BU CS 332 – Theory of Computation

https://forms.gle/Mdxu6ahHoV8fbFn1A

Lecture 23:

- More NP-completeness

Reading:

Sipser Ch 7.4-7.5

HW 11 due Wednesday 11:59 PM

Final Monday 5/5 3-5 PM

Mark Bun

April 28, 2025

NP-completeness

"The hardest languages in NP"

Definition: A language B is NP-complete if

1) $B \in \mathrm{NP}$, and

2) B is NP-hard: Every language $A \in \mathrm{NP}$ is poly-time reducible to B, i.e., $A \leq_{\mathrm{p}} B$

Last time:

$TMSAT = \{\langle N, w, 1^t \rangle \mid$
NTM N accepts input w within t steps$\}$ is NP-complete

<u>Cook-Levin Theorem:</u>
$\{\langle \varphi \rangle \mid$ Boolean formula φ is satisfiable$\}$ is NP-complete
SAT

New NP-complete problems from old

Lemma: If $A \leq_{\mathrm{p}} B$ and $B \leq_{\mathrm{p}} C$, then $A \leq_{\mathrm{p}} C$

(poly-time reducibility is transitive)

Theorem: If $B \leq_{\mathrm{p}} C$ for some NP-hard language B, then C is also NP-hard

The usual way to prove NP-completeness:

