CS 2009

Design and Analysis of Algorithms

Waheed Ahmed

Email : waheedahmed@nu.edu.pk

Week 14.2:

NP-Completeness Reduction

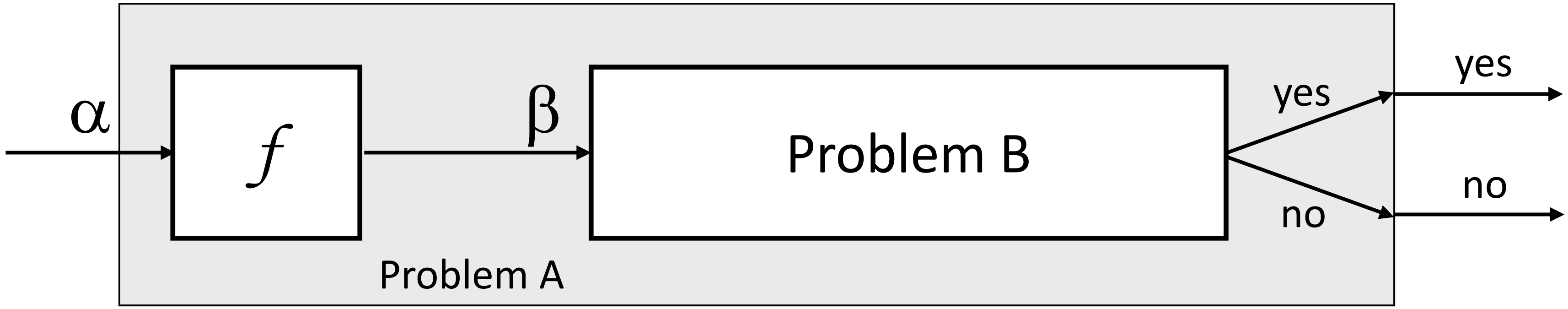
Thomas H. Coreman (CLRS), Chapter 34.

JON KLEINBERG, EVA TARDOS, Chapter 8.

Reductions

- Reduction is a way of saying that one problem is **"easier"** than another.
- We say that problem A is easier than problem B,
  (i.e., we write **"A $\leq$ B"**)

  if we can solve A using the algorithm that solves B.
- **Idea:** transform the inputs of A to inputs of B

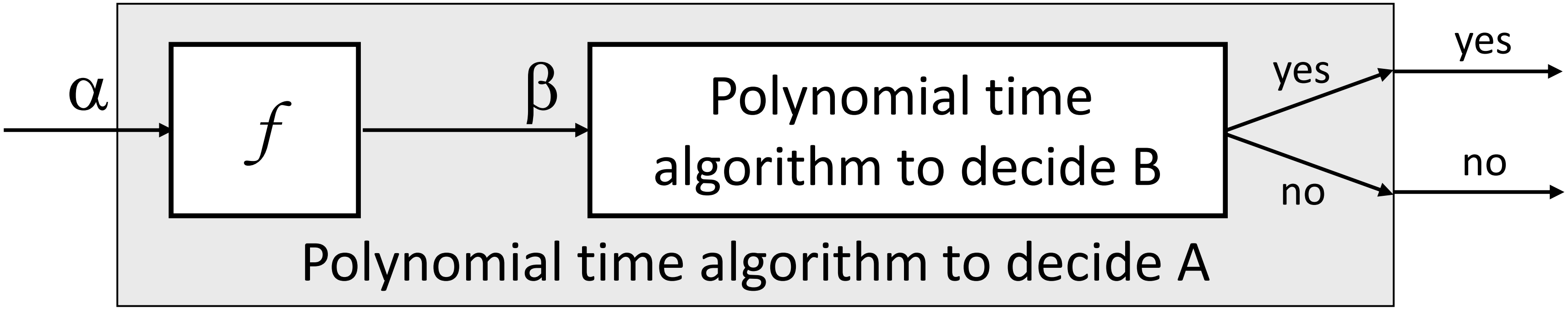


Polynomial Reductions

- Given two problems A, B, we say that A is polynomially **reducible** to B ($A \leq_p B$) if:

1. There exists a function f that converts the input of A to inputs of B in polynomial time
2. $A(i) = YES \Leftrightarrow B(f(i)) = YES$

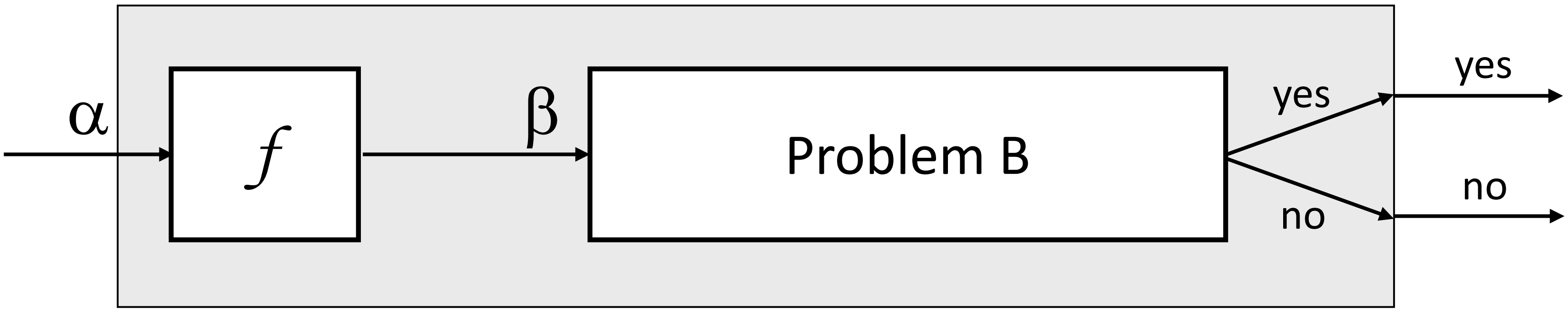
Polynomial Reduction



1. Use a **polynomial time** reduction algorithm to transform A into B
2. Run a known **polynomial time** algorithm for B
3. Use the answer for B as the answer for A

Implications of Reduction

- Problem A reduces to problem B if you can use an algorithm that solves B to help solve A.



