

BU CS 332 – Theory of Computation

<https://forms.gle/spqMNF3e4ddh2szh7>



Lecture 22:

- More NP-completeness

Reading:

Sipser Ch 7.4-7.5

Mark Bun & Alexander Poremba

April 23, 2026

Final exam
May 7th

Last time: Understanding P vs. NP

Most believe $P \neq NP$, but we are very far from proving it

Question : How can studying specific computational problems help us get a handle on resolving P vs. NP?

Idea: Identify the “hardest” problems in NP

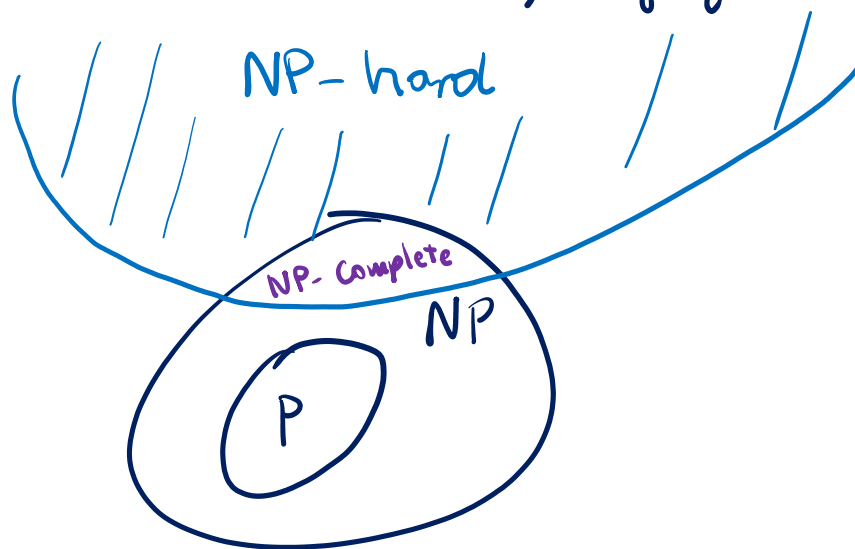
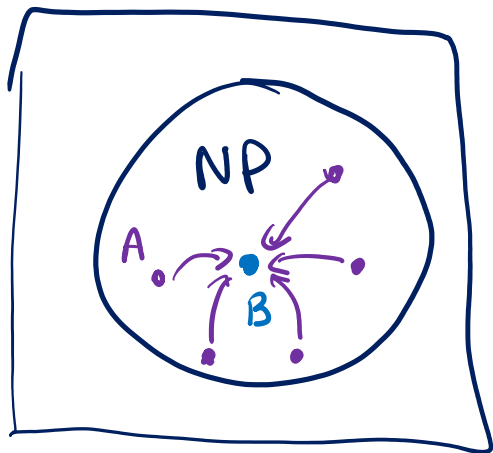
Languages $L \in NP$ such that $L \in P$ iff $P = NP$

NP-completeness

Definition: A language B is NP-complete if

- 1) $B \in \text{NP}$, and
- 2) B is NP-hard: **Every** language $A \in \text{NP}$ is poly-time reducible to B , i.e., $A \leq_p B$

" B is at least as hard as any language A in NP "



Implications of NP-completeness

Theorem: Suppose B is NP-complete.

Then $B \in P$ iff $P = NP$

Proof:

" $\Leftarrow$ ": Assume $P = NP$. We know B is NP-complete
 $\Rightarrow \underline{B \in NP} = P \Rightarrow B \in P$.

" $\Rightarrow$ ": Suppose B is NP-complete and $B \in P$.

Thus, for any language $A \in NP$:

$A \leq_P B$ (B is NP-hard)

But $B \in P \Rightarrow A \in P$!

Therefore: $NP \leq P \stackrel{P \subseteq NP}{\Rightarrow} P = NP$

Implications of NP-completeness

Theorem: Suppose B is NP-complete.

Then $B \in P$ iff $P = NP$

Consequences of B being NP-complete:

- 1) If you want to prove $P = NP$, you just have to prove $B \in P$
- 2) If you want to prove $P \neq NP$, a good candidate is to try to show that $B \notin P$
- 3) If you believe $P \neq NP$, then you also believe $B \notin P$

Cook-Levin Theorem and NP-Complete Problems

Do NP-complete problems exist?

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

Proof sketch: 1) $TMSAT \in NP$:

Non-deterministically guess t transitions of NTM N on input w ,
then check if it leads N to accept w .