If

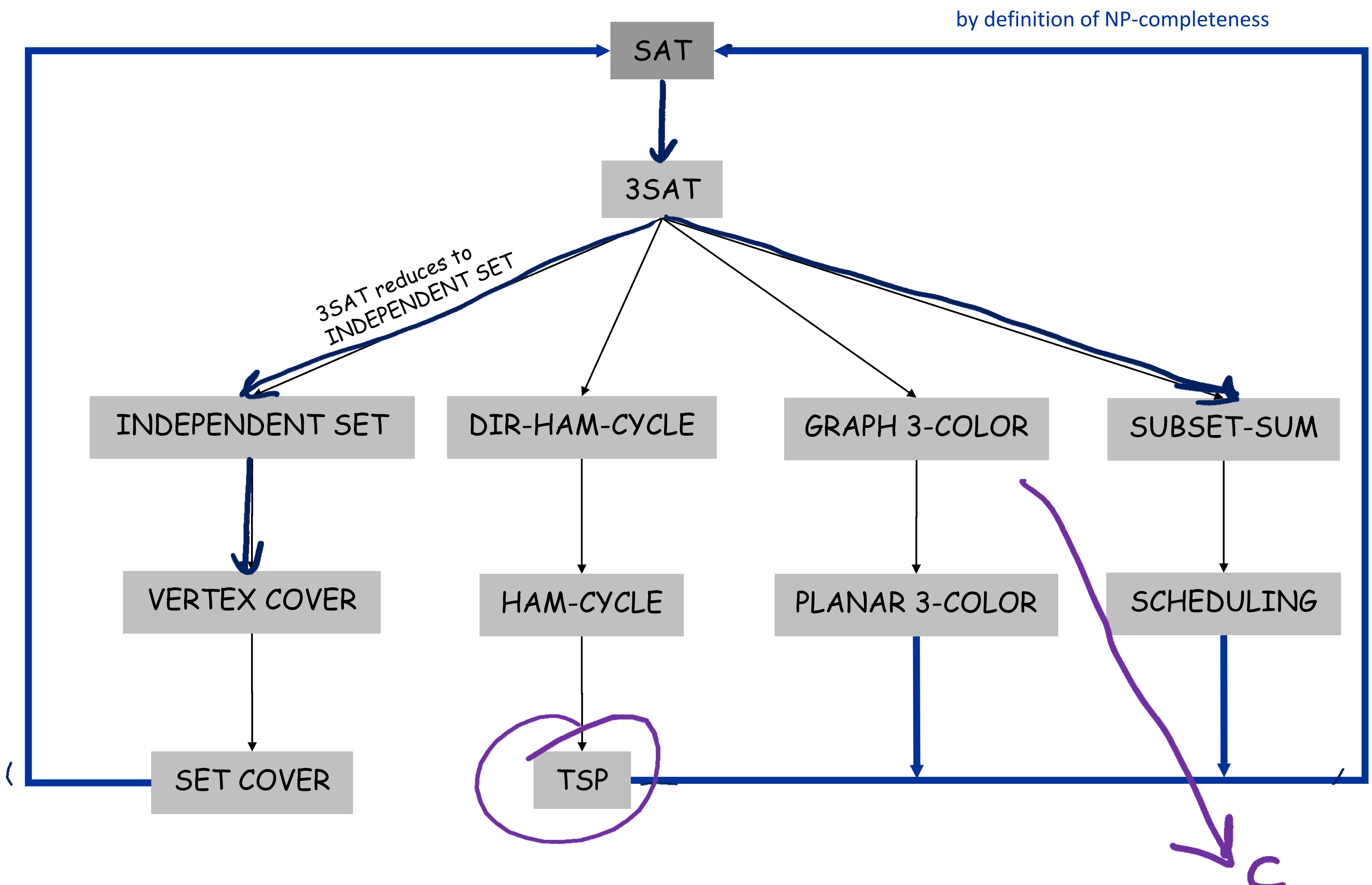
1) $C \in \mathrm{NP}$ and

2) There is an NP-complete language B (e.g., 3SAT, VERTEX-COVER, IND-SET, …) such that $B \leq_{\mathrm{p}} C$,

then C is also NP-complete.

New NP-complete problems from old

All problems below are NP-complete and hence poly-time reduce to one another!



$3SAT$ (3-CNF Satisfiability)

Definitions:

- A literal either a variable or its negation x_5 , $\overline{x_7}$
- A clause is a disjunction (OR) of literals Ex. $x_5 \vee \overline{x_7} \vee x_2$
- A 3-CNF is a conjunction (AND) of clauses where each clause contains exactly 3 literals

Ex. $C_1 \wedge C_2 \wedge \ldots \wedge C_m =$

$$\underbrace{(x_5 \vee \overline{x_7} \vee x_2)}_{C_1} \wedge \underbrace{(\overline{x_3} \vee x_4 \vee x_1)}_{C_2} \wedge \cdots \wedge \underbrace{(x_1 \vee x_1 \vee x_1)}_{C_m}$$

$$3SAT = \{\langle \varphi \rangle \mid \varphi \text{ is a satisfiable } 3-\text{CNF}\}$$

$3SAT$ is NP-complete

Theorem: $3SAT$ is NP-complete

Proof idea: 1) $3SAT$ is in NP (why?)

2) Show that $SAT \leq_{\mathrm{p}} 3SAT$

Shows $3SAT \leq_p SAT$

Your classmate suggests the following reduction from SAT to $3SAT$: "On input φ, a 3-CNF formula (an instance of $3SAT$), output φ, which is already an instance of SAT." Is this reduction correct?

a) Yes, this is a poly-time reduction from SAT to $3SAT$

b) No, because φ is not an instance of the SAT problem

c) No, the reduction does not run in poly time

d) No, this is a reduction from $3SAT$ to SAT; it goes in the wrong direction

$3SAT$ is NP-complete

Theorem: $3SAT$ is NP-complete

Proof idea: 1) $3SAT$ is in NP (why?)

2) Show that $SAT \leq_{\mathrm{p}} 3SAT$

Idea of reduction: Give a poly-time algorithm converting an arbitrary formula φ into a 3CNF ψ such that φ is satisfiable iff ψ is satisfiable

