# CS 381 – FALL 2019

## Week 13.1, Monday, Nov 11

**Homework 6 Due: November ~~14th~~ 15th at 11:59PM (Gradescope)**
**Homework 7 Released: November 15th**

# 8.1 Polynomial-Time Reductions

# Clicker Review: Polynomial-Time Reduction

Reduction. Problem X polynomial reduces to problem Y if arbitrary instances of problem X can be solved using:

- Polynomial number of standard computational steps, plus
- Polynomial number of calls to oracle that solves problem Y.

Suppose that X polynomial reduces to problem Y ($X \leq_P Y$). Which of the following claims are true?

A. If X can be solved in polynomial time then Y can also be solved in polynomial time.
B. If X cannot be solved in polynomial time then Y cannot be solved in polynomial time.
C. If Y cannot be solved in polynomial time then X cannot be solved in polynomial time.
D. If Y can be solved in time $O(n^2)$ then X can be solved in time $O(n)$
E. There is a DP algorithm to solve both problems in polynomial time

# Reduction By Simple Equivalence

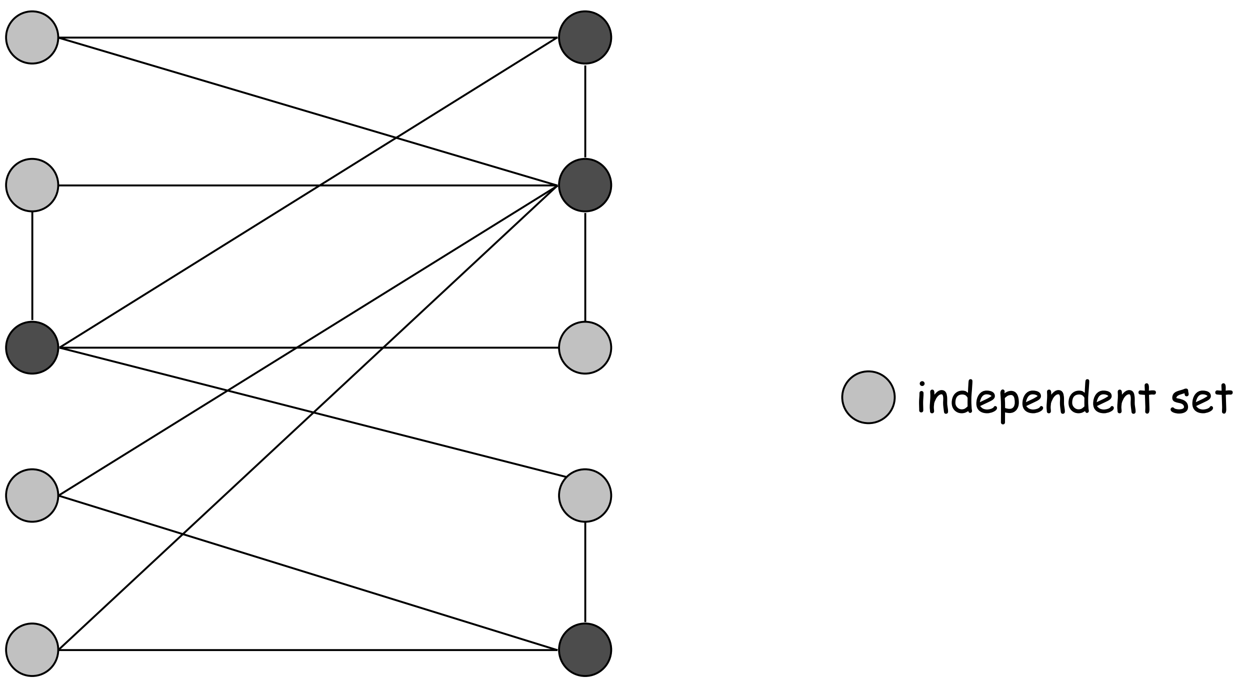
Basic reduction strategies.

