BU CS 332 – Theory of Computation

https://forms.gle/z4Ydo8MQfj5dWYiZ7

Lecture 23:

- More NP-completeness

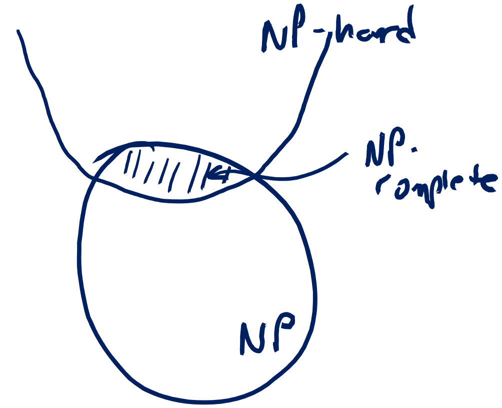
Reading:

Sipser Ch 7.4-7.5

Mark Bun

December 2, 2021

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: There exists an NP-complete language

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

Cook-Levin Theorem and NP-Complete Problems

Cook-Levin Theorem

Theorem: SAT (Boolean satisfiability) is NP-complete

"Proof": Already know $SAT \in \mathrm{NP}$. (Much) harder direction: Need to show every problem in NP reduces to SAT

Stephen A. Cook (1971)

Leonid Levin (1973)

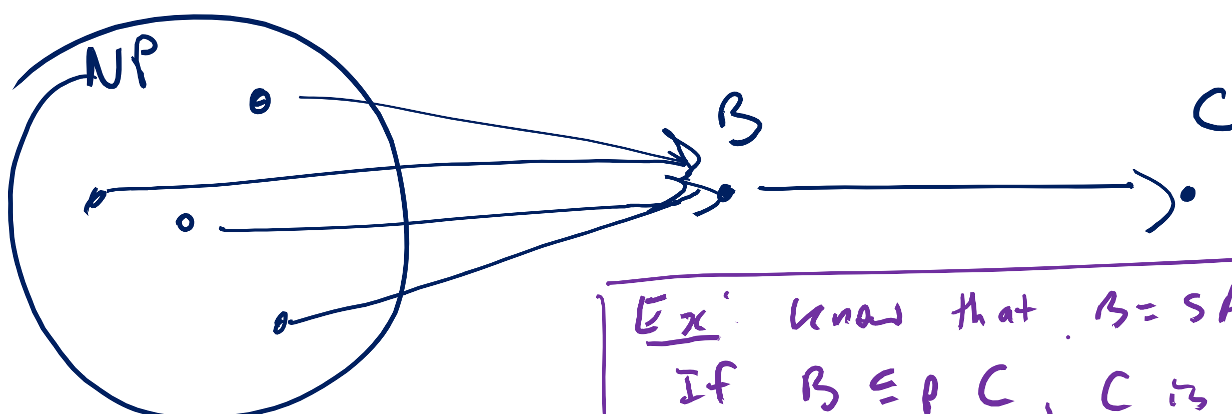
New NP-complete problems from old

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

(poly-time reducibility is <u>transitive</u>)

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

Corollary: If $C \in \mathrm{NP}$ and $B \leq_{\mathrm{p}} C$ for some NP-complete language B, then C is also NP-complete