ie. $\underline{\varphi \in SAT} \iff \psi \in 3\underline{SAT}$

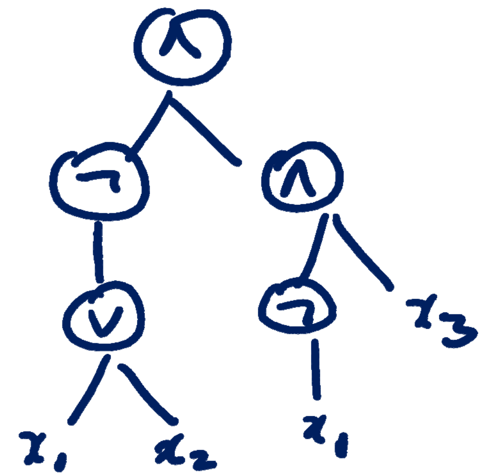
Illustration of conversion from φ to ψ

a	b	c	$a \wedge b \equiv c$
0	0	0	1
0	0	1	0
⋮		⋮	

"Proof by example" DO NOT DO THIS ON HW

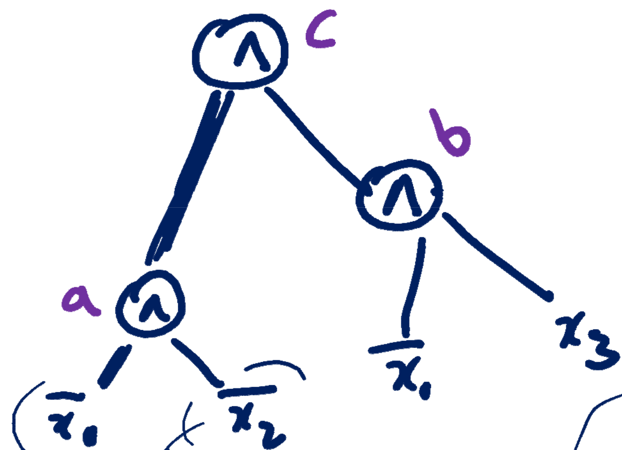
1) Let φ be a general Boolean formula

$(x_1 \vee x_2) \wedge (\bar{x}_1 \wedge x_3)$



2) "Push all negations to bottom" using de Morgan

3) Introduce auxiliary variables to capture values @ intermediate nodes



c ∨ False ∨ False
c ∨ c ∨ c

4) $\hat{\psi}(x_1, x_2, x_3, a, b, c) = (\bar{x}_1 \wedge \bar{x}_2 \equiv a) \wedge (\bar{x}_1 \wedge x_3 \equiv b) \wedge (a \wedge b \equiv c) \wedge c$

claim: $\hat{\psi}$ is satisfiable iff φ satisfiable

depends on only 3 variables

5) obtain ψ from $\hat{\psi}$ by replacing every "$a \wedge b \equiv c$" w/ an equivalent 3-CNF

Some general reduction strategies

- Reduction by simple equivalence

  Ex. $IND-SET \leq_{\mathrm{p}} VERTEX-COVER$

  $VERTEX-COVER \leq_{\mathrm{p}} IND-SET$

- Reduction from special case to general case

  Ex. $VERTEX-COVER \leq_{\mathrm{p}} SET-COVER$

  $3SAT \leq_{\mathrm{p}} SAT$

- “Gadget” reductions

  Ex. $SAT \leq_{\mathrm{p}} 3SAT$

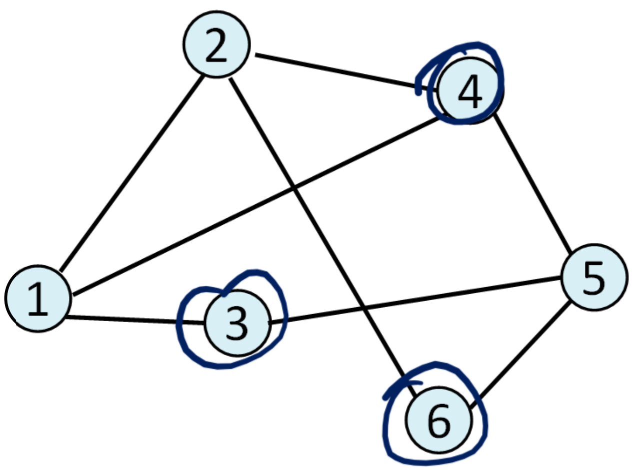
  $3SAT \leq_{\mathrm{p}} IND-SET$

Independent Set

An **independent set** in an undirected graph G is a set of vertices that includes at most one endpoint of every edge.