- Reduction by simple equivalence.
- Reduction from special case to general case.
- Reduction by encoding with gadgets.

# Independent Set

INDEPENDENT SET: Given a graph $G = (V, E)$ and an integer $k$, is there a subset of vertices $S \subseteq V$ such that $|S| \geq k$, and for each edge at most one of its endpoints is in $S$?

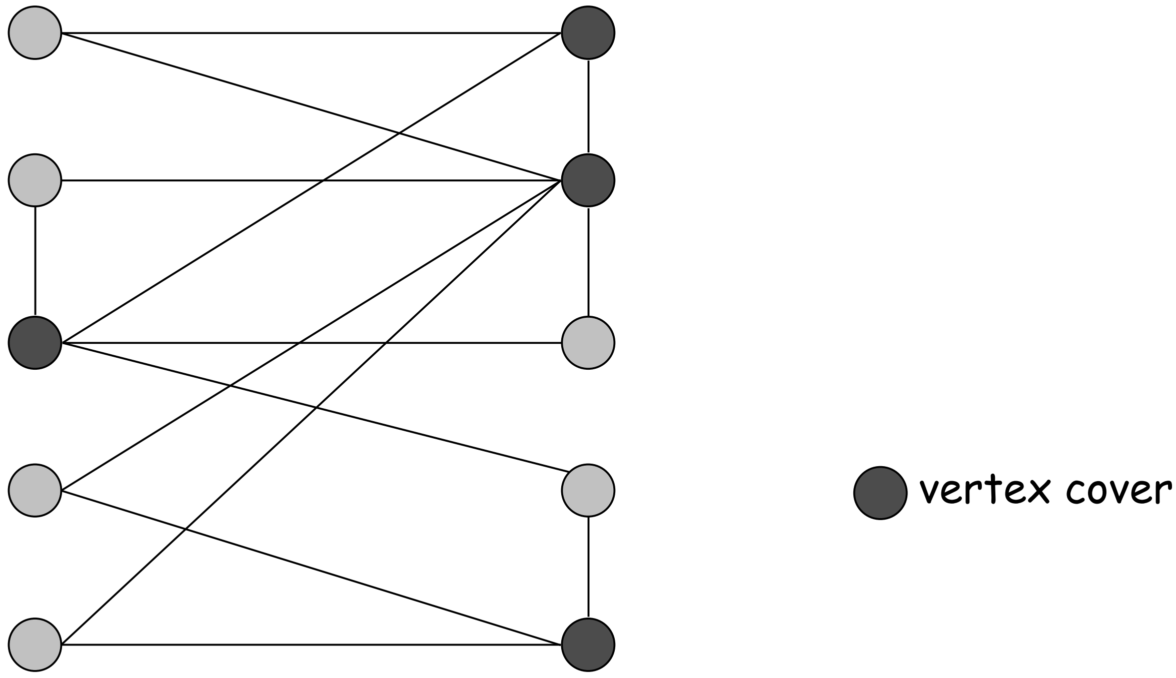
Ex. Is there an independent set of size $\geq 6$? Yes.
Ex. Is there an independent set of size $\geq 7$? No.



# Vertex Cover

VERTEX COVER: Given a graph $G = (V, E)$ and an integer $k$, is there a subset of vertices $S \subseteq V$ such that $|S| \leq k$, and for each edge, at least one of its endpoints is in $S$?

