Intractability: Reduction

Adapted from Kevin Wayne's slides

Question: which problems will we be able to solve in practice?

A working definition: those with **polynomial** runtime algorithms

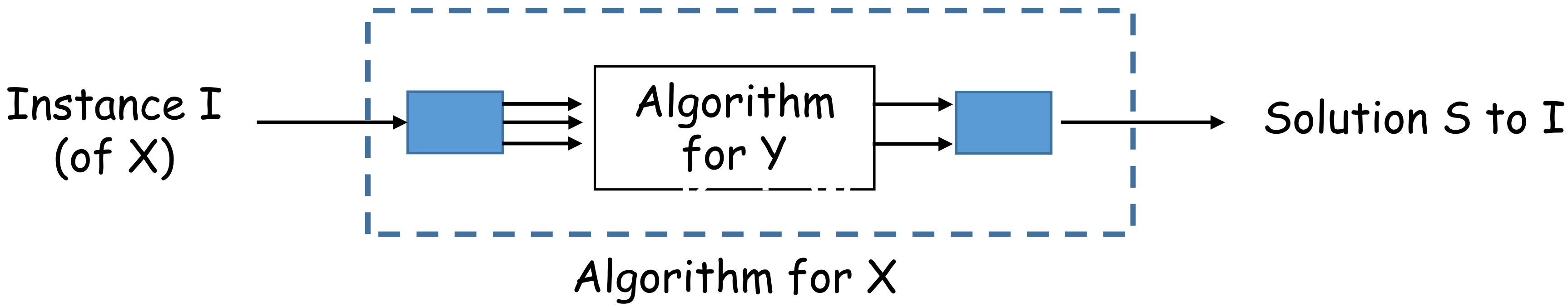
Question: suppose we can solve problem Y in poly-time.

What else can we solve in polynomial time?

Answer: any problem that can reduce to problem Y in polynomial time

<u>Reduction</u>: Problem X polynomial-time reduces to problem Y if any instance of X can be solved using

- Polynomial number of standard computational steps and
- Polynomial number of calls to a oracle that solves problem Y



Runtime of Algorithm for X?

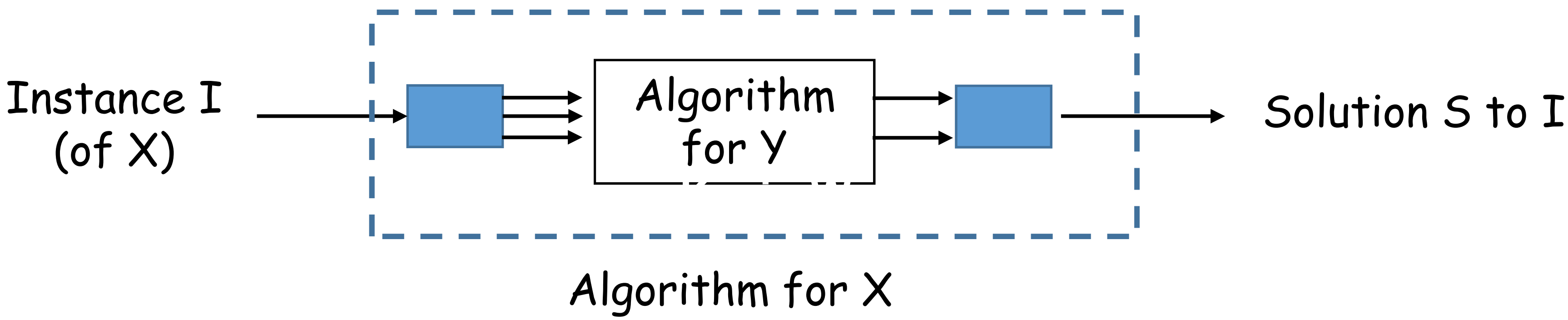
Notation: $X \leq_p Y$

The use of poly-time reductions

- Design algorithms.
  - By reducing X to Y, we can design algorithms for solving X
- Establish intractability.
  - If X$\leq_p$Y and X cannot be solved in polynomial time
  - Then Y cannot be solved in Polynomial time either
- Establish equivalence
  - If X$\leq_p$Y and Y$\leq_p$X X and Y are equivalent and X can be solved in polynomial time if and only if Y can be
  - Denoted as X $\equiv_p$ Y