Algorithm for Problem A

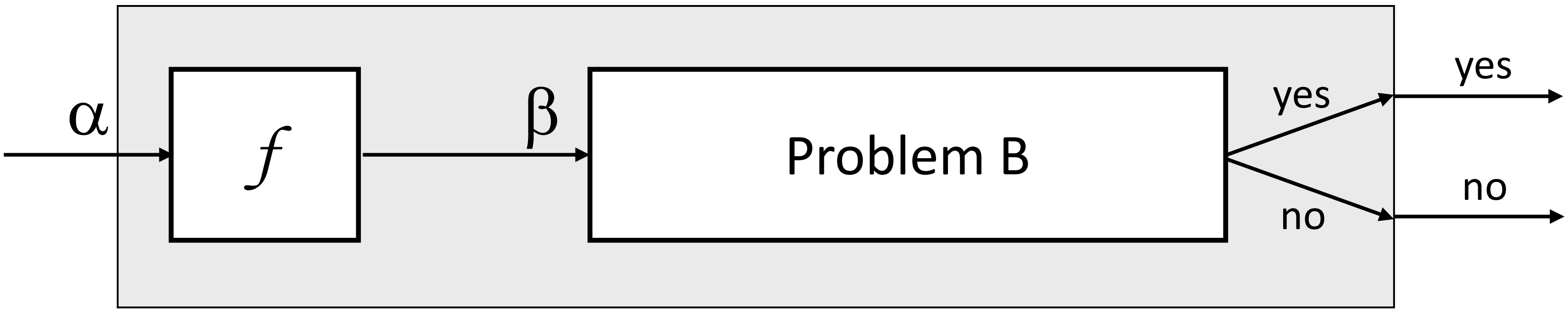
- If $A \leq_p B$ and $B \in P$, then $A \in P$

- if $A \leq_p B$ and $B \notin P$, then $A \notin P$

Cost of solving *A* = total cost of solving *B* + cost of reduction.

Reductions Examples

- Problem A reduces to problem B if you can use an algorithm that solves B to help solve A.



Algorithm for Problem A

- Example 1. [finding the median reduces to sorting]
- To find the median of *N* items:
  - Sort N items
  - Return item in the middle:

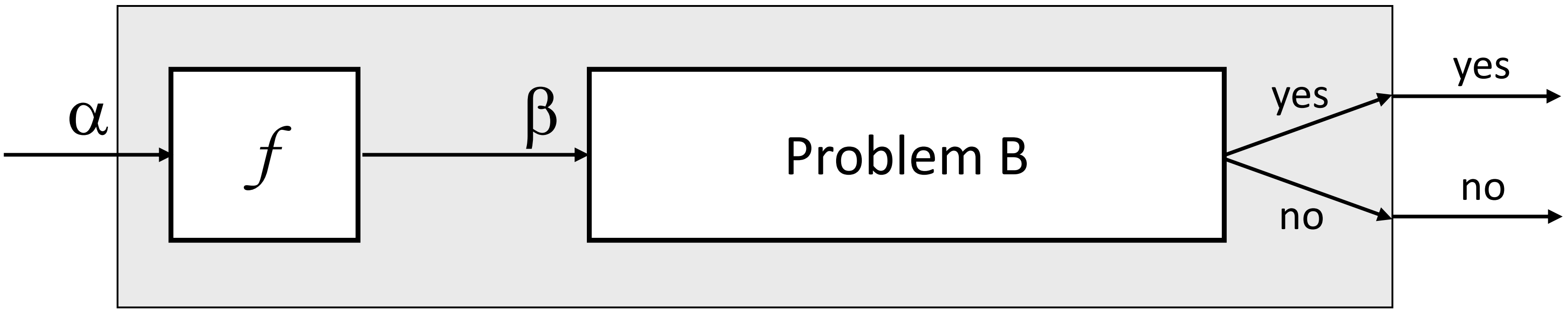
Cost of Sorting

Cost of Reduction

Cost of solving *finding median* = O(n log n) + 1

Reductions Examples

- Problem A reduces to problem B if you can use an algorithm that solves B to help solve A.



Algorithm for Problem A

- Example 2. [element distinctness reduces to sorting]
- To solve element distinctness on N items:
  – Sort N items
  – Check adjacent pairs for equality.

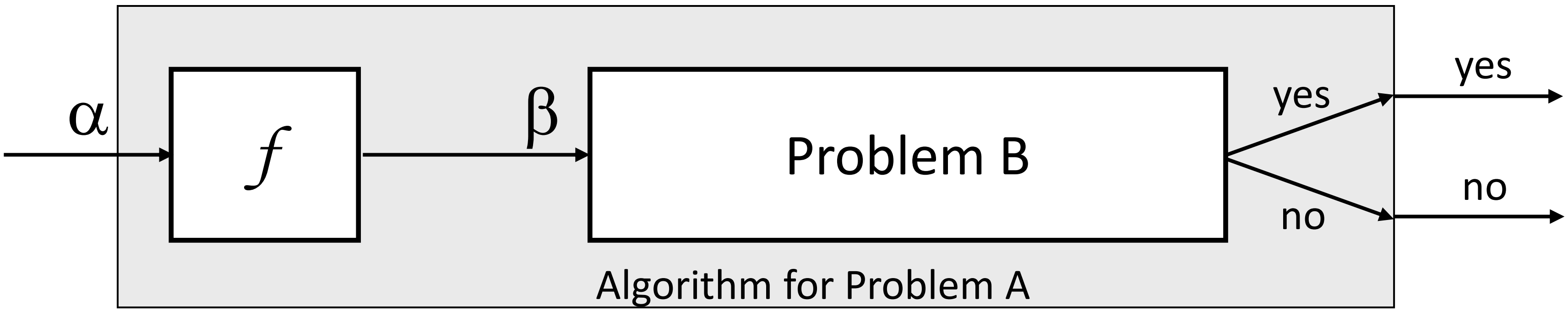
Cost of Sorting

Cost of Reduction

Cost of solving *element distinctness* = O(n log n) + O(n)

Reductions Examples

- Convex hull. Given N points in the plane, identify the extreme points of the convex hull (in counterclockwise order).



- Example 3. [Convex hull reduces to sorting.]
- To solve convex hull:
  - Choose point p with smallest (or largest) y-coordinate.
  - Sort points by polar angle with p.
  - Consider points in order, and discard those that would create a clockwise turn.

Cost of solving *element distinctness* = $\underbrace{O(n \log n)}_{\text{Cost of Sorting}} + \underbrace{O(n)}_{\text{Cost of Reduction}}$

RECIPE FOR PROVING PROBLEM Is NP-Complete

To Prove B Is NP-Complete

1. **Prove that B is a member of the class NP.** $B \in NP$
2. **Choose an NP-complete problem A.**
3. **Prove that there is a Levin reduction from A to B.**

Independent Set (Example 1)

INDEPENDENT-SET. Given a graph $G = (V, E)$ and an integer k, is there a subset of k (or more) vertices such that no two are adjacent?