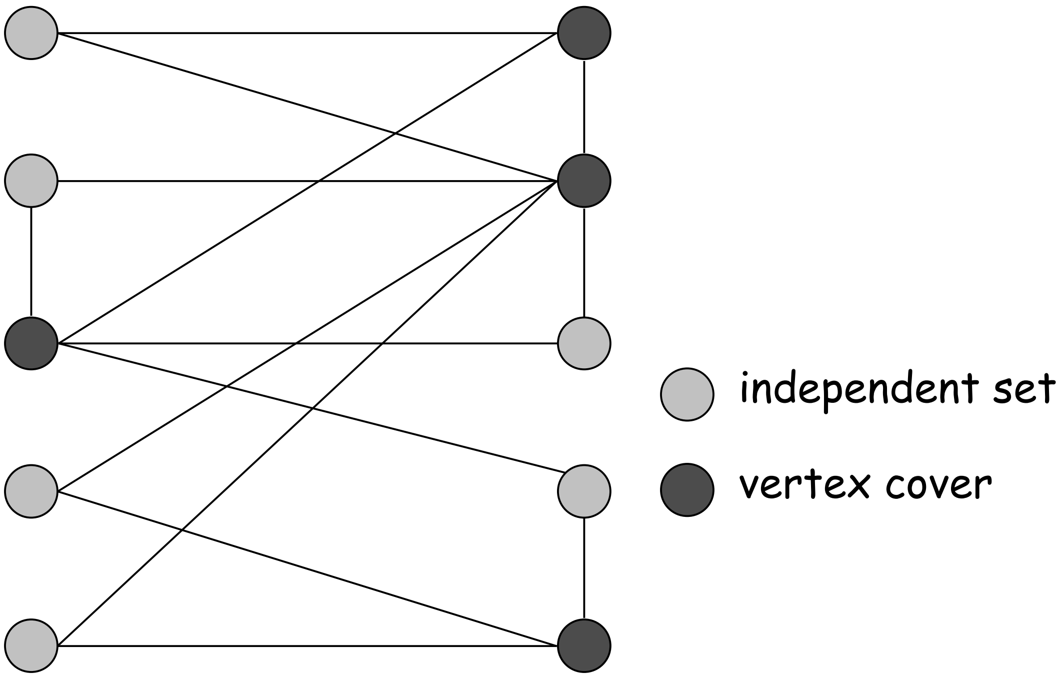
Ex. Is there a vertex cover of size $\leq 4$? Yes.
Ex. Is there a vertex cover of size $\leq 3$? No.



# Vertex Cover and Independent Set

Claim. VERTEX-COVER $\equiv_P$ INDEPENDENT-SET.

Pf. We show S is an independent set iff V – S is a vertex cover.



# Vertex Cover and Independent Set

Claim. VERTEX-COVER $\equiv_P$ INDEPENDENT-SET.

Pf. We show S is an independent set iff V – S is a vertex cover.
(G has VC of size k iff G has independent set of size n-k)

$\Rightarrow$

- Let S be any independent set.
- Consider an arbitrary edge (u, v).
- S independent $\Rightarrow$ u $\notin$ S or v $\notin$ S $\Rightarrow$ either u $\in$ V – S or v $\in$ V – S.
- Thus, V – S covers (u, v).

$\Leftarrow$

- Let V – S be any vertex cover.
- Consider two nodes u $\in$ S and v $\in$ S.
- Observe that (u, v) $\notin$ E since V – S is a vertex cover.
- Thus, no two nodes in S are joined by an edge $\Rightarrow$ S is an independent set. ▪

# Reduction from Special Case to General Case

Basic reduction strategies.

- Reduction by simple equivalence.
- Reduction from special case to general case.
- Reduction by encoding with gadgets.

# Set Cover

SET COVER: Given a set U of elements, a collection $S_1, S_2, \ldots, S_m$ of subsets of U, and an integer k, does there exist a collection of $\leq$ k of these sets whose union is equal to U?

Sample application.

- m available pieces of software.
- Set U of n capabilities that we would like our system to have.
- The ith piece of software provides the set $S_i \subseteq U$ of capabilities.
- Goal: achieve all n capabilities using fewest pieces of software.

