CS 7800: Advanced Algorithms

Class 14: Reductions and Intractability

- Finish Image Segmentation
- NP-Completeness

Jonathan Ullman

October 21, 2025

Image Segmentation

- Separate image into foreground and background
- We have some idea of:
  - whether pixel i is in the foreground or background
  - whether pair (i,j) are likely to go together

Image Segmentation

- Input:
  - a directed graph $G = (V, E)$
    - V = "pixels", E = "pairs"
  - likelihoods $a_i, b_i \geq 0$ for every $i \in V$
  - separation penalty $p_{ij} \geq 0$ for every $(i, j) \in E$
- Output:
  - a partition of V into (A, B) that maximizes

$$q(A, B) = \sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{\substack{(i,j) \in E \\ \text{cut by } A,B}} p_{ij}$$

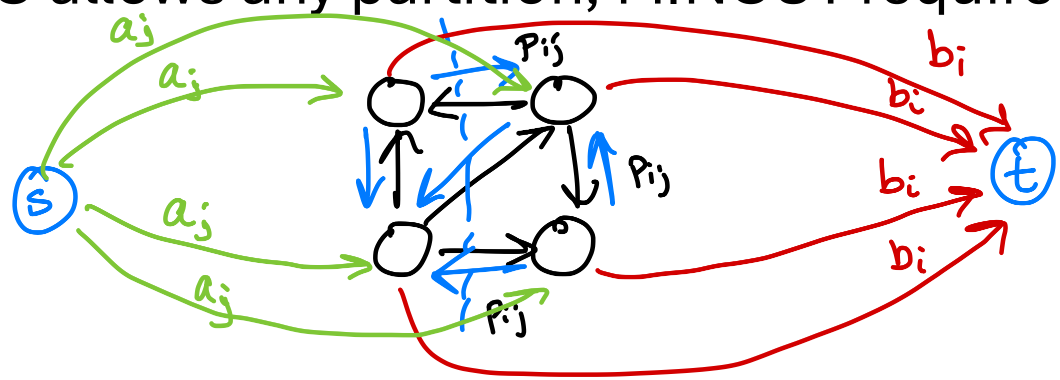
Reduction to MinCut

- Differences between SEG and MINCUT:
  - SEG asks us to maximize, MINCUT asks us to minimize

How to include these?

$$\max_{A,B} \sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{\substack{(i,j) \in E \\ \text{btw } A \text{ and } B}} p_{ij} \quad \Longleftrightarrow \quad \min_{A,B} \sum_{i \in A} b_i + \sum_{j \in B} a_j + \sum_{\substack{(i,j) \in E \\ \text{btw } A \text{ and } B}} p_{ij}$$

  - SEG allows any partition, MINCUT requires $s \in A$, $t \in B$



  - SEG counts any cut edge, MINCUT counts $A \to B$ edges

Reduction to MinCut

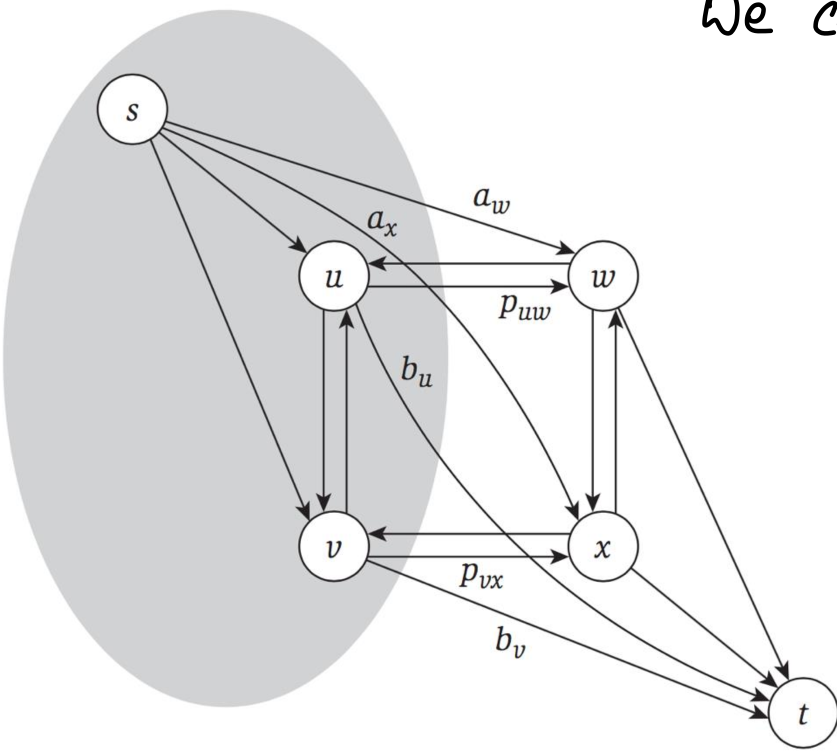
- How can we set up a flow network where the cost of the segmentation is the capacity of a cut

$$\boxed{\min_{A,B} \sum_{i \in A} b_i + \sum_{j \in B} a_j + \sum_{\substack{(i,j) \in E \\ \text{btw } A \text{ and } B}} p_{ij}}$$