Ex. Is there an independent set of size ≥ 6 ?

Ex. Is there an independent set of size ≥ 7 ?

{1} is independent Set

{2, 3} is independent Set

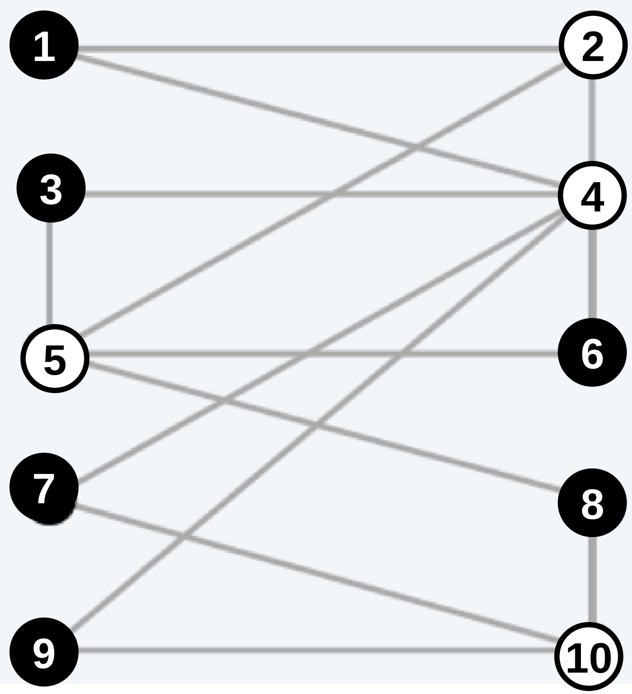
{1, 3} is independent Set

{1, 3, 6} is independent Set

~~{2, 5, 6, 7} is independent Set~~

{1, 3, 6, 7} is independent Set

{1, 3, 6, 7, 8 , 9} is independent Set



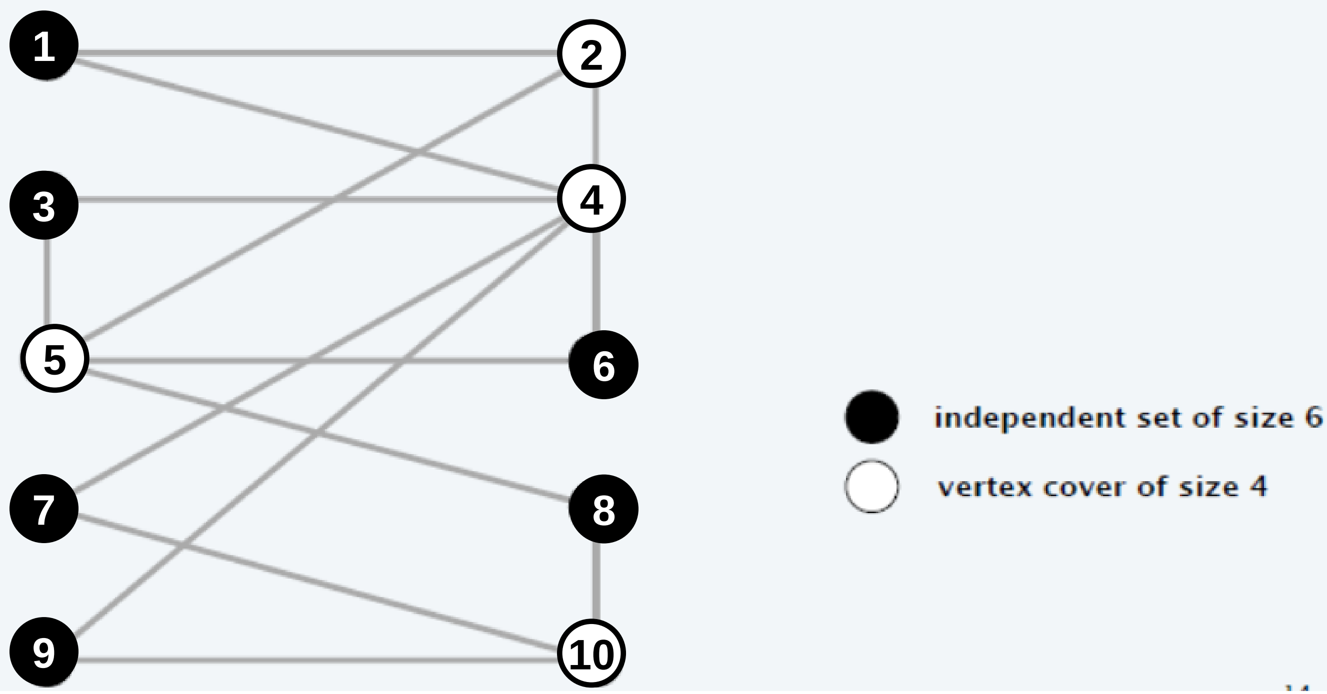
independent set of size 6

Vertex Cover (Example 1)

VERTEX-COVER. Given a graph $G = (V, E)$ and an integer k, is there a subset of k (or fewer) vertices such that each edge is incident to at least one vertex in the subset?

Ex. Is there a vertex cover of size ≤ 4 ?

Ex. Is there a vertex cover of size ≤ 3 ?



Vertex Cover is NP-Complete

Recipe to Prove a Problem Is NP-Complete

- Prove that a Vertex Cover is NP-Complete
- Step 1. Vertex Cover ∈ NP
- Step 2. Choose an NP-Complete problem A (Independent Set).

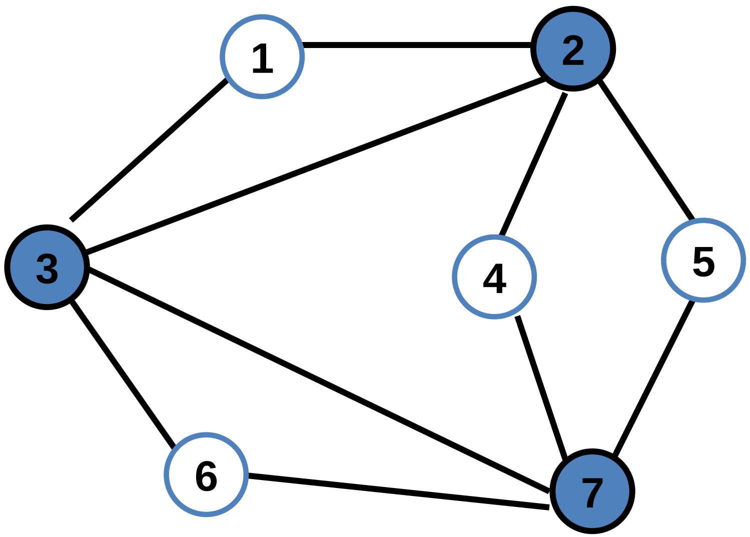
  Prove that A (Independent set) reduces to B (Vertex Cover)

  INDEPENDENT-SET $\leq_p$ VERTEX-COVER.