Ex:

$U = \{1, 2, 3, 4, 5, 6, 7\}$

$k = 2$

| | |
|---|---|
| $S_1 = \{3, 7\}$ | $S_4 = \{2, 4\}$ |
| $S_2 = \{3, 4, 5, 6\}$ | $S_5 = \{5\}$ |
| $S_3 = \{1\}$ | $S_6 = \{1, 2, 6, 7\}$ |

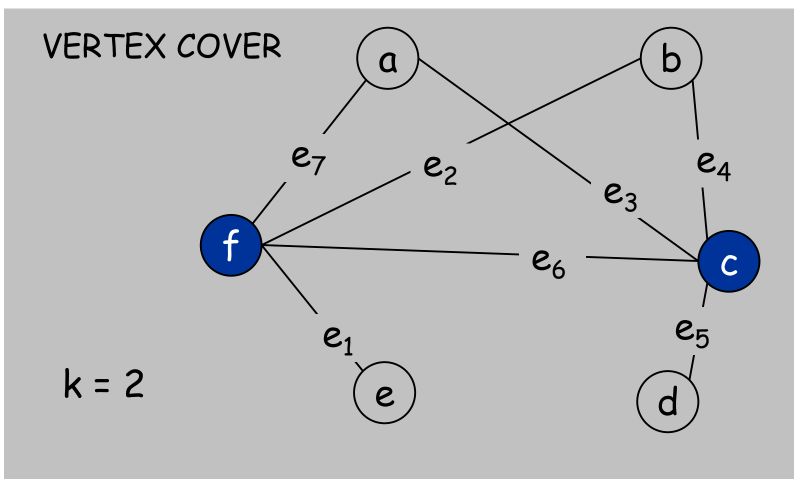
# Vertex Cover Reduces to Set Cover

**Claim.** VERTEX-COVER $\leq_P$ SET-COVER.

**Pf.** Given a VERTEX-COVER instance $G = (V, E)$, $k$, we construct a set cover instance whose size equals the size of the vertex cover instance.

**Construction.**

- Create SET-COVER instance:
  - $k = k$, $U = E$, $S_v = \{e \in E : e \text{ incident to } v\}$
- Set-cover of size $\leq k$ iff vertex cover of size $\leq k$. ▪




SET COVER

$U = \{1, 2, 3, 4, 5, 6, 7\}$
$k = 2$

$S_a = \{3, 7\}$    $S_b = \{2, 4\}$
$S_c = \{3, 4, 5, 6\}$    $S_d = \{5\}$
$S_e = \{1\}$    $S_f = \{1, 2, 6, 7\}$

# Vertex Cover Reduces to Set Cover

Claim. VERTEX-COVER $\leq_P$ SET-COVER.

Assume that SET-COVER cannot be solved in polynomial time. Based only on the above claim which of the following claims are necessarily true?

A. VERTEX-COVER is solvable in polynomial time.
B. VERTEX-COVER is not solvable in polynomial time.
C. If we also had a reduction SET-COVER $\leq_P$ VERTEX-COVER then we could conclude that VERTEX-COVER is solvable in polynomial time
D. If we also had a reduction SET-COVER $\leq_P$ VERTEX-COVER then we can conclude that VERTEX-COVER is not solvable in polynomial time
E. None of the above claims are true

# Vertex Cover Reduces to Set Cover

Claim. VERTEX-COVER $\leq_P$ SET-COVER.

Assume that SET-COVER cannot be solved in polynomial time. Based only on the above claim which of the following claims are necessarily true?

A. VERTEX-COVER is solvable in polynomial time.
B. VERTEX-COVER is not solvable in polynomial time. (We need to have a reduction in the opposite direction to conclude this.)
C. If we also had a reduction SET-COVER $\leq_P$ VERTEX-COVER then we could conclude that VERTEX-COVER is solvable in polynomial time
D. If we also had a reduction SET-COVER $\leq_P$ VERTEX-COVER then we can conclude that VERTEX-COVER is not solvable in polynomial time (**Note**: We do have such a reduction from SET-COVER to VERTEX-COVER)
E. None of the above claims are true

# Polynomial-Time Reduction

Basic strategies.

- Reduction by simple equivalence.
- Reduction from special case to general case.
- Reduction by encoding with gadgets.

# 8.2 Reductions via "Gadgets"

Basic reduction strategies.

