CS 7800: Advanced Algorithms

Lecture 11: Intractability I

- Polynomial-time Reductions
- Class NP and NP-complete problems

Instructor: Jonathan Ullman
Lecturer: Lydia Zakynthinou

10-18-22

Polynomial-time Reductions

So far: Designed "efficient" algorithms for several problems.

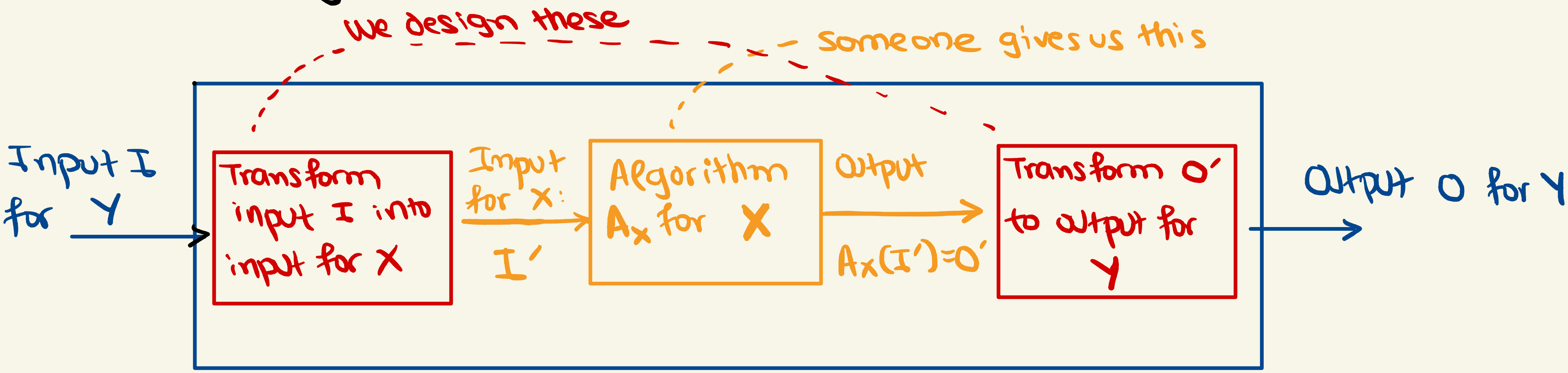
↳ run in polynomial time
with respect to their input size

Some solutions were using known algorithms as a "black-box".

Today: Reductions

- Way to solve a problem given algorithm for another problem
- Help us compare the relative difficulty between problems

Polynomial-time Reductions



Algorithm A_Y for Y using A_X as a "black box"

More generally:

<u>Def.</u> Y is polynomial-time reducible to X if Y can be solved using a polynomial number of standard computational steps, plus a polynomial number of calls to a "black box" that solves problem X.

We write: $Y \leq_p X$.

Suppose $Y \leq_p X$. If <u>X can</u> be solved in polynomial time, then <u>Y can</u> be solved in polynomial time. ($X \in P \Rightarrow Y \in P$)

$\Updownarrow$

Suppose $Y \leq_p X$. If <u>Y cannot</u> be solved in polynomial time, then <u>X cannot</u> be solved in polynomial time. ($Y \notin P \Rightarrow X \notin P$)

VERTEX COVER $\equiv_p$ INDEPENDENT SET ($\equiv_p$ is $\leq_p$ and $\geq_p$)

Input: Graph $G(V,E)$, integer K

Output: YES iff G contains vertex cover $S \subseteq V$ of size $\leq K$.

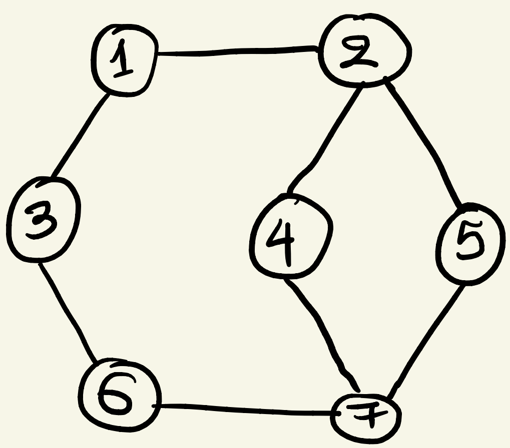
All edges have at least one end in S.