Construction of a proper reduction

- The direction is critical
  - To show X$\leq_p$Y, we will assume there is an algorithm for Y and use it to solve X
  - No need to really solve Y

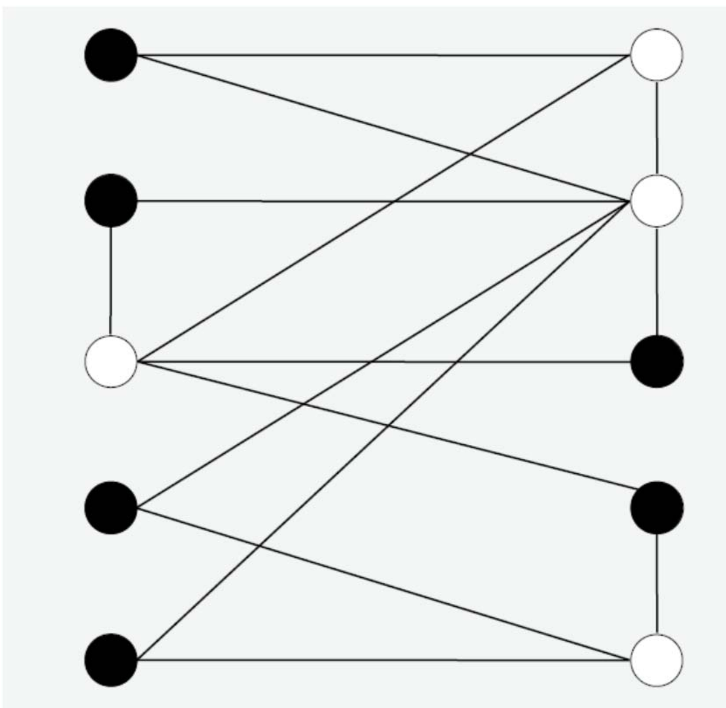


- Need to prove that the solution we construct for I is correct

- Need to make sure that the pre-processing and post-processing completes in polynomial time

Independent Set

Given a graph $G = (V, E)$, an independent set is a set $S \subseteq V$ such that for each edge in E, at most one of its end points is in S.



It is easy to find small independent set. The question is can we find big ones?

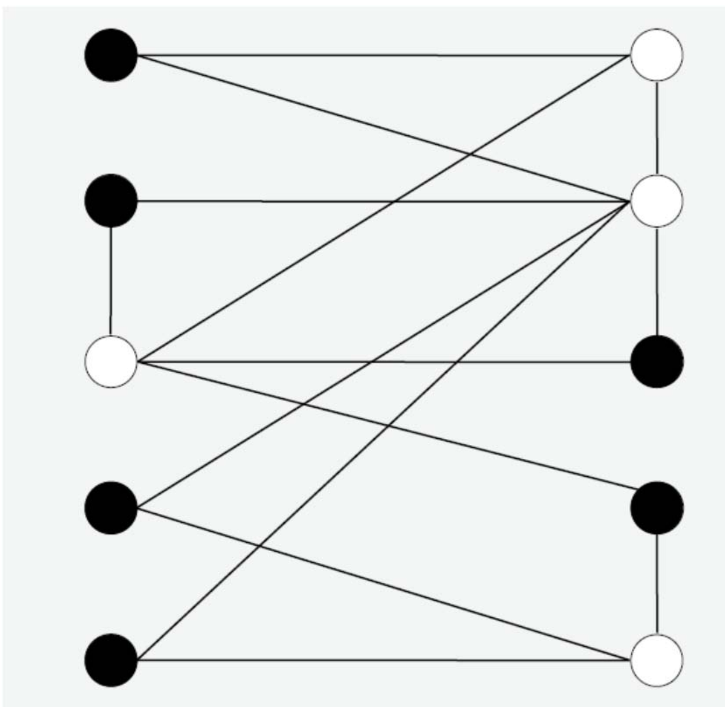
An independent set problem: Given a graph G and an integer k, does G have an independent set of size $\geq k$?

Does this graph has an independent set of size ≥ 7?

Does this graph has an independent set of size ≥ 6?

Vertex cover

Given a graph G=(V,E), a vertex cover is a set of vertices $S \subseteq V$ such that for each edge in E, at least one of its end point is in S



It is easy to find large vertex cover. The question is can we find small ones?