- Reduction by simple equivalence.
- Reduction from special case to general case.
- Reduction via "gadgets."

# Satisfiability

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

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

Conjunctive normal form: A propositional formula $\Phi$ that is the conjunction of clauses. $\Phi = C_1 \wedge C_2 \wedge C_3 \wedge C_4$

SAT: Given CNF formula $\Phi$, does it have a satisfying truth assignment?

3-SAT: SAT where each clause contains (*at most*) 3 literals.

each corresponds to a different variable

Ex: $\left(\overline{x_1} \vee x_2 \vee x_3\right) \wedge \left(x_1 \vee \overline{x_2} \vee x_3\right) \wedge \left(x_2 \vee x_3\right) \wedge \left(\overline{x_1} \vee \overline{x_2} \vee \overline{x_3}\right)$

Yes: $x_1$ = true, $x_2$ = true, $x_3$ = false.

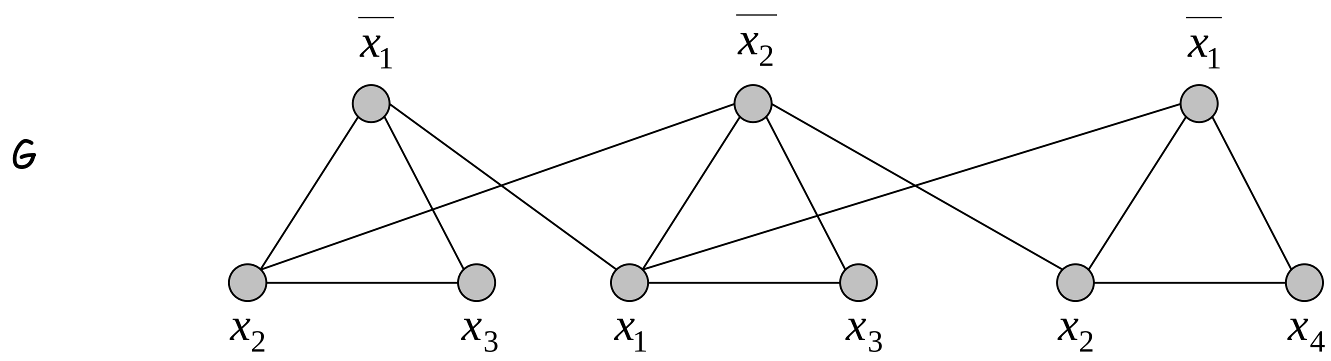
# 3 Satisfiability Reduces to Independent Set

Claim. 3-SAT $\leq_P$ INDEPENDENT-SET.

Pf. Given an instance $\Phi$ of 3-SAT, we construct an instance (G, k) of INDEPENDENT-SET that has an independent set of size k iff $\Phi$ is satisfiable.

Construction.

- G contains 3 vertices for each clause, one for each literal.
- Connect 3 literals in a clause in a triangle.
- Connect literal to each of its negations.




k = 3

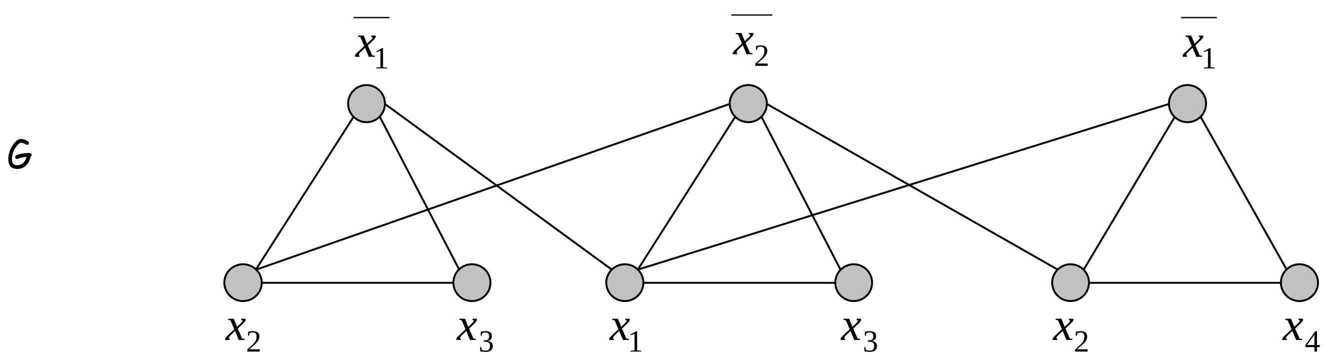
$$\Phi = \left(\overline{x_1} \vee x_2 \vee x_3\right) \wedge \left(x_1 \vee \overline{x_2} \vee x_3\right) \wedge \left(\overline{x_1} \vee x_2 \vee x_4\right)$$