Step 1: Transform the Input

Given $G=(V,E,\{a_i\},\{b_i\},\{p_{ij}\})$ for SEG $\Longrightarrow$ $G'=(V,E,s,t,\{c_e\})$ for MINCUT

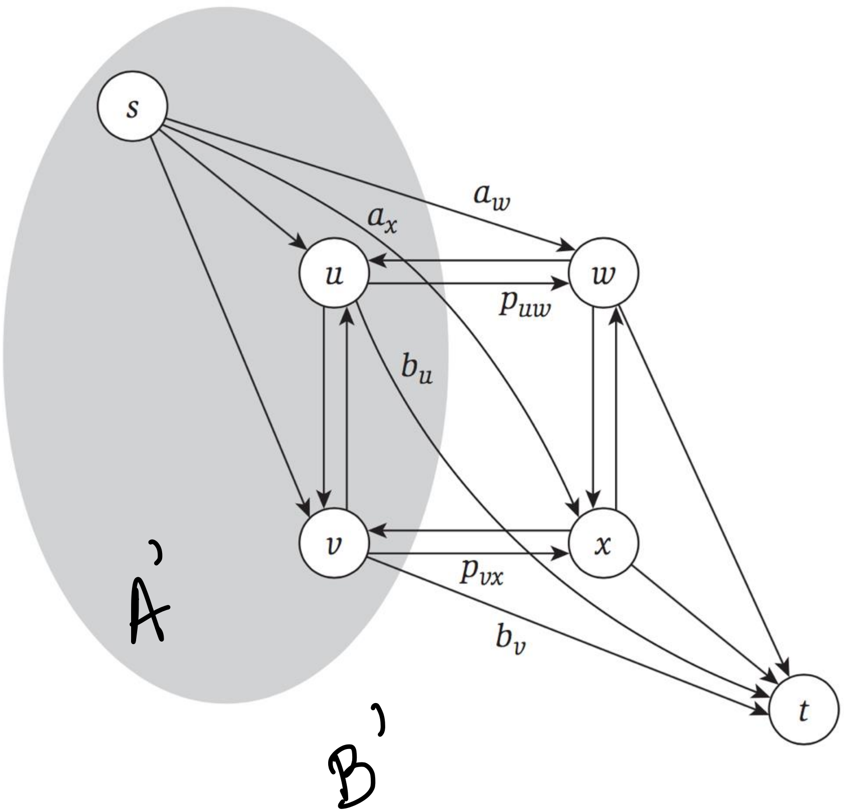
We can run this step in $O(m)$ time



Step 2: Receive the Output

Receive A', B'

for MINCUT



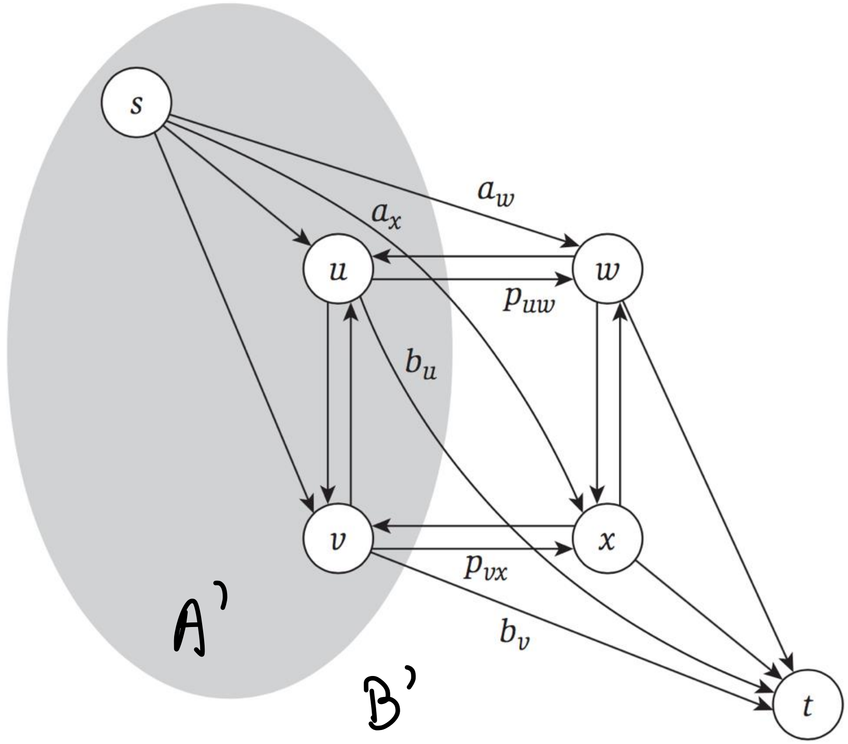
Step 3: Transform the Output

A', B' for MINCUT $\Longrightarrow$ A, B for SEG

$A = A' \setminus \{s\}$

$B = B' \setminus \{t\}$

Can be done in $O(n)$ time



Summary

Solving minimum s-t cut in a graph with.
$n+2$ nodes and $2m+2n$ edges in time T

⬇

Solving image segmentation in a graph with n
nodes and m edges in time $T+O(m)$

- Can solve image segmentation in $O(mn)$ time

Flow Applications Summary

- Network flow algorithms are powerful
  - Can use them to solve many optimization problems
  - Improvements for maxflow implies lots of new algorithms