Step 1: Vertex Cover ∈ NP

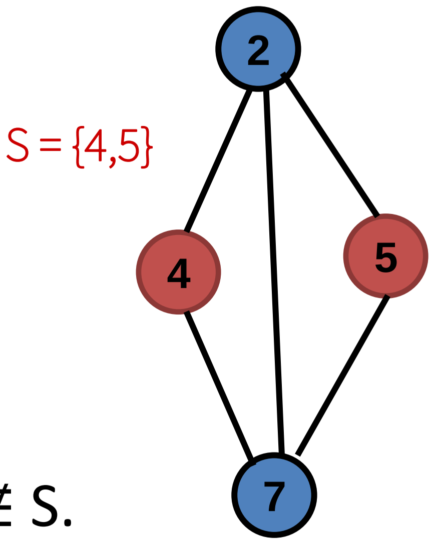
- Given Graph G = (V, E) contains Vertex cover of Size 3?

S is vertex cover if every edge in E has at least one endpoint in S. (Example V.C = {2,3,7})



Vertex cover and independent set reduce to one another

- **Lemma:** INDEPENDENT-SET $\leq_p$ VERTEX-COVER.
- **Proof:** We show *S* is an independent set of size *k*, iff $V - S$ is a vertex cover of size $n - k$.



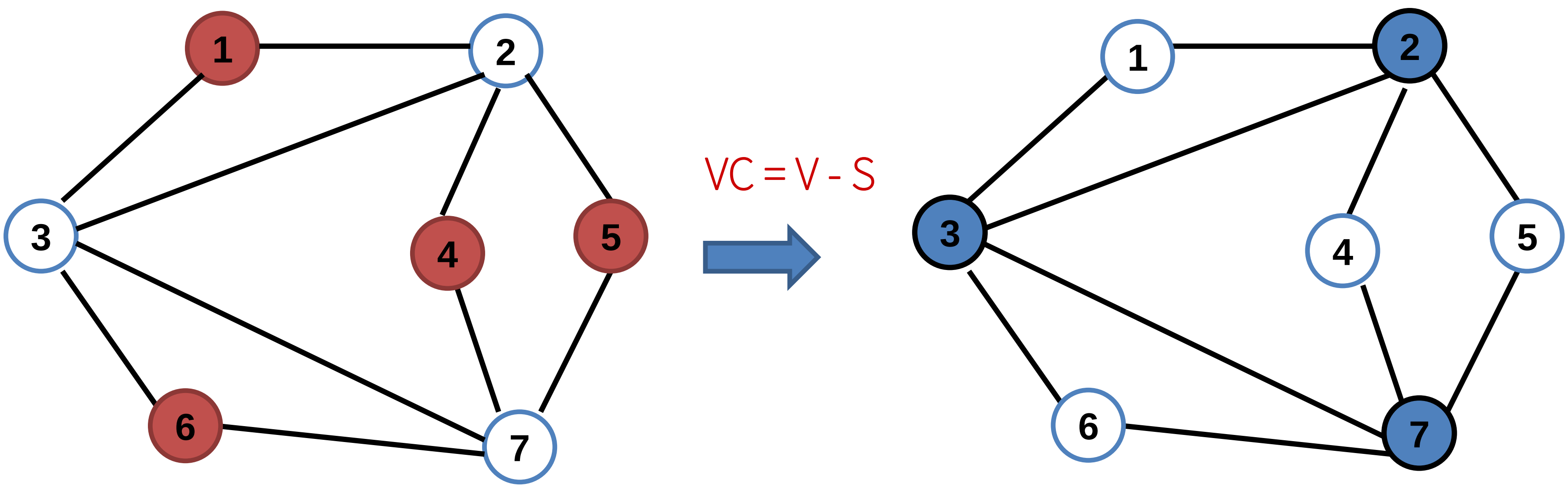
  – Let S be any independent set of size k.
  – Consider an arbitrary edge $(u, v) \in E$.
  – S independent $\Rightarrow$ either $u \notin S$, or $v \notin S$, or both $\notin S$.
  – Vertex Cover (V-S) $\Rightarrow$ either $u \in V - S$, or $v \in V - S$, or both $\in V - S$.
  – Thus, $V - S$ covers (u, v). ▪

Independent Set (Example 2)

- S is independent if there are no edges between vertices in S. (Example I.S = {1,4,5,6})

V.C = V – S

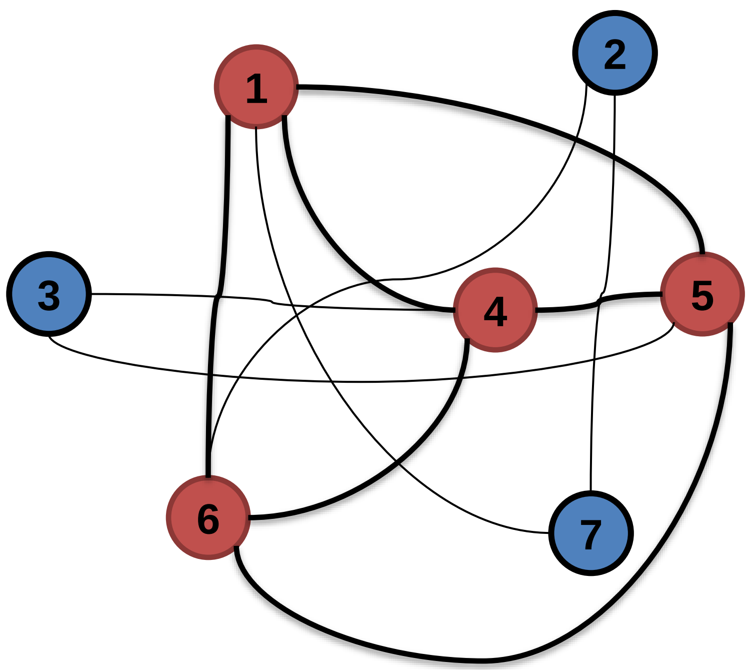
(V.C ={2,3,7})



Clique (Example 2)

- Clique
  - Graph G = (V, E), a subset S of the vertices is a clique if there is an edge between every pair of vertices in S. (Example Clique = {1,4,5,6})

Clique
of size 4



Clique is NP-Complete

Recipe to Prove a Problem Is NP-Complete

- Prove that a Clique is NP-Complete
- Step 1. Clique $\in$ NP.
- Step 2. Choose known NP-Complete problem A (IS).