Input: Graph $G(V,E)$, integer K

Output: YES iff G contains independent set $S \subseteq V$ of size $\geq K$.

No edges between nodes in S.

G



Vertex cover of size ≤ 3? 2, 7, 3

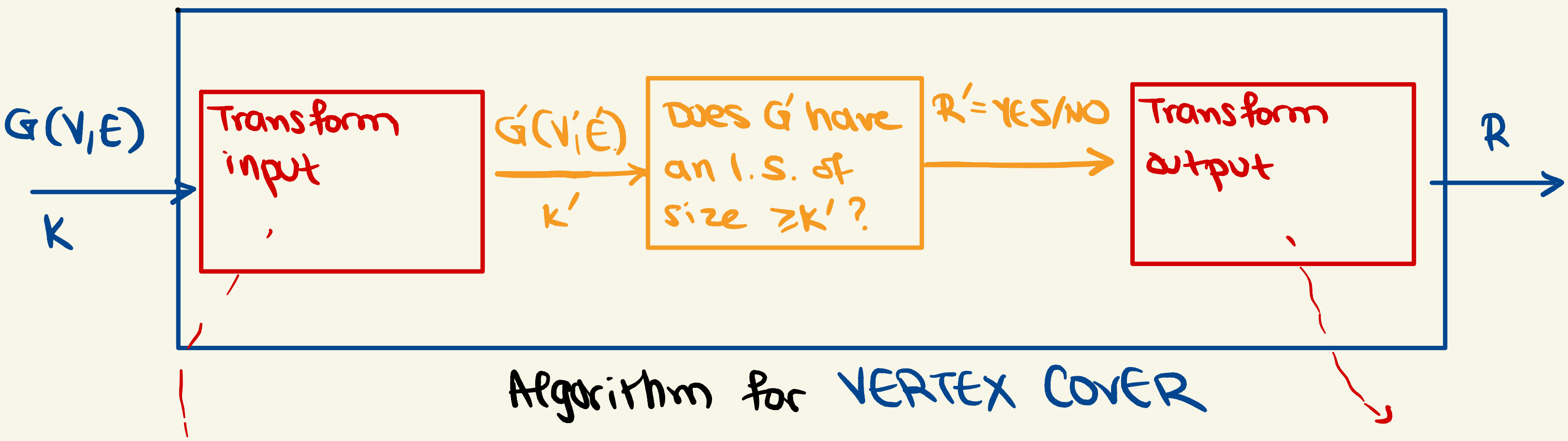
Independent set of size ≥ 4? 1, 6, 4, 5

Decision version of problem.

VERTEX COVER $\equiv_p$ INDEPENDENT SET

<u>Obs</u>: S is an independent set iff $V \setminus S$ is a vertex cover.

$\Rightarrow$ VERTEX COVER $\leq_p$ INDEPENDENT SET



$G'(V',E') = G(V,E)$

$k' = n - k$

$R = R' = \text{YES/NO}$

Polynomial time ✓

Correctness: ✓

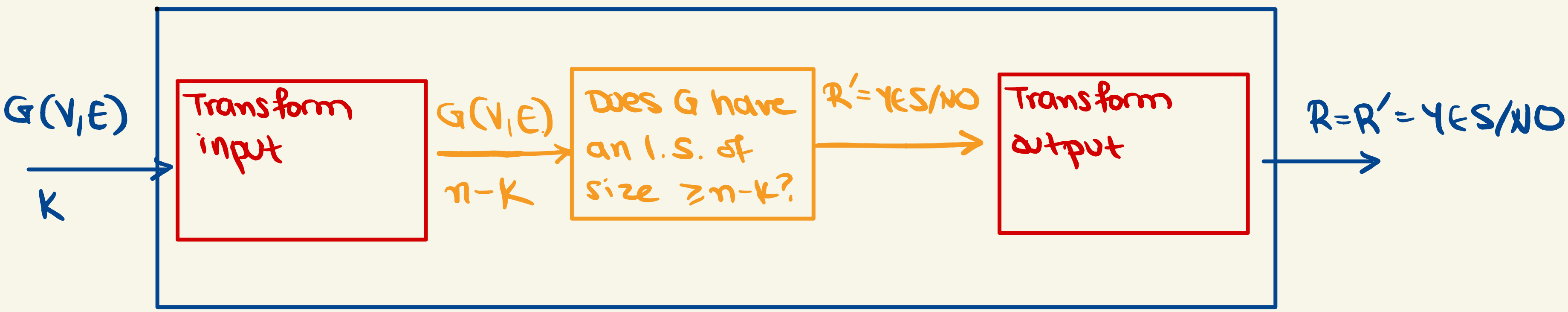
If $\exists$ VC of size $\leq k$, then $\exists$ IS of size $\geq k'$.