A vertex cover problem: Given a graph G and an integer k, does G have an vertex cover of size $\leq k$?

Is there a vertex cover of size ≤ 4 ?

Is there a vertex cover of size ≤ 3 ?

Independent Set and Vertex cover reduces to one another

Given a graph $G=(V, E)$, a set $S \subseteq V$ is an independent set if and only if $V-S$ is a vertex cover

$\Rightarrow$

Let (u, v) be an arbitrary edge in E.

Because S is an independent set, either $u \notin S$ or $v \notin S$

Thus either $u \in V-S$ or $v \in V-S$

Thus $V-S$ is a vertex cover

$\Leftarrow$

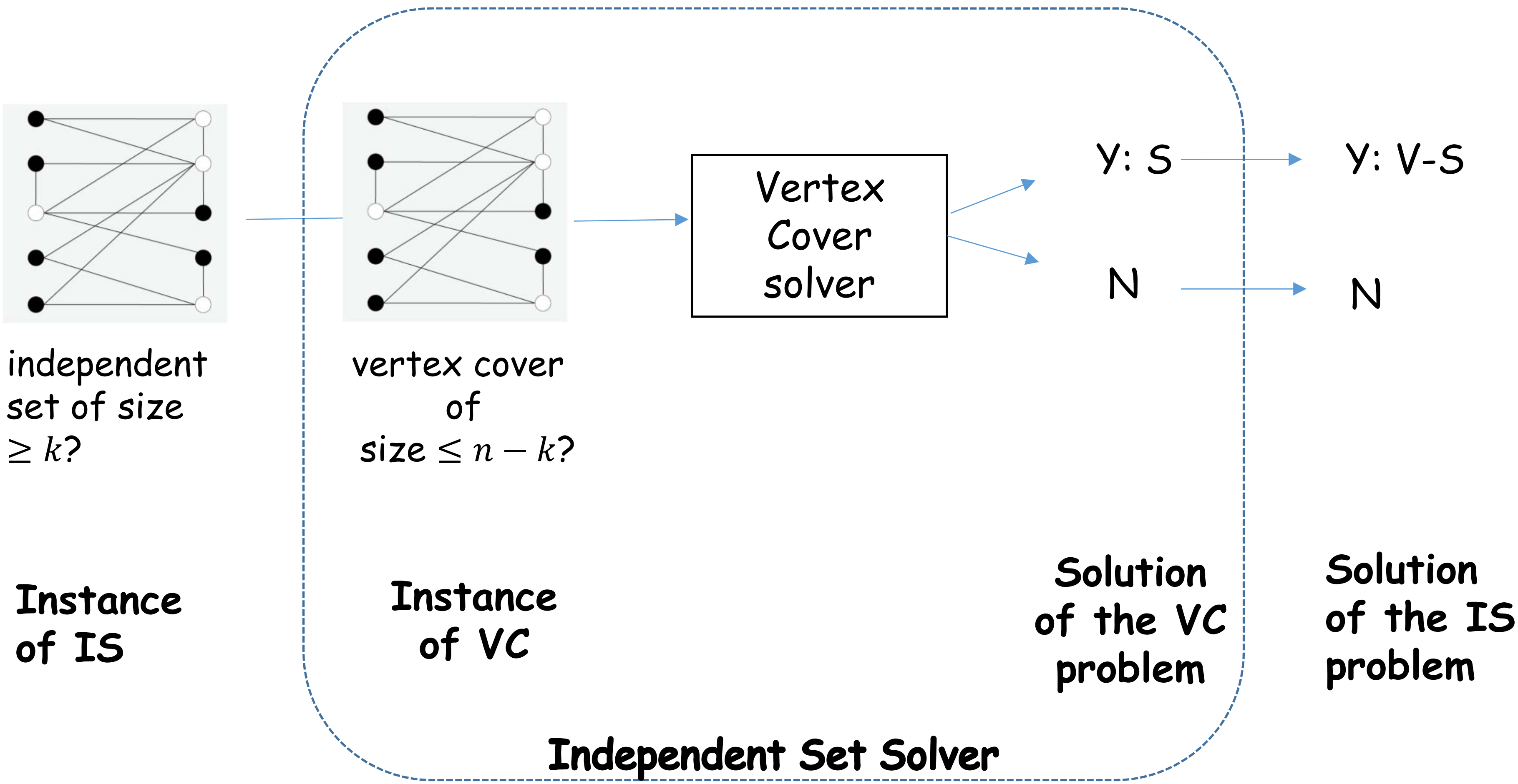
Let $e=(u, v)$ be an arbitrary edge in E.