$\iff$ S is an independent set if $\forall$ $u, v \in S$, u, v not connected by an edge

$IND-SET = \{\langle G, k\rangle | G$ is an undirected graph containing an independent set with $\geq k$ vertices$\}$



Which of the following are independent sets in this graph?

a) $\{1\}$
b) $\{1, 5\}$
c) $\{2, 3, 6\}$
d) $\{3, 4, 6\}$

Independent Set is NP-complete

1) $IND - SET \in \mathrm{NP}$
2) Reduce $3SAT \leq_{\mathrm{p}} IND - SET$

Proof of 1) The following gives a poly-time verifier for $IND - SET$

Certificate: Vertices $v_1, \ldots, v_k$ a purported independent set of size k

Verifier:

"On input $\langle G, k; v_1, \ldots, v_k \rangle$, where G is a graph, k is a natural number,

1. Check that $v_1, \ldots v_k$ are distinct vertices in G
2. Check that there are no edges between the v_i's."

Check that $\{v_1, \ldots, v_k\}$ form an independent set of size k

Independent Set is NP-complete

1) $IND - SET \in \mathrm{NP}$
2) Reduce $3SAT \leq_{\mathrm{p}} IND - SET$ ⇒ IND-SET is NP-hard since we know 3SAT is NP-hard

Proof of 2) The following TM computes a poly-time reduction.

"On input $\langle \varphi \rangle$, where φ is a 3CNF formula,

1. Construct graph G from φ
   - G contains 3 vertices for each clause, one for each literal.
   - Connect 3 literals in a clause in a triangle.
   - Connect every literal to every appearance of its negation.
2. Output $\langle G, k \rangle$, where k is the number of clauses in φ."