If $\exists$ IS of size $\geq k'$, then $\exists$ VC of size $\leq k$.

VERTEX COVER $\equiv_p$ INDEPENDENT SET

<u>Obs.</u>: S is an independent set iff $V \setminus S$ is a vertex cover.

$\Rightarrow$ VERTEX COVER $\leq_p$ INDEPENDENT SET



Algorithm for VERTEX COVER

$\Rightarrow$ Similarly, INDEPENDENT SET $\leq_p$ VERTEX COVER ∎

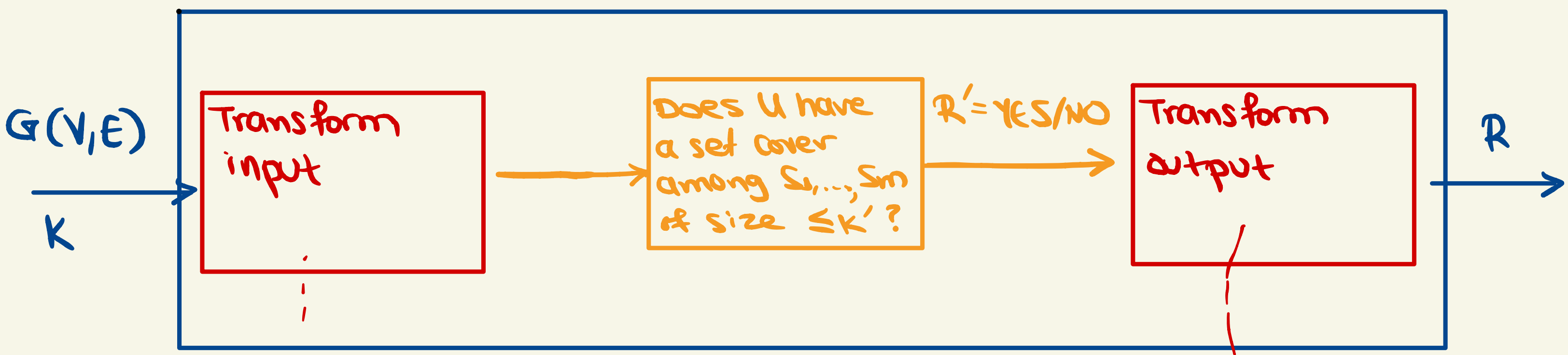
Another equivalence with similar strategy: CLIQUE $\equiv_p$ INDEPENDENT SET

<u>Obs.</u>: S is an independent set in G iff S is a clique in its complement $\bar{G}$.

VERTEX COVER $\le_p$ SET COVER

Input: Elements U, $|U|=n$. Collection of sets $S_1,\dots,S_m \subseteq U$. Integer K.

Output: YES iff $\exists$ collection of at most K of these sets whose union equals U.



Algorithm for VERTEX COVER

$R = R' = $ YES/NO

$U = E$

$S_1,\dots,S_m$: $S_i = \{e \in E : e=\{i,j\} \text{ or } e=\{j,i\}\}$

$K' = K$

Polynomial time ✓

Correctness ✓

VERTEX COVER $\le_p$ 0-1 INTEGER LINEAR PROGRAMMING

Input: $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, $v \in \mathbb{R}$.

Output: YES iff $\exists\, x \in \{0,1\}^n$ s.t. $c^T x \le v$ and $Ax \ge b$.