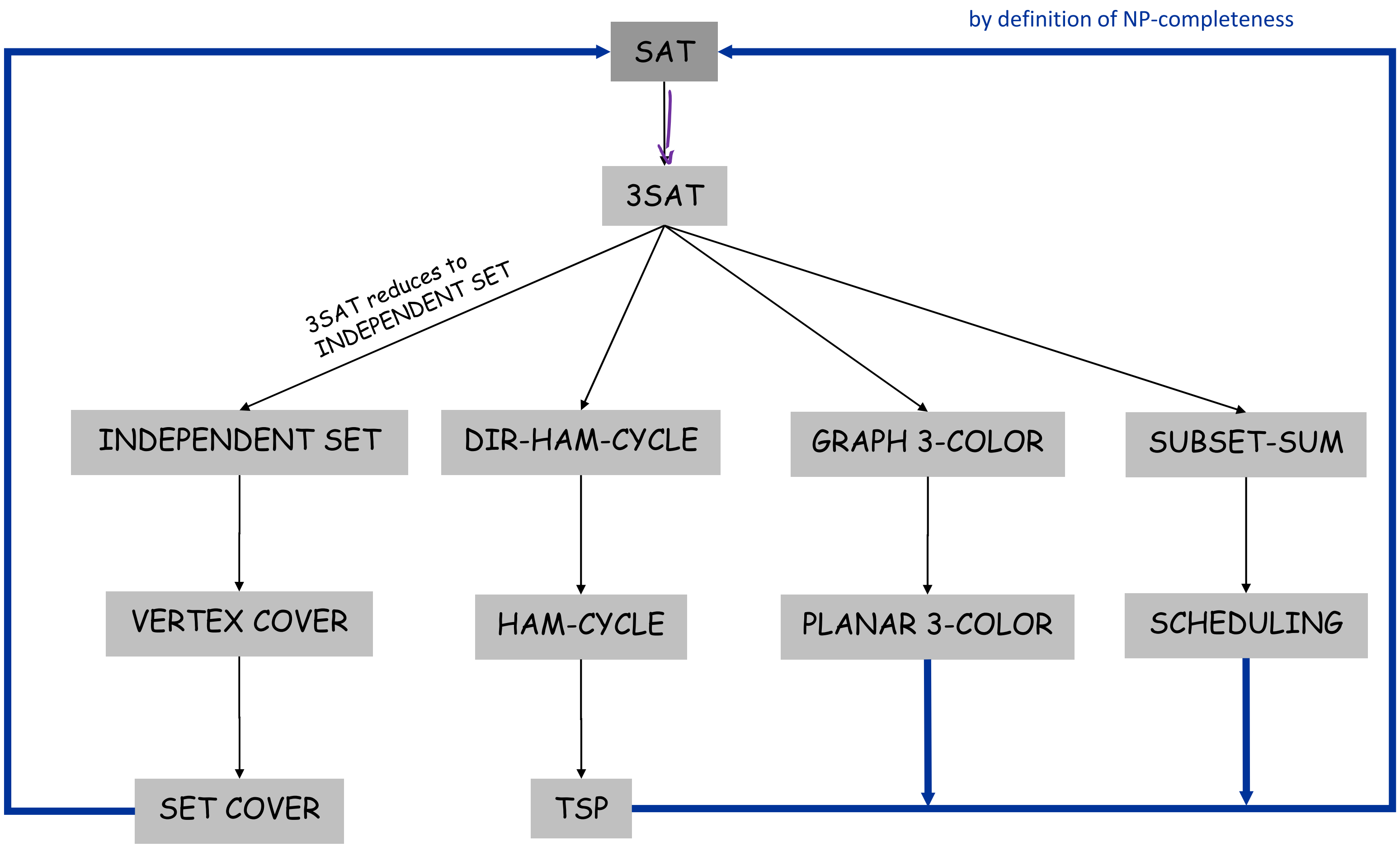
Ex: know that $B = SAT$ is NP-complete

If $B \leq_p C$, C is in NP

Then C is 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 of its negation $x_5\ ,\ \overline{x_7}$
- A clause is a disjunction (OR) of literals **Ex.** $x_5 \lor \overline{x_7} \lor x_2$
- A 3-CNF is a conjunction (AND) of clauses where each clause contains exactly 3 literals

clauses

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

$$(\underbrace{x_5 \lor \overline{x_7} \lor x_2}_{C_1}) \land (\overline{x_3} \lor x_4 \lor x_1) \land \cdots \land (\underbrace{x_1 \lor x_1 \lor x_1}_{C_m})$$

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

$3SAT$ is NP-complete

Theorem: $3SAT$ is NP-complete

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

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

Want: Instance of SAT $\rightarrow$ Instance 3SAT

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? Instance of 3SAT $\xrightarrow{f}$ Instance of SAT

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

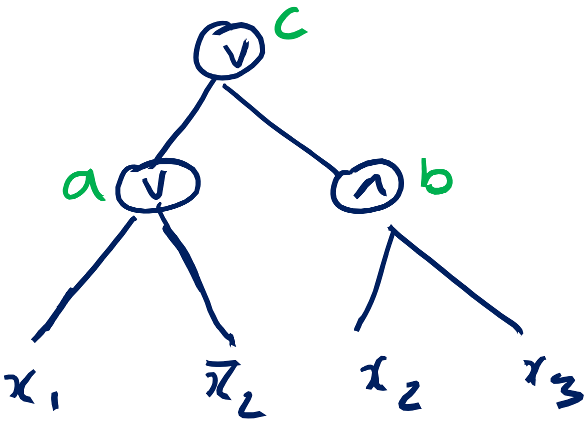
Proof idea: 1) $3SAT$ is in NP (why?)