2) $TMSAT$ is NP-hard: Let $L \in NP$ be decided by NTM
 N running in time $T(n)$. The following poly-time TM shows
 $L \leq_p TMSAT$:

“On input w (an instance of L):

Output $\langle N, w, 1^{T(|w|)} \rangle$.”

Correctness

• $w \in L \Rightarrow N$ accepts w within
 $T(|w|)$ steps

$\Rightarrow \langle N, w, 1^{T(|w|)} \rangle \in TMSAT$ ✓

• $w \notin L \Rightarrow N$ does not accept w

$\Rightarrow \langle N, w, 1^{T(|w|)} \rangle \notin TMSAT$ ✓

Cook-Levin Theorem (Sipser Ch. 7.4)

Theorem: *SAT* (Boolean satisfiability) is NP-complete

“Proof”: Already know *SAT* $\in$ 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 $A \leq_p B$ and $B \leq_p C$, then $A \leq_p C$

(poly-time reducibility is transitive)

Theorem: If $B \leq_p C$ for some NP-hard language B , then C is also NP-hard

New NP-complete problems from old

Lemma: If $A \leq_p B$ and $B \leq_p C$, then $A \leq_p C$

(poly-time reducibility is transitive)

Theorem: If $B \leq_p C$ for some NP-hard language B , then C is also NP-hard

The usual way to prove NP-completeness:

If

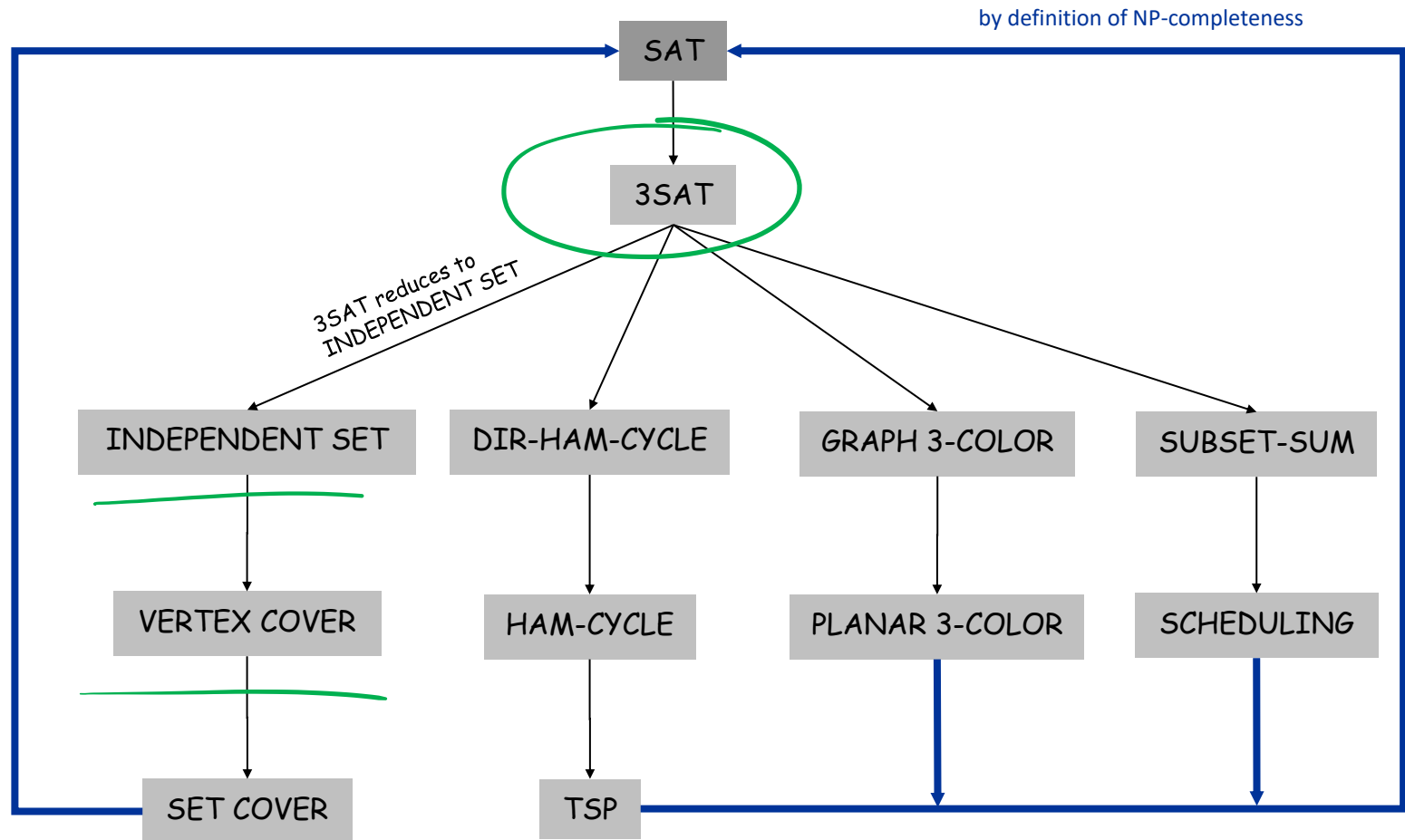
1) $C \in \text{NP}$ and

2) There is an NP-complete language B (e.g., 3SAT, VERTEX-COVER, IND-SET, ...) such that $B \leq_p C$,

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. $C_1 \wedge C_2 \wedge \dots \wedge C_m =$

$$\underbrace{(x_5 \vee \overline{x_7} \vee x_2)} \wedge \underbrace{(\overline{x_3} \vee x_4 \vee x_1)} \wedge \dots \wedge \underbrace{(x_1 \vee x_1 \vee x_1)}$$

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

3SAT is NP-complete

Theorem: 3SAT is NP-complete

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

2) Show that $SAT \leq_p 3SAT$



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?

- 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 $SAT \leq_p 3SAT$

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

$$\boxed{\varphi \in SAT \iff \psi \in 3SAT}$$

Sketch: For each clause in φ , split it into a sequence of 3-literal clauses, and introduce fresh variables to link the pieces together.

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. $VERTEX - COVER \leq_p SET - COVER$

$3SAT \leq_p SAT$

- “Gadget” reductions

Ex. $\underline{SAT} \leq_p 3SAT$

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

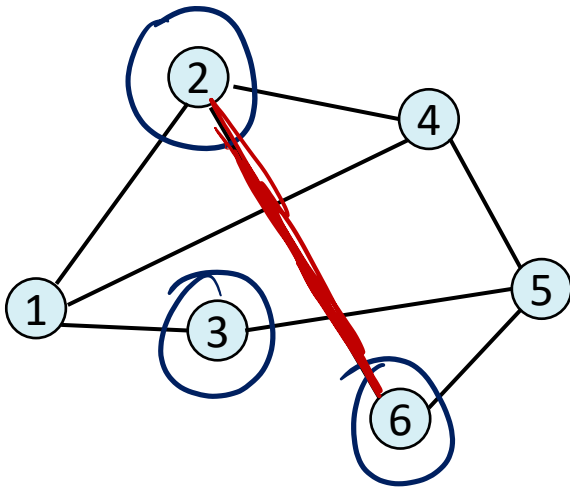
(transitivity of “ $\leq_p$ ”)

Independent Set

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

$\Leftrightarrow S$ is an independent set if $\forall (u,v) \in E, u$ and v share no edge.

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



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) $IND - SET \in NP$
- 2) Reduce $3SAT \leq_p IND - SET$

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

Certificate: Vertices $v_1, \dots, v_k$

Verifier:

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

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

$\Rightarrow IND-SET \in NP$



Independent Set is NP-complete

- 1) $IND - SET \in NP$
- 2) Reduce $3SAT \leq_p IND - SET$

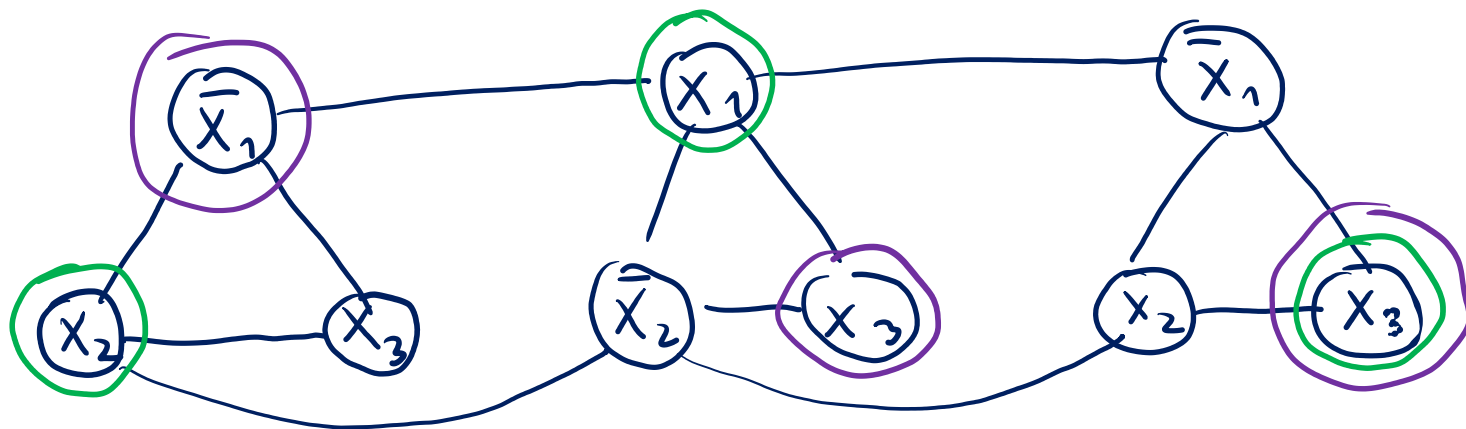
Proof of 2) The following TM computes a poly-time reduction.