$G(V,E)$

k

Does there $\exists\, x_1, \ldots, x_n$ s.t.

$$\sum_{i \in V} x_i \le k,$$

$$x_i + x_j \ge 1 \quad \forall e = \{i,j\} \in E,$$

$$x_i \in \{0,1\} \quad \forall i \in V \quad ?$$

$$
\begin{tikzcd}
 & \text{SET COVER} \\
\text{VERTEX COVER} \arrow[ur, "\le_p"] \arrow[r, "\le_p"] \arrow[d, leftrightarrow, "\equiv_p"] & \text{INTEGER LINEAR PROGRAMMING} \\
\text{INDEPENDENT SET} \arrow[d, leftrightarrow, "\equiv_p"] & \\
\text{CLIQUE} &
\end{tikzcd}
$$

Prop.:

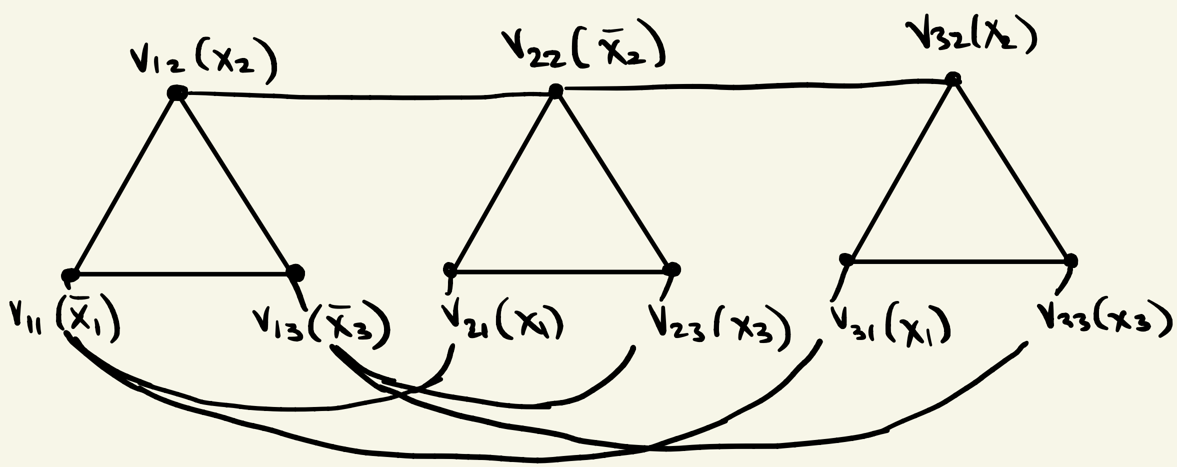
Reductions are transitive: If $Y \le_p X$ and $X \le_p Z$, then $Y \le_p Z$.

3-SAT $\le_p$ INDEPENDENT SET

Input: Set X of n Boolean variables $x_1, \dots, x_n$. Clauses $C_1, \dots, C_k$, each of length 3.
Output: YES iff $\exists$ truth assignment $v: X \to \{0,1\}$ such that all clauses evaluate to 1.

e.g.: $\varphi = \underbrace{(\bar{x}_1 \vee x_2 \vee \bar{x}_3)}_{C_1} \wedge \underbrace{(x_1 \vee \bar{x}_2 \vee x_3)}_{C_2} \wedge \underbrace{(x_1 \vee x_2 \vee x_3)}_{C_3}$

Given formula φ over X with clauses $C_1, \dots, C_k$, transform into input to Ind. Set $G(V,E), k$.



✓ Polynomial time
✓ Correctness
φ is satisfiable
$\iff \exists$ independent set of size $\geq k$.

- Each clause C_i is a triangle: "clause gadget"
- Add extra edges to indicate conflicts between x_j and $\bar{x}_j$.

$$
\begin{tikzcd}
 & & & \text{SET COVER} \\
 & \text{VERTEX COVER} \arrow[d, leftrightarrow, "\equiv_p"] \arrow[urr, "\le_p"] \arrow[rr, "\le_p"] & & \text{INTEGER LINEAR PROGRAMMING} \\
 & \text{INDEPENDENT SET} \arrow[d, leftrightarrow, "\equiv_p"] & & \\
 & \text{CLIQUE} & & \\
\text{3-SAT} \arrow[uur, "\le_p"] & & &
\end{tikzcd}
$$

The Class NP

Non-deterministic Polynomial time

Def. NP is the class of problems for which $\exists$ an efficient certifier.