Because $V-S$ is a vertex cover, we have either $u \in V-S$ or $v \in V-S$

As such, S can contain at most one end point of e

Thus S is an independent set

Reducing Independent Set to Vertex Cover



Set Cover

- Given a set *U* of elements, a collection *S* of subsets of *U*, and an integer *k*, are there $\leq k$ of these subsets whose union is equal to *U* ?

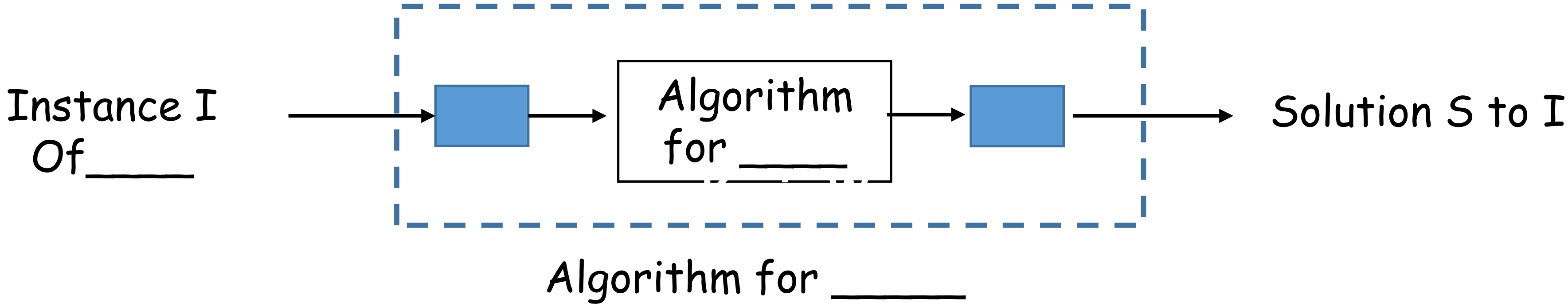
Example:

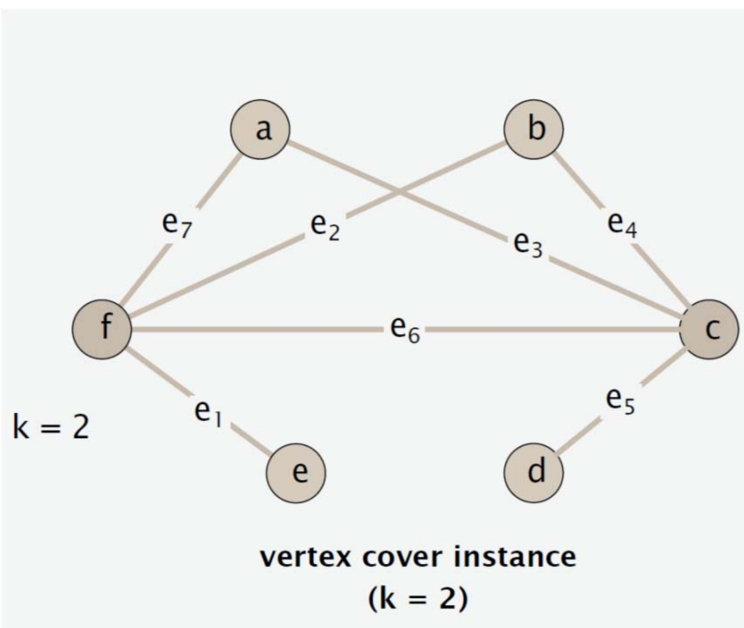
$U=\{1,2,3,4,5,6,7\}$

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

$K=2$

Vertex Cover $\leq_p$ Set Cover





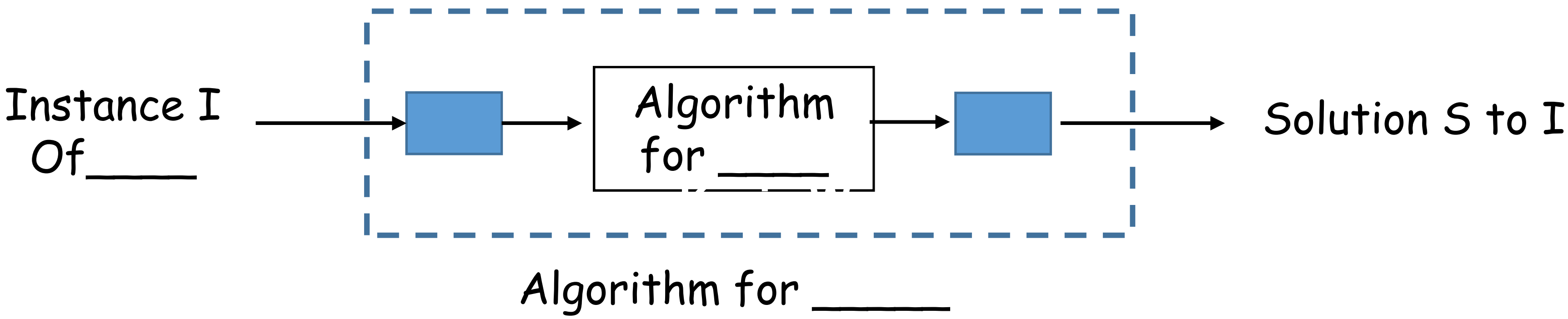
vertex cover instance
(k = 2)

Satisfiability

- Literal: A Boolean variable or its negation x_i or $\neg x_i$
- Clause: A disjunction of literals $C_j = x_1 \vee \neg x_2 \vee x_3$
- Conjunctive normal form (CNF): A Boolean formula that is a conjunction of clauses $\Phi = C_1 \wedge C_2 \wedge C_3 \wedge C_4$
- 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).

$$\Phi = (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4)$$

3-Sat $\leq_p$ Independent Set



Given an instance of 3-Sat ϕ, we need to construct an instance of independent set (G, k) such that G has an independent set of size $\geq$ k iff ϕ is satisfiable

3-Sat $\leq_p$ Independent Set

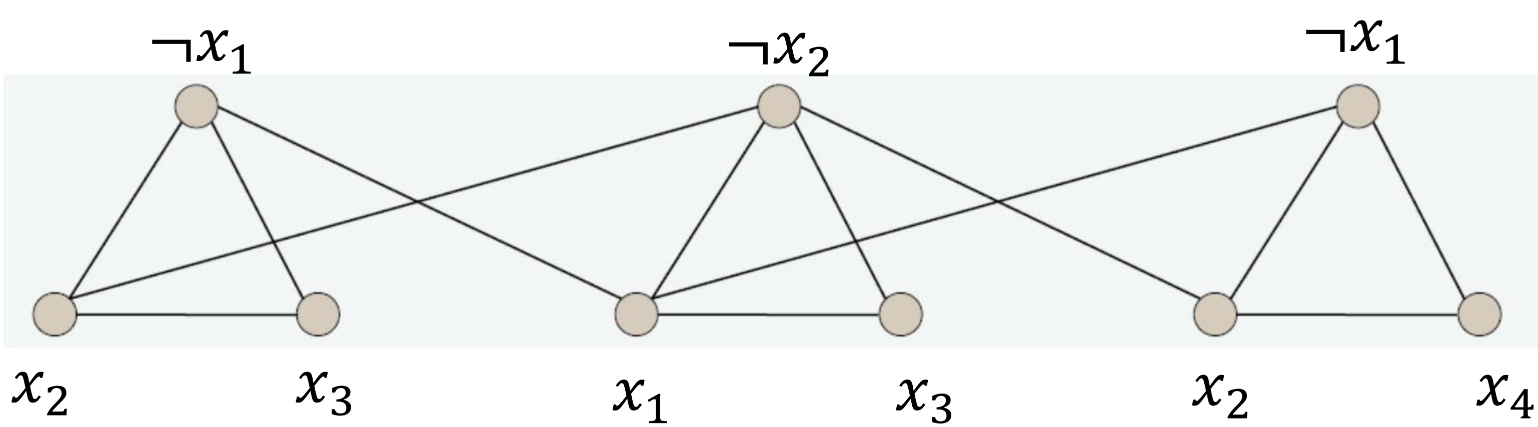
Construction from ϕ to (G,k):