“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 every appearance of its negation.
2. Output $\langle G, k \rangle$, where k is the number of clauses in φ .”

Example of the reduction

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



Case 1 : φ satisfiable $\Rightarrow$ each clause can be satisfied
e.g. $x_1 = x_2 = x_3 = 1 \Rightarrow$ independent set of size 3.

Case 2 : S is an independent set of size 3.
 $\Rightarrow S$ contains one node in each triangle.
We can set these nodes to TRUE $\Rightarrow$ satisfiable assignment.

Proof of correctness for reduction

Let $k = \# \text{ clauses}$ and $l = \# \text{ literals in } \varphi$

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

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

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

- Reduction from special case to general case

Ex. $VERTEX - COVER \leq_p SET - COVER$
 $3SAT \leq_p SAT$

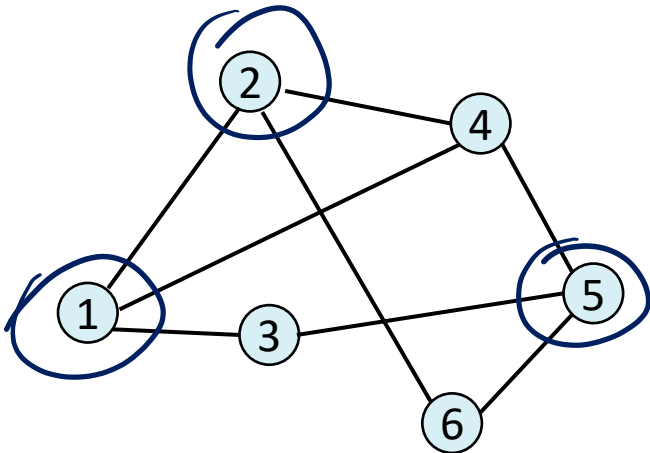
- “Gadget” reductions

Ex. $SAT \leq_p 3SAT$
 $3SAT \leq_p IND - SET$

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.

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



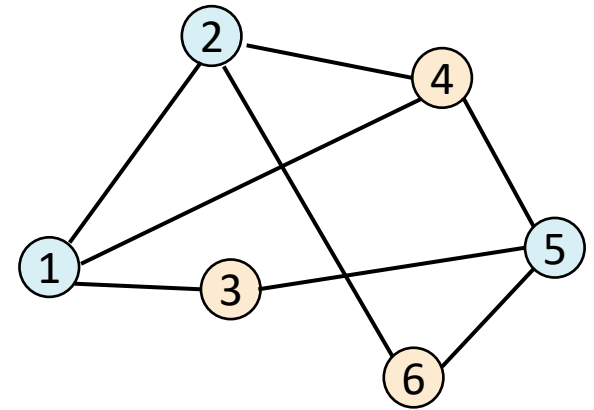
Ex: $\{1, 2, 5\}$ is a vertex cover.

Independent Set and Vertex Cover

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

$\Rightarrow$ Let S be any independent set.

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



$\Leftarrow$ 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 $\Rightarrow S$ is an independent set.

INDEPENDENT SET reduces to VERTEX COVER

Theorem. $\text{IND-SET} \leq_p \text{VERTEX-COVER}$.

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. $\text{IND-SET} \leq_p \text{VERTEX-COVER}$.

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 $n - k$.

Runtime:

- Reduction runs in linear time.

VERTEX COVER reduces to INDEPENDENT SET

Theorem. $\text{VERTEX-COVER} \leq_p \text{IND-SET}$

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.