# 3 Satisfiability Reduces to Independent Set

Claim. G contains independent set of size k = $|\Phi|$ iff $\Phi$ is satisfiable.

Pf. $\Rightarrow$ Let S be independent set of size k.

- S must contain exactly one vertex in each triangle.
- Set these literals to true. $\leftarrow$ and any other variables in a consistent way
- Truth assignment is consistent and all clauses are satisfied.




k = 3

$$\Phi = \left(\overline{x_1} \vee x_2 \vee x_3\right) \wedge \left(x_1 \vee \overline{x_2} \vee x_3\right) \wedge \left(\overline{x_1} \vee x_2 \vee x_4\right)$$

# 3 Satisfiability Reduces to Independent Set

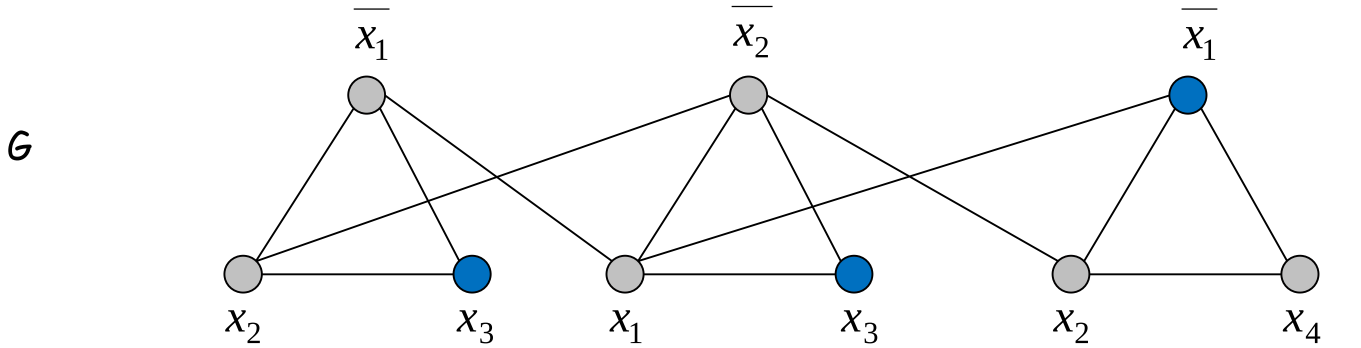
Claim. G contains independent set of size k = $|\Phi|$ iff $\Phi$ is satisfiable.

Pf. $\Rightarrow$ Let S be independent set of size k.

- S must contain exactly one vertex in each triangle.
- Set these literals to true. $\leftarrow$ and any other variables in a consistent way
- Truth assignment is consistent and all clauses are satisfied.