Def. Algorithm B is an efficient certifier for problem X if:

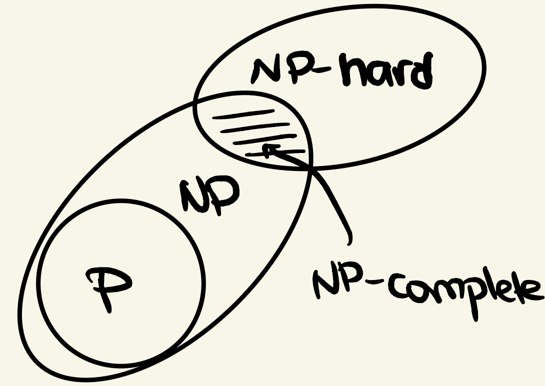
1. It is a polynomial time algorithm that takes input s and certificate t.
2. $\exists$ polynomial p so that $s \in X$ (YES instance) iff $\exists t$ with length $|t| \le p(|s|)$ for which $B(s,t) = \text{YES}$.

Hard to think of problems not in NP.

- $NP \ni$ 3-SAT, Vertex Cover, Independent Set....
- $P \subseteq NP$ ← easy to check solution   easy to find solution

But we don't know $\boxed{P \stackrel{?}{=} NP}$

The Class NP



Def. Y is NP-hard iff $\forall X \in NP \quad X \leq_p Y$

$\Rightarrow$ If Y is NP-hard and $Y \in P$ then $P = NP$.

Def. Y is NP-complete iff Y is NP-hard and $Y \in NP$.

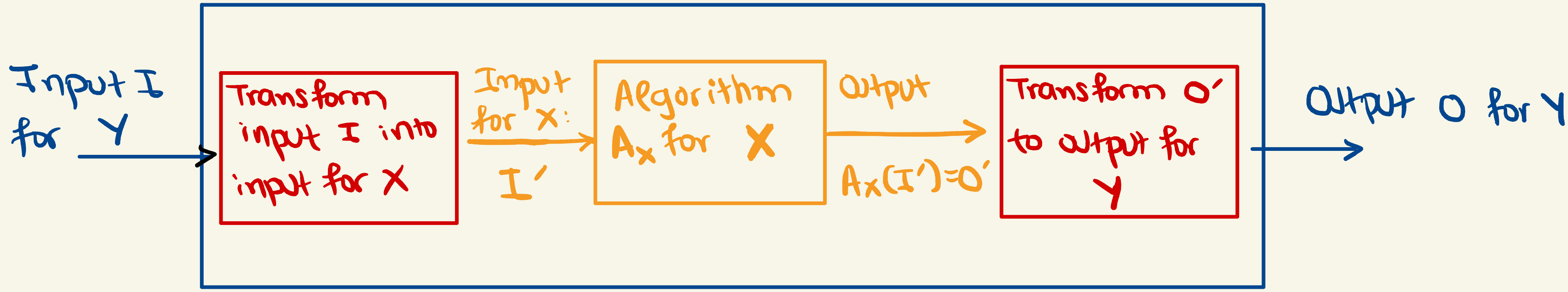
Theorem (Cook '71, Levin '73): CIRCUIT-SAT is NP-complete.
Also, CIRCUIT-SAT $\leq_p$ 3-SAT.

Since 3-SAT $\in$ NP, 3-SAT is NP-complete.

$$
\begin{tikzcd}
 & & \text{SET COVER} \\
 & \text{VERTEX COVER} \arrow[ur, "\le_p"] \arrow[r, "\le_p"'] \arrow[d, leftrightarrow, "\equiv_p"] & \text{INTEGER LINEAR PROGRAMMING} \\
 & \text{INDEPENDENT SET} \arrow[d, leftrightarrow, "\equiv_p"] & \\
 & \text{CLIQUE} & \\
\text{Any problem } Y \in NP \arrow[r, "\le_p"] & \text{3-SAT} \arrow[uu, bend left, "\le_p"] &
\end{tikzcd}
$$

All these are in NP $\Rightarrow$ All are NP-complete.

Strategy to prove that X is NP-complete



(1) Prove $X \in NP$.

(2) Find problem Y that is known to be NP-complete, and prove $Y \leq_p X$*:

- Consider arbitrary input I to problem Y.
- Construct a poly-time transformation of input I to a (special) instance I' of X and prove correctness:
- If I is a YES instance for Y $\Rightarrow$ I' is a YES instance for X.
- If I' is a YES instance for X $\Rightarrow$ I is a YES instance for Y.

"packing", "covering", "sequencing", "partitioning", "numerical".

* Karp reduction. More general reductions are Cook reductions.