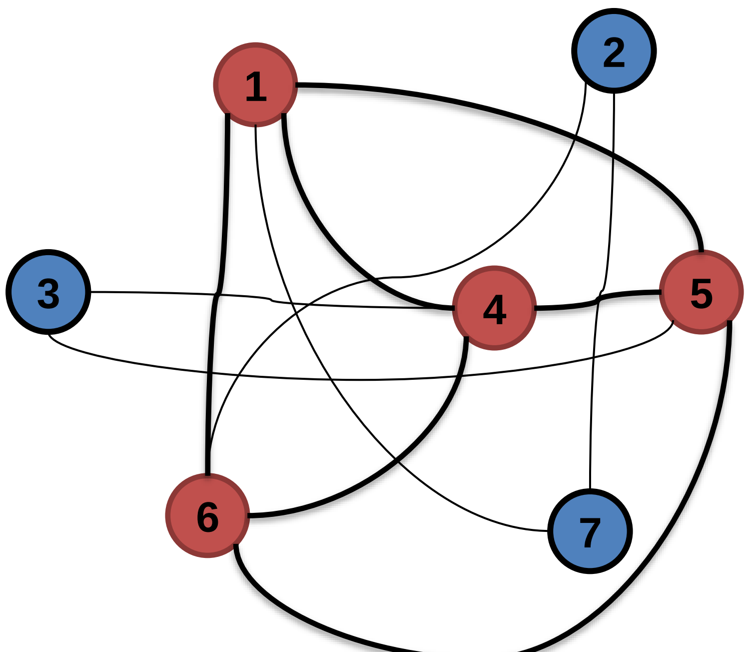
  Prove that A (IS) reduces to B (Clique)

  IS $\leq_p$ Clique.

Step 1: Clique ∈ NP.

- Graph G = (V, E), a subset S of the vertices is a clique if there is an edge between every pair of vertices in S. (Example Clique = {1,4,5,6})

- Given graph contains Clique of size = 4 ?



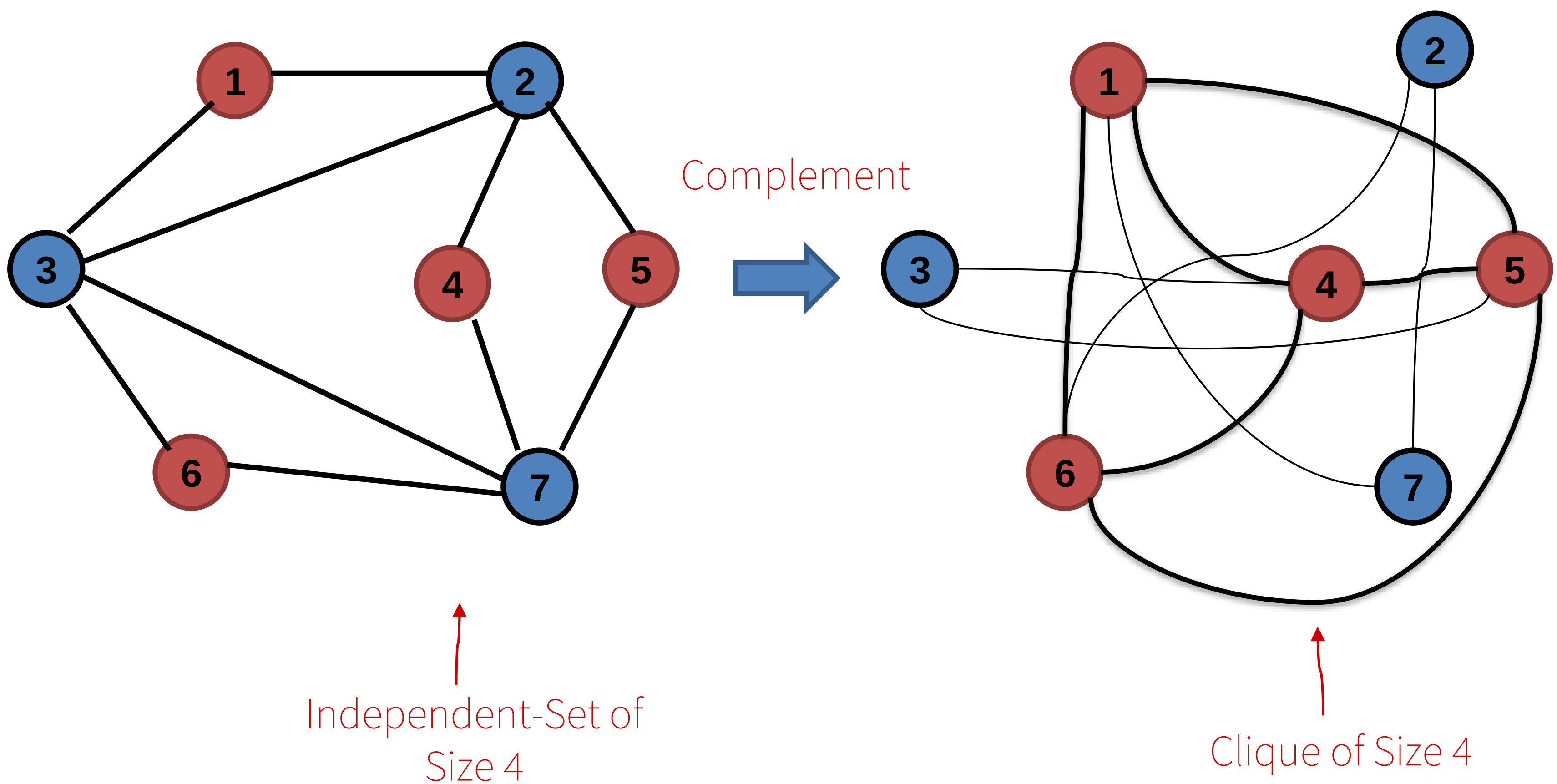
Step 2: IS $\leq_p$ Clique

- Lemma: S is Independent in G iff S is a Clique in the complement of G

- To reduce IS to Clique, we compute the complement of the graph. The complement has a clique of size K iff the original graph has an independent set of size K.

- Construction of Complement of the graph can easily be done in polynomial time.