For each clause, we construct 3 nodes, one for each literal

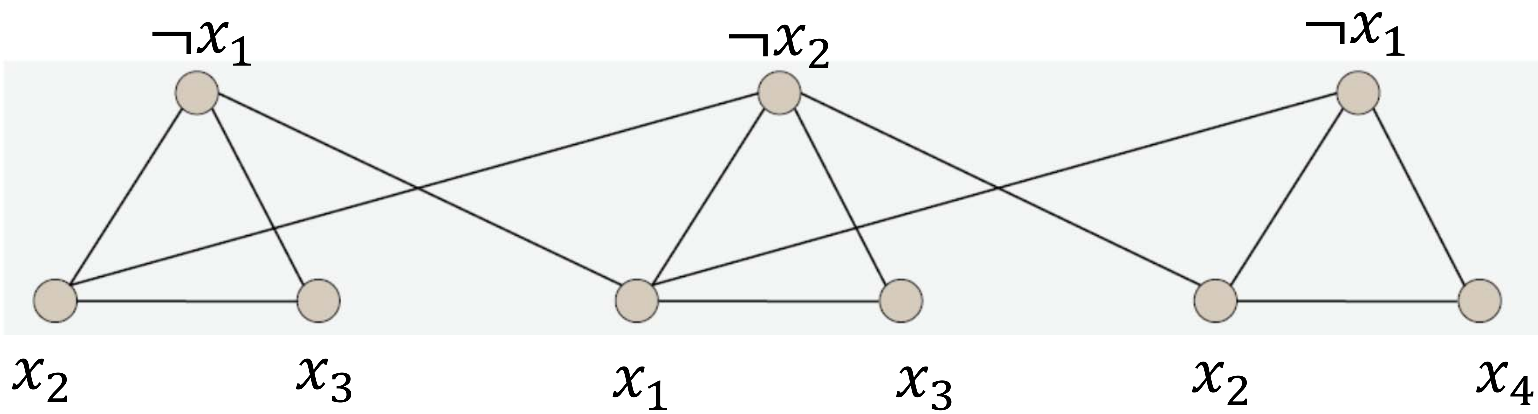
Connect the 3 nodes in a clause into a triangle

Connect literal to its negations

k = # of clauses in ϕ



$$\phi = (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4)$$

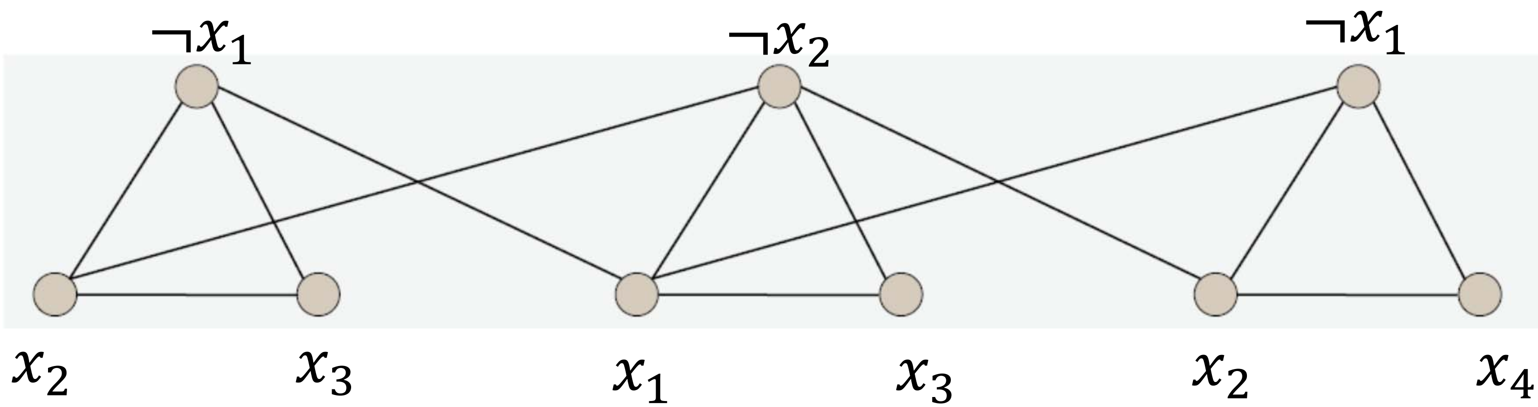


$$\phi = (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4)$$

Let k = # of clauses in ϕ

If G has an independent set S of size k $\Rightarrow$ ϕ is satisfiable

- S must contain exactly one literal from each triangle
- Set these literals to true (and remaining variables consistently)
- All clauses are then true (as each clause only need one literal to be true) --- ϕ is satisfied by this truth assignment



$$\phi = (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4)$$

If ϕ is satisfiable $\Rightarrow$ G has an independent set of size k

- Consider the satisfying truth assignment
- At least one literal must be true for each clause
- Selecting one true literal from each clause gives us an independent set of size k