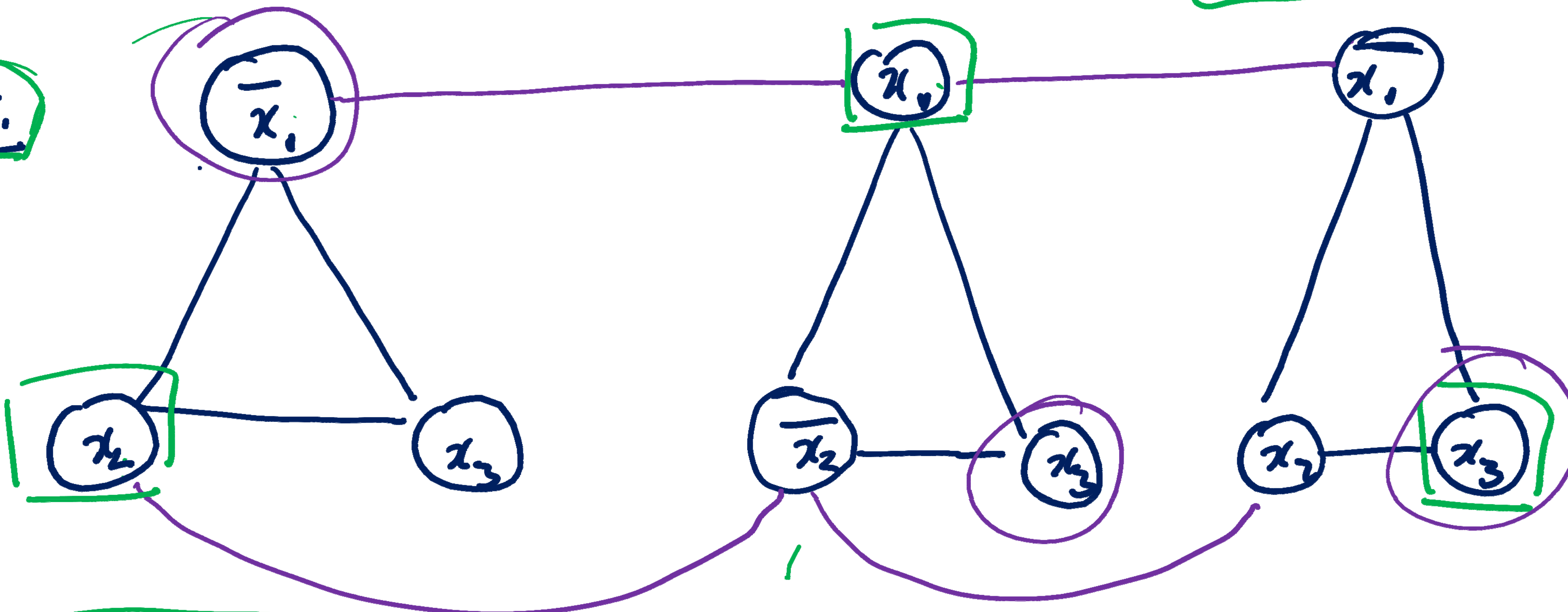
Example of the reduction

φ satisfiable $\Rightarrow$
φ has a satisfying assignment

e.g. $x_1 = 1$
$x_2 = 1$
$x_3 = 1$

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

G:



G has an independent set size 3 $\Leftrightarrow$ φ is satisfiable

$\Leftarrow$

$\overline{x_1} = 1 \Rightarrow x_1 = 0$

$x_3 = 1$ $\qquad$ $x_2 = 0$

Proof of correctness for reduction

Let k = # clauses and l = # literals in φ

Correctness: φ is satisfiable iff G has an independent set of size k

φ satisfiable $\Rightarrow$ G has an ind. set of size k

$\Longrightarrow$ Given a satisfying assignment, select one true literal from each triangle. This is an independent set of size k

$\Longleftarrow$ Let S be an independent set in G of size k

- S must contain exactly one vertex in each triangle
- Set these literals to true, and set all other variables arbitrarily
- Truth assignment is consistent and all clauses are satisfied

Runtime: $O(k + l^2)$ which is polynomial in input size

Some general reduction strategies

- Reduction by simple equivalence

  Ex. $\boxed{\begin{array}{l} IND - SET \leq_{\mathrm{p}} VERTEX - COVER \\ VERTEX - COVER \leq_{\mathrm{p}} IND - SET \end{array}}$

- Reduction from special case to general case

  Ex. $VERTEX - COVER \leq_{\mathrm{p}} SET - COVER$

  $3SAT \leq_{\mathrm{p}} SAT$

- "Gadget" reductions

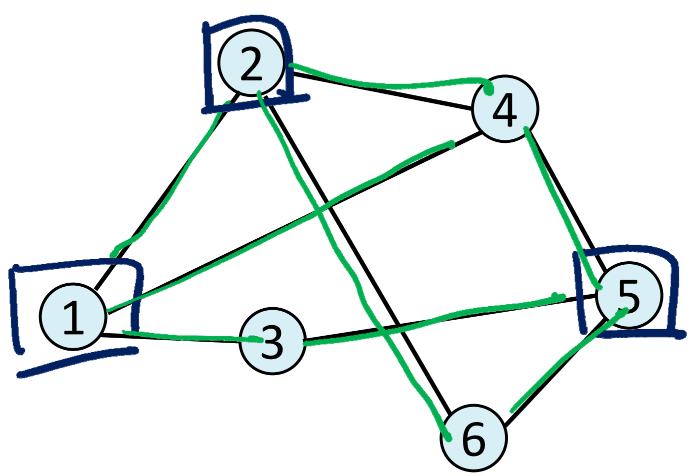
  Ex. $SAT \leq_{\mathrm{p}} 3SAT$

  $3SAT \leq_{\mathrm{p}} IND - SET$

Vertex Cover

Given an undirected graph G, a **vertex cover** in G is a subset of nodes which includes at ***least*** one endpoint of every edge.

