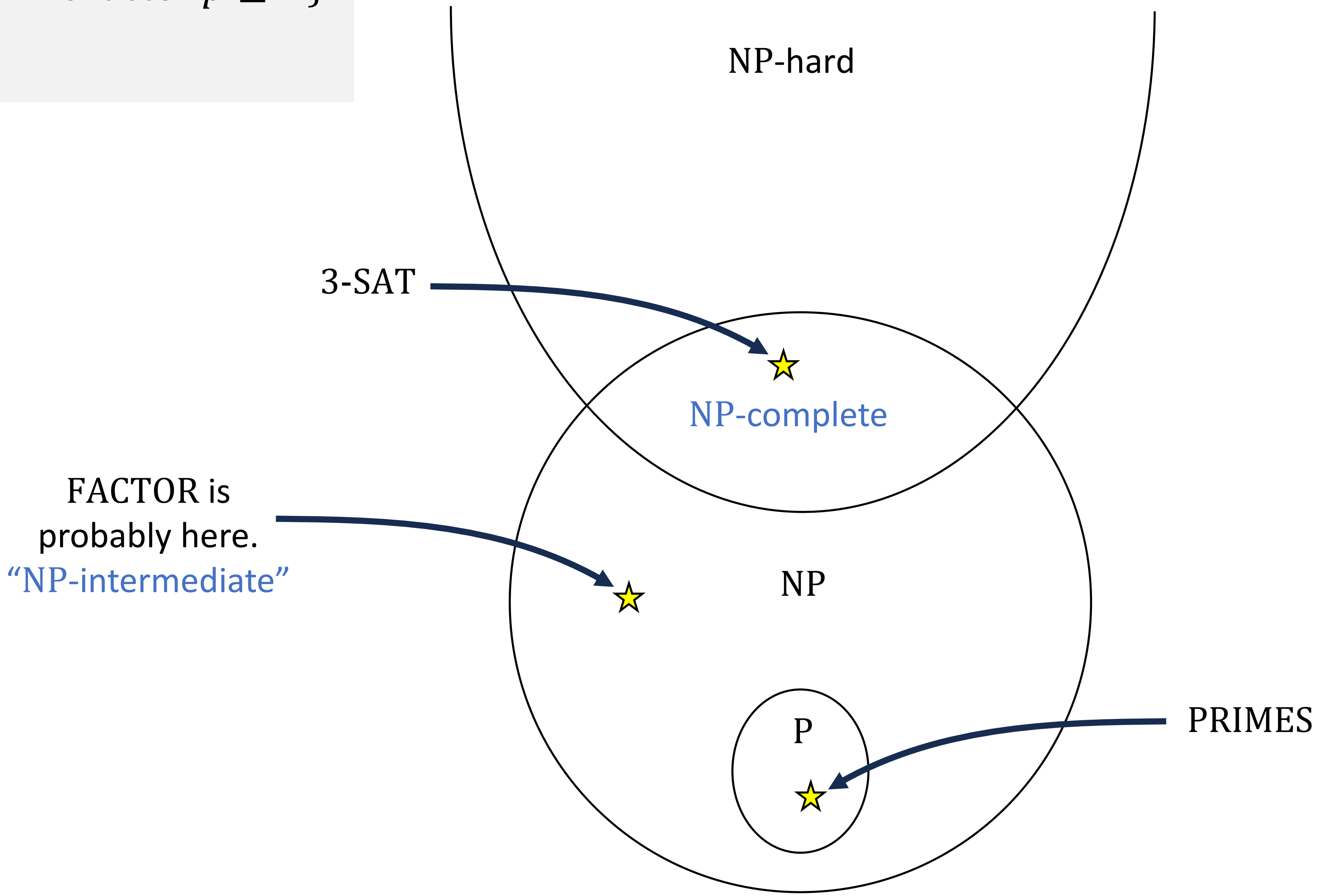
- If you prove that one NP-complete language cannot be decided in poly time, then $\mathrm{P} \neq \mathrm{NP}$, so no NP-complete language can be decided in poly time!

Complexity of factoring integers

- Recall $\text{FACTOR} = \{\langle K, R\rangle : K \text{ has a prime factor } p \leq R\}$
- In most cases, if a language Y is in NP, then we can either prove $Y \in \text{P}$ or we can prove that Y is NP-complete
- FACTOR is one of the rare exceptions to this rule
- **Conjecture:** FACTOR is neither in P nor NP-complete!

$$\text{FACTOR} = \{\langle K, R\rangle : K \text{ has a prime factor } p \leq R\}$$

$$\text{PRIMES} = \{\langle K\rangle : K \text{ is prime}\}$$



Complexity of factoring integers

- Why do experts expect that FACTOR is not NP-complete?
- Key: The complexity class coNP
- Informal definition: coNP is like NP, except that we swap the roles of “yes” and “no”

The complexity class coNP

- Let $Y \subseteq \{0,1\}^*$
- **Definition:** $Y \in \mathrm{coNP}$ if there exists a randomized polynomial-time Turing machine M such that for every $w \in \{0,1\}^*$:
  - If $w \in Y$, then $\Pr[M \text{ accepts } w] = 1$
  - If $w \notin Y$, then $\Pr[M \text{ accepts } w] \neq 1$

The complexity class coNP

- Let $Y \subseteq \{0,1\}^*$ and let $\bar{Y} = \{0,1\}^* \setminus Y$
- **Fact:** $Y \in \mathrm{NP}$ if and only if $\bar{Y} \in \mathrm{coNP}$
- coNP is the set of complements of languages in NP

What is coP?

A: The set of languages that are not in P

B: $\mathrm{coP} = \mathrm{P}$

C: The set of algorithms that do not run in polynomial time

D: The notion of "coP" doesn't make any sense

Respond at PollEv.com/whoza or text "whoza" to 22333

The complexity class $\mathbf{coNP}$

- Example: We say that a Boolean formula is unsatisfiable if it is not satisfiable
- Let $\text{3-UNSAT} = \{\langle\phi\rangle : \phi \text{ is an unsatisfiable 3-CNF formula}\}$
- Then $\text{3-UNSAT} \in \text{coNP}$, because a satisfying assignment is a certificate showing that $\phi \notin \text{3-UNSAT}$

FACTOR $\in$ coNP

- $\text{FACTOR} = \{\langle K, R\rangle : K \text{ has a prime factor } p \text{ such that } p \leq R\}$
- **Claim:** $\text{FACTOR} \in \text{coNP}$
- **Proof:** Given $\langle K, R\rangle$:
  - Nondeterministically guess numbers $d \leq \log K$ and $p_1, p_2, \ldots, p_d \leq K$
  - If $p_1, \ldots, p_d$ are prime, $p_1 \cdot p_2 \cdot p_3 \cdots p_d = K$, and $\min(p_1, \ldots, p_d) > R$, reject
  - Otherwise, accept

PRIMES $\in$ P