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

Illustration of conversion from φ to ψ

φ:



means $(x_1 \vee \bar{x}_2) \vee (x_2 \wedge x_3)$

Assume all negations at bottom

$c = (c \vee c \vee c)$

$\hat{\psi}(x_1, x_2, x_3, a, b, c)$

$= c \wedge \underline{(c = a \vee b)} \wedge$

$\underline{(a = x_1 \vee \bar{x}_2)} \wedge \underline{(b = x_2 \wedge x_3)}$

pseudoclauses

Thm: Every $f: \{0,1\}^3 \to \{0,1\}$ can be written as a 3CNF

i.e.

$f(x,y,z) = (\ell_1 \vee \ell_2 \vee \ell_3) \wedge \ldots$
$\wedge (\ell_{21} \vee \ell_{23} \vee \ell_{24})$

where each ℓ_i is a literal over x, y, z

Obtain ψ from $\hat{\psi}$ by applying Thm to each "pseudoclause"

Some general reduction strategies

- Reduction by simple equivalence

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

  $VERTEX - COVER \leq_p IND - SET$

- Reduction from special case to general case

  Ex. $VERTEX - COVER \leq_p SET - COVER$

  $3SAT \leq_p SAT$

  φ is a 3CNF (instance of 3SAT)
  then φ is also an instance of SAT

- "Gadget" reductions

  Ex. $SAT \leq_p 3SAT$

  $\boxed{3SAT \leq_p IND - SET}$

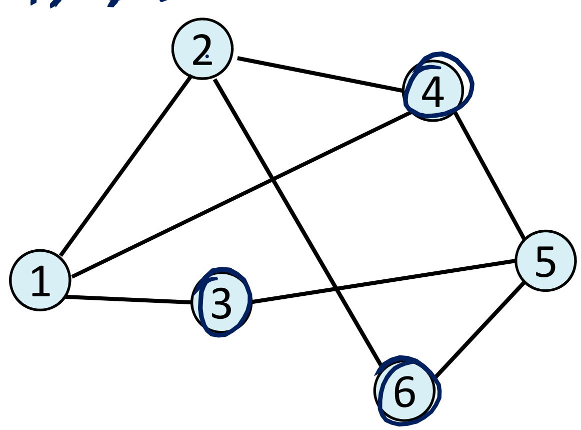
Independent Set

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

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

$\langle G, 3\rangle \in IND\text{-}SET$
$\langle G, 4\rangle \notin IND\text{-}SET$

$G =$



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 $v_1, \ldots, v_k$

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 $v_1, \ldots, v_k$ is actually an independent set of size k

Independent Set is NP-complete

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

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 each of its negations.
2. Output $\langle G, k \rangle$, where k is the number of clauses in φ."