IS $\leq_p$ Clique (Example 2)



Reduction: Independent Set, Vertex Cover, and Clique (Example 3)

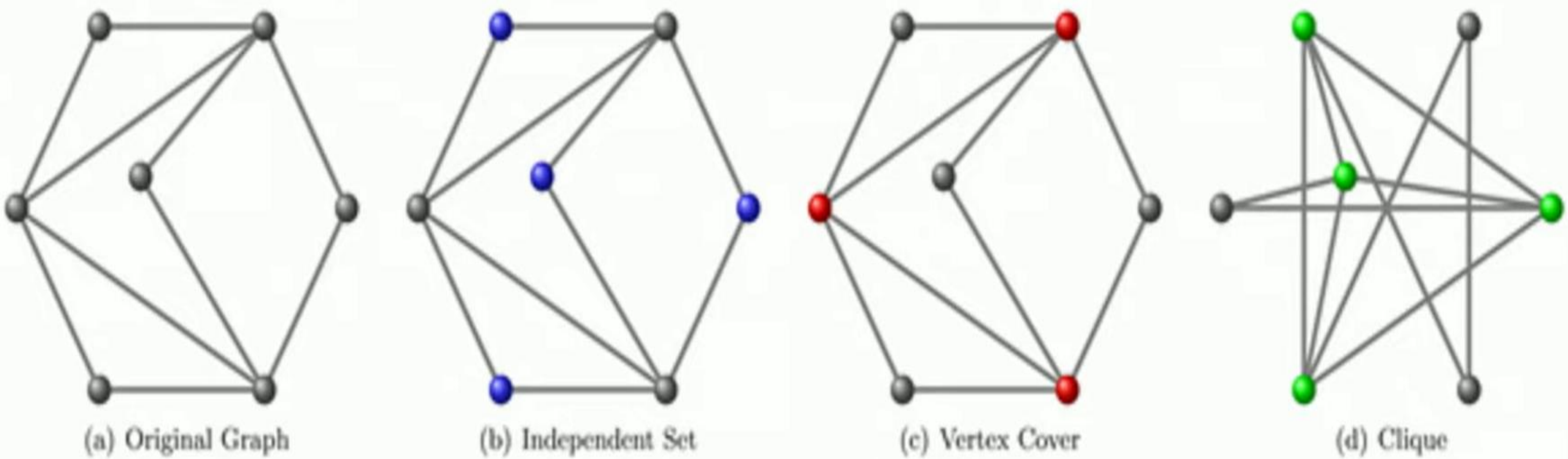


Figure 1: Relations among Independent Set, Vertex Cover, and Clique

INDEPENDENT-SET is NP-Complete

Recipe to Prove a Problem Is NP-Complete

- To prove that a problem B (INDEPENDENT-SET) is NP-Complete:
- Step 1. INDEPENDENT-SET $\in$ NP
- Step 2. Choose an NP-Complete problem A (3-SAT).

  Prove that A (3-SAT) reduces to B (INDEPENDENT-SET)

  3-SAT $\leq_p$ INDEPENDENT-SET

Satisfiability

Literal. A Boolean variable or its negation. x_i or $\overline{x_i}$

Clause. A disjunction of literals. $C_j = x_1 \lor \overline{x_2} \lor x_3$

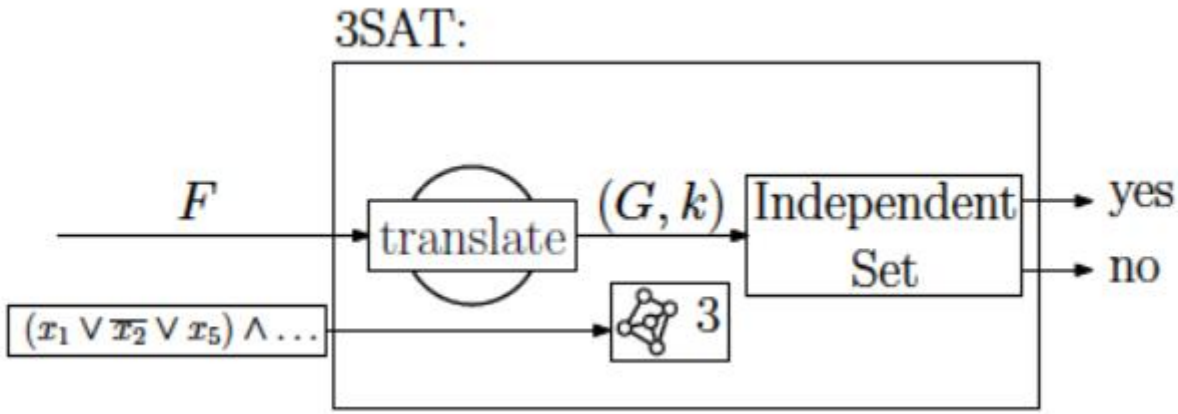
Conjunctive normal form (CNF). A propositional formula Φ that is a conjunction of clauses. $\Phi = C_1 \land C_2 \land C_3 \land C_4$

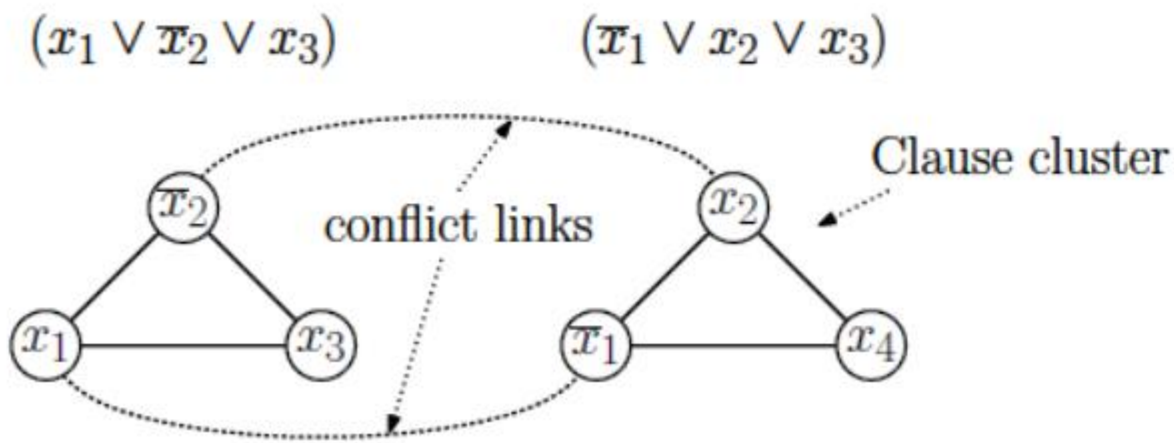
SAT. Given a CNF formula Φ, does it have a satisfying truth assignment?
3-SAT. SAT where each clause contains exactly 3 literals (and each literal corresponds to a different variable).