Independent Set S has size k=3

→ Set $x_3$=true, $x_1$=false and $x_2$ = (arbitrary)




k = 3

$$\Phi = \left(\overline{x_1} \vee x_2 \vee x_3\right) \wedge \left(x_1 \vee \overline{x_2} \vee x_3\right) \wedge \left(\overline{x_1} \vee x_2 \vee x_4\right)$$

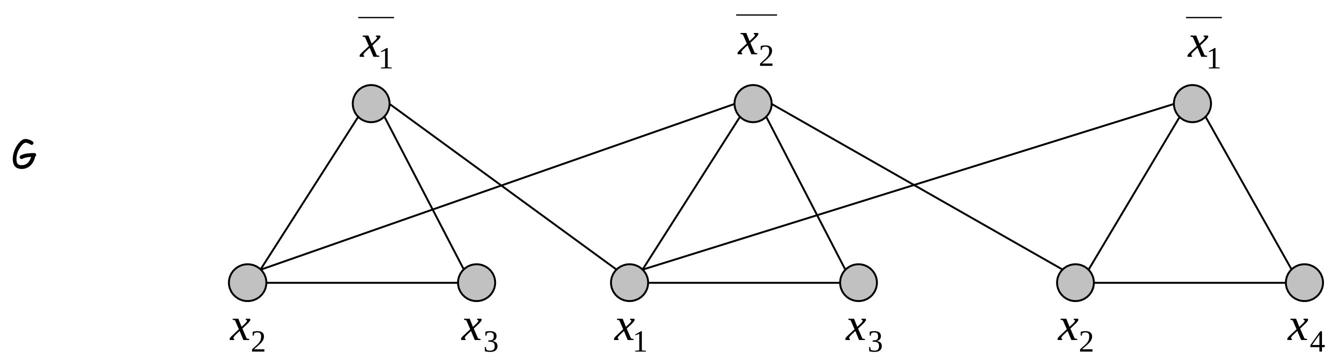
# 3 Satisfiability Reduces to Independent Set

Claim. G contains independent set of size k = $|\Phi|$ iff $\Phi$ is satisfiable.

Pf. $\Rightarrow$ Let S be independent set of size k.

- S must contain exactly one vertex in each triangle.
- Set these literals to true. $\longleftarrow$ and any other variables in a consistent way
- Truth assignment is consistent and all clauses are satisfied.
  - How do we know we don't assign $x_i$=true and $x_i$=false?

Pf $\Leftarrow$ Given satisfying assignment, select one true literal from each triangle. This is an independent set of size k. (Why independent?) ▪




k = 3

$$\Phi = (\overline{x_1} \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee x_4)$$

# Review

Basic reduction strategies.

- Simple equivalence: INDEPENDENT-SET $\equiv_P$ VERTEX-COVER.
- Special case to general case: VERTEX-COVER $\leq_P$ SET-COVER.
- Encoding with gadgets: 3-SAT $\leq_P$ INDEPENDENT-SET.

Transitivity. If $X \leq_P Y$ and $Y \leq_P Z$, then $X \leq_P Z$.
Pf idea. Compose the two algorithms.

Ex: 3-SAT $\leq_P$ INDEPENDENT-SET $\leq_P$ VERTEX-COVER $\leq_P$ SET-COVER.

**Implication 1:** If we have a polynomial time algorithm for SET-COVER then we have a polynomial time algorithm for 3-SAT, INDEPENDENT-SET and VERTEX-COVER.

**Implication 2:** If there is no polynomial time algorithm for 3-SAT then there is no polynomial time algorithm for INDEPENDENT-SET, VERTEX-COVER or SET-COVER.

# Self-Reducibility

Decision problem. Does there exist a vertex cover of size $\leq$ k?

Search problem. Find vertex cover of minimum cardinality.

Self-reducibility. Search problem $\leq_P$ decision version.

- Applies to all (NP-complete) problems in this course.
- Justifies our focus on decision problems.

Ex: to find min cardinality vertex cover.

- (Binary) search for cardinality k* of min vertex cover.
  - Oracle Query 1: Does G have a vertex cover of size k=n/2? etc…
- Find a vertex v such that $G - \{ v \}$ has a vertex cover of size $\leq$ k* - 1.
  - any vertex in any min vertex cover will have this property
- Include v in the vertex cover.
- Recursively find a min vertex cover in $G - \{ v \}$.

delete v and all incident edges