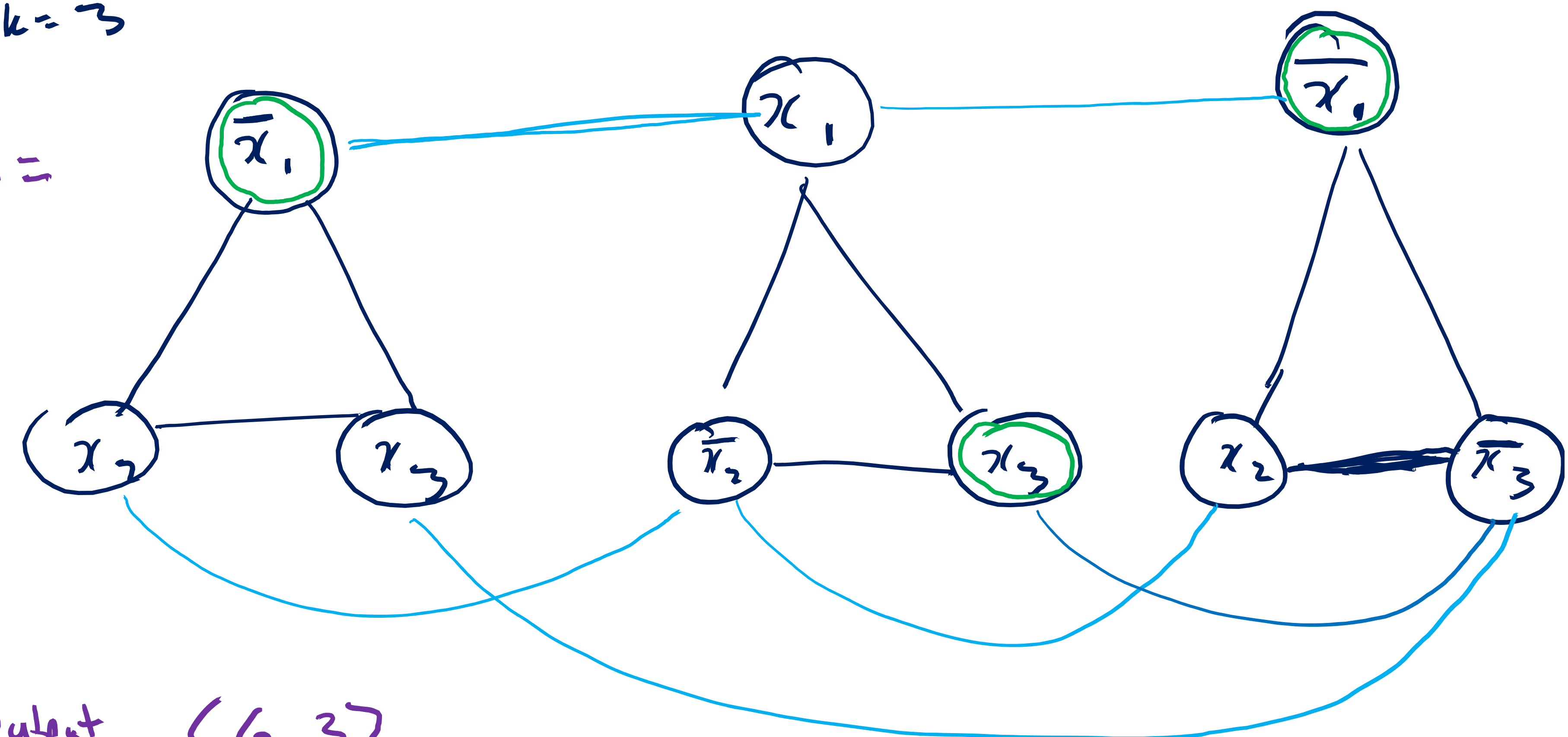
Example of the reduction

Ex: $x_1 = 0$, $x_2 = 0$, $x_3 = 1$

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

$k = 3$

$G =$



Output $\langle G, 3 \rangle$

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

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

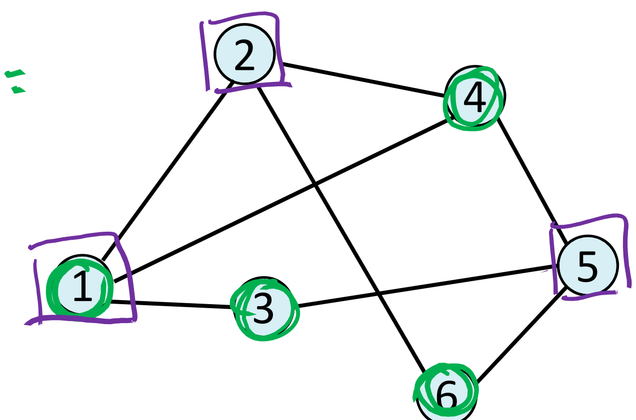
  $3SAT \leq_{\mathrm{p}} IND-SET$

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$\}$

$G =$



$\langle G, 4\rangle \in$ VERTEX-COVER

$\langle G, 3\rangle \in$ VERTEX-COVER

$\langle G, 2\rangle \notin$ VERTEX-COVER

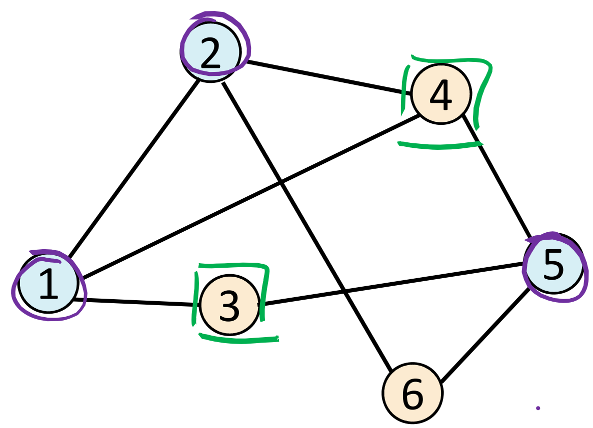
(G does not have a vert. cover of size ≤ 2)

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.

Consequence: $\exists$ indep. set in G of size k $\Longleftrightarrow$ $\exists$ vertex cover of G of size $|V| - k$

INDEPENDENT SET reduces to VERTEX COVER

Theorem. IND-SET $\leq_{\mathrm{p}}$ 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
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 $n-k$. Hence $\langle G,k\rangle \in$ IND-SET $\iff$ $\langle G, n-k\rangle \in$ VERTEX-COVER

Runtime:

- Reduction runs in linear time.

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.