$$\Phi = (\overline{x_1} \lor x_2 \lor x_3) \land (x_1 \lor \overline{x_2} \lor x_3) \land (\overline{x_1} \lor x_2 \lor x_4)$$

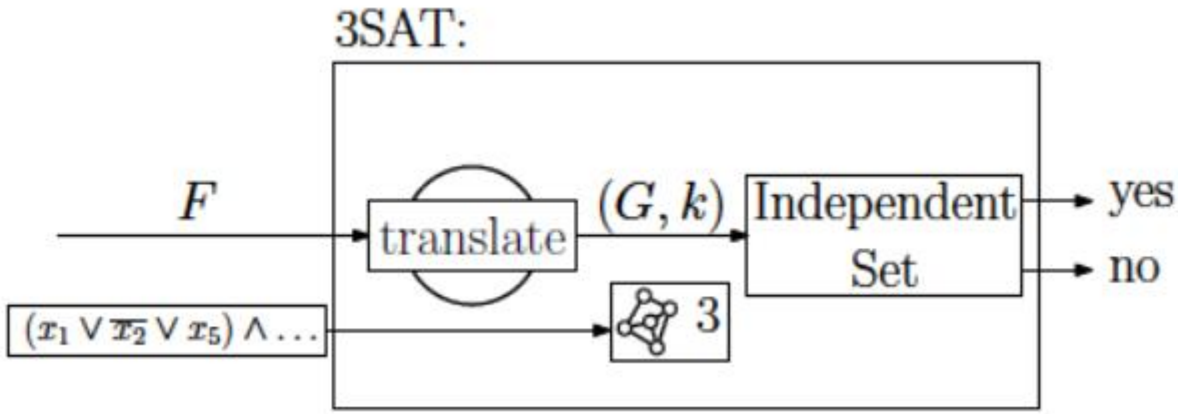
yes instance: x_1 = true, x_2 = true, x_3 = false, x_4 = false

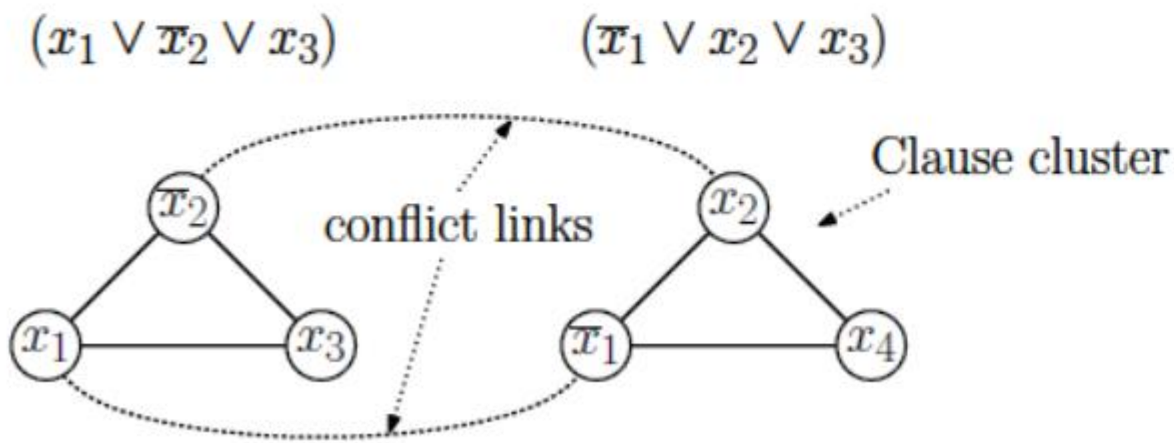
3-satisfiability reduces to independent set





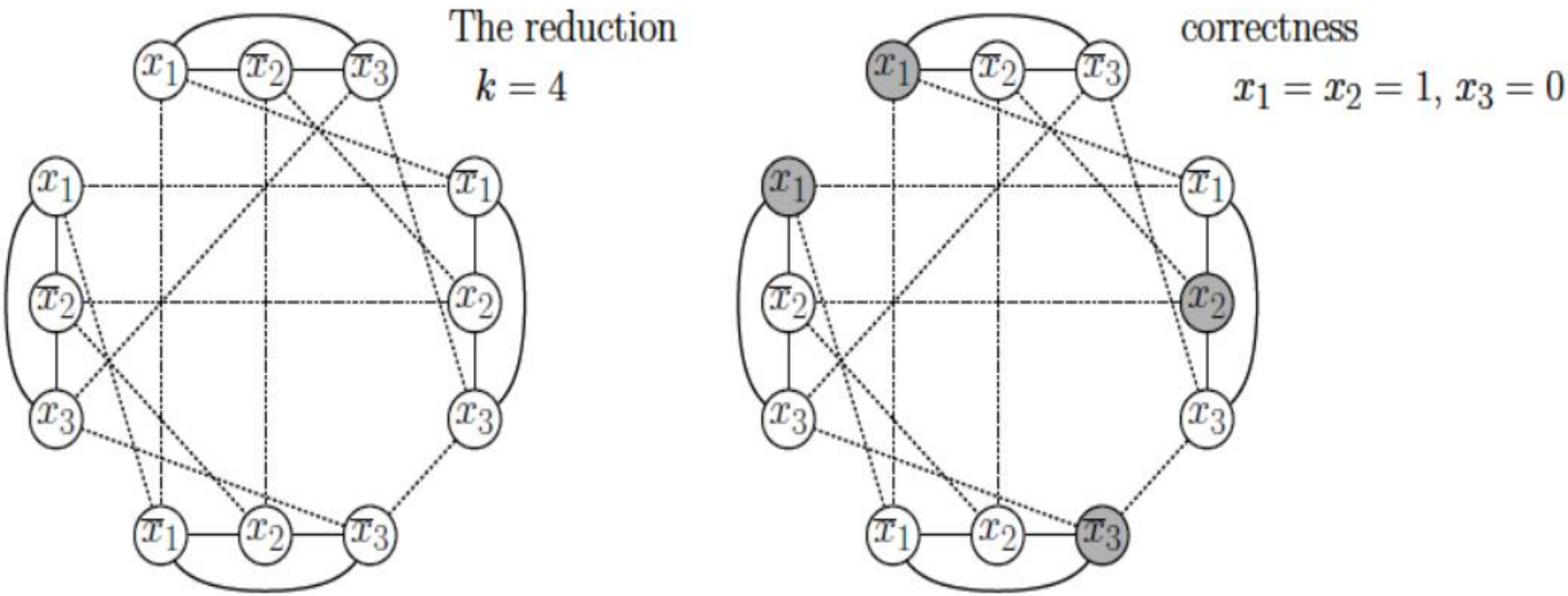
3-satisfiability reduces to independent set