Basic reduction strategies

- Simple equivalence
  - Independent Set $\equiv_p$ Vertex cover
- Special case to a more general case
  - Vertex cover $\leq$ set cover
- Encoding with gadgets
  - 3SAT $\leq$ Independent Set

- Transitivity.
  - If X $\leq_p$ Y and Y $\leq_p$ Z, then X $\leq_p$ Z

  - 3SAT $\leq_p$ Independent Set $\leq_p$ Vertex Cover $\leq_p$ Set Cover

Search vs. Optimization

- Optimization problem: find the smallest vertex cover
- Search problem: find a vertex cover of size $\leq k$
- Decision problem: is there a vertex cover of size $\leq k$?

VC-search$\leq_p$ VC-optimization

VC $\leq_p$ VC-Search

VC-Optimization $\leq_p$ VC-Search

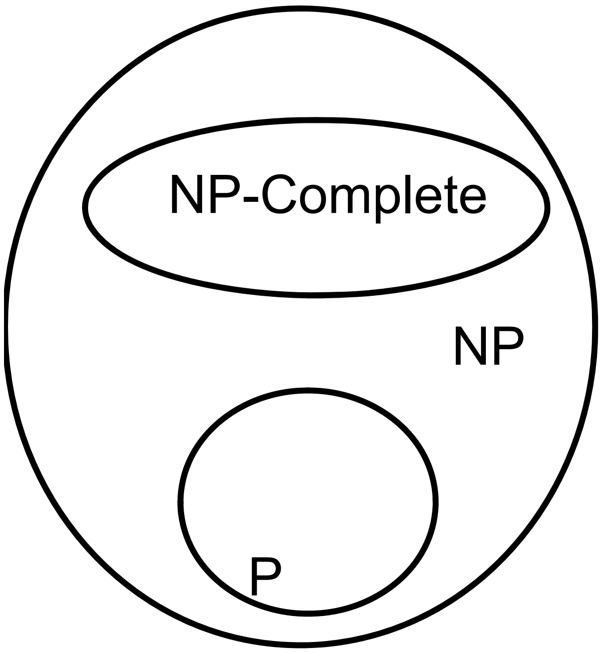
VC-Search $\leq_p$ VC

VC$\equiv_p$VC-Search $\equiv_p$ VC-Optimization

In remainder of this module, we will focus on decision problem

Theory of NP Completeness

Classifying Decision Problems



Assuming that $P \neq NP$

P: Class of problems that can be solved in polynomial time

- Corresponds with problems that can be solved efficiently in practice

NP **<u>does not stand for</u>** **"Not Polynomial"**

What is NP?

- Problems solvable in Non-deterministic Polynomial time . . .

A decision problem is in NP if it satisfies the following:

Given a problem instance I, and any proposed solution S (referred to as a "certificate") to I, we have an algorithm to verify it that runs in time that is polynomial in |I| (input size of I)

Example NP problems

- Independent set of size K
  - Certificate: an Independent Set S
  - Verification: check each edge $e \in E$ to see if both end points are in S --- $O(|E|)$
- SAT
  - Certificate: A truth assignment to all the variables
  - Verification: check if the formula is satisfied --- O(size of the formula)
- Vertex Cover of size k
  - Certificate: A vertex cover S
  - Verification: check each edge $e \in E$ to see if one of the end points is in S --- $O(|E|)$

NP-Complete

- A problem X is NP-complete if
  1. X is in NP
  2. For every Y in NP, $Y \leq_p X$
- X is among the "hardest" problems in NP

- To show a problem Z is NP-complete, we need to show
  1. Z is in NP
  2. Starting from one NP-complete problem X and show that $X \leq_p Z$