$VERTEX-COVER$ = $\{\langle G, k\rangle \mid G$ is an undirected graph which has a vertex cover with $\leq k$ vertices$\}$



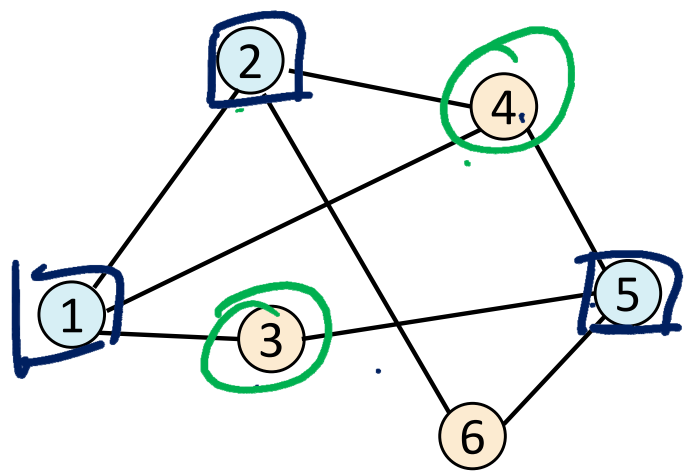
Ex: $\{1,2,5\}$ is a vertex cover

Independent Set and Vertex Cover

Claim. S is an independent set iff $V \setminus S$ is a vertex cover.

$\Longrightarrow$ Let S be any independent set.

- Consider an arbitrary edge (u, v).
- S is independent $\Longrightarrow u \notin S$ or $v \notin S \Longrightarrow u \in V \setminus S$ or $v \in V \setminus S$.
- Thus, $V \setminus S$ covers (u, v).



$\Longleftarrow$ Let $V \setminus S$ be any vertex cover.

- Consider two nodes $u \in S$ and $v \in S$.
- Then $(u, v) \notin E$ since $V \setminus S$ is a vertex cover.
- Thus, no two nodes in S are joined by an edge $\Longrightarrow S$ is an independent set.

INDEPENDENT SET reduces to VERTEX COVER

Theorem. IND-SET $\leq_p$ VERTEX-COVER.

What do we need to do to prove this theorem?

a) Construct a poly-time nondet. TM deciding IND-SET

b) Construct a poly-time deterministic TM deciding IND-SET

c) Construct a poly-time nondet. TM mapping instances of IND-SET to instances of VERTEX-COVER

d) Construct a poly-time deterministic TM mapping instances of IND-SET to instances of VERTEX-COVER

$\langle G,k\rangle \in$ IND-SET
$\iff$ $f(\langle G,k\rangle) \in$ VERTEX-COVER

e) Construct a poly-time nondet. TM mapping instances of VERTEX-COVER to instances of IND-SET

f) Construct a poly-time deterministic TM mapping instances of VERTEX-COVER to instances of IND-SET

INDEPENDENT SET reduces to VERTEX COVER

Theorem. IND-SET $\leq_{\mathrm{p}}$ VERTEX-COVER.

Proof. The following TM computes the reduction.

"On input $\langle G, k\rangle$, where G is an undirected graph and k is an integer,

1. Output $\langle G, n-k\rangle$, where n is the number of nodes in G."

Correctness:

- G has an independent set of size at least k iff it has a vertex cover of size at most $n-k$.

$\Rightarrow$ $\exists$ S an indep. set of size k

$\Rightarrow$ $V \setminus S$ is vertex cover of size $n-k$

G has a vertex cover of size $n-k$

$\Rightarrow$ G has an indep. set of size k

Runtime:

- Reduction runs in linear time.

at least on a multi-tape TM

VERTEX COVER reduces to INDEPENDENT SET

Theorem. VERTEX-COVER $\leq_{\mathrm{p}}$ IND-SET

Proof. The following TM computes the reduction.

"On input $\langle G, k\rangle$, where G is an undirected graph and k is an integer,

1. Output $\langle G, n-k\rangle$, where n is the number of nodes in G."

Correctness:

- G has a vertex cover of size at most k iff it has an independent set of size at least $n-k$.

Runtime:

- Reduction runs in linear time.