3-satisfiability reduces to independent set

$$F = (x_1 \vee \overline{x}_2 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee x_2 \vee x_3) \wedge (\overline{x}_1 \vee x_2 \vee \overline{x}_3) \wedge (x_1 \vee \overline{x}_2 \vee x_3).$$



(a) (b)

The Reduction

1. G_φ will have one vertex for each literal in a clause
2. Connect the 3 literals in a clause to form a triangle; the independent set will pick at most one vertex from each clause, which will correspond to the literal to be set to true
3. Connect 2 vertices if they label complementary literals; this ensures that the literals corresponding to the independent set do not have a conflict
4. Take k to be the number of clauses

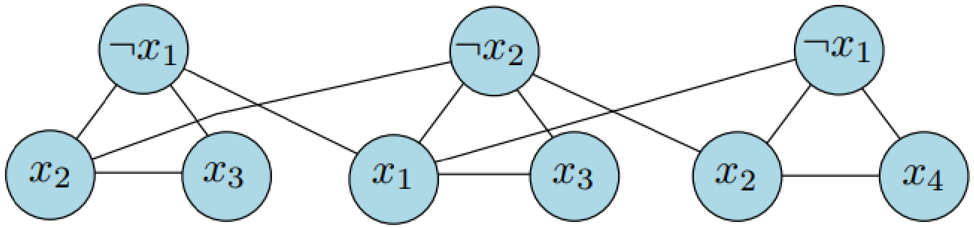


Figure: Graph for $\varphi = (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4)$

Correctness

Proposition 23.2.

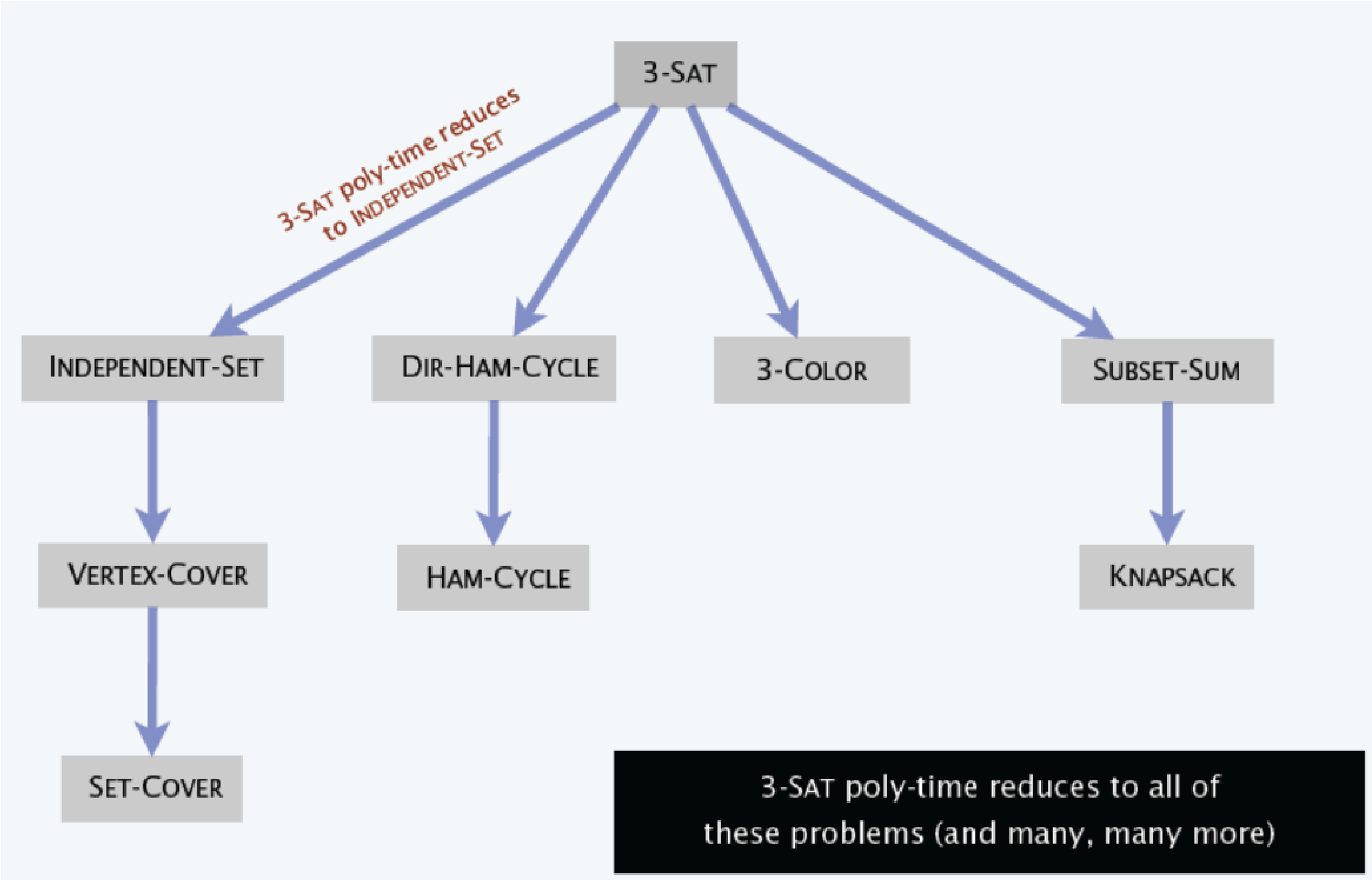
φ is satisfiable iff $\boldsymbol{G_\varphi}$ has an independent set of size $\boldsymbol{k}$ ($=$ number of clauses in φ).

Proof.

$\Leftarrow$ Let $\boldsymbol{S}$ be an independent set of size $\boldsymbol{k}$

1. $\boldsymbol{S}$ must contain <u>exactly</u> one vertex from each clause
2. $\boldsymbol{S}$ cannot contain vertices labeled by conflicting literals
3. Thus, it is possible to obtain a truth assignment that makes in the literals in $\boldsymbol{S}$ true; such an assignment satisfies one literal in every clause $\square$

Poly-time reductions



Cook-Levin theorem shows that 3-SAT is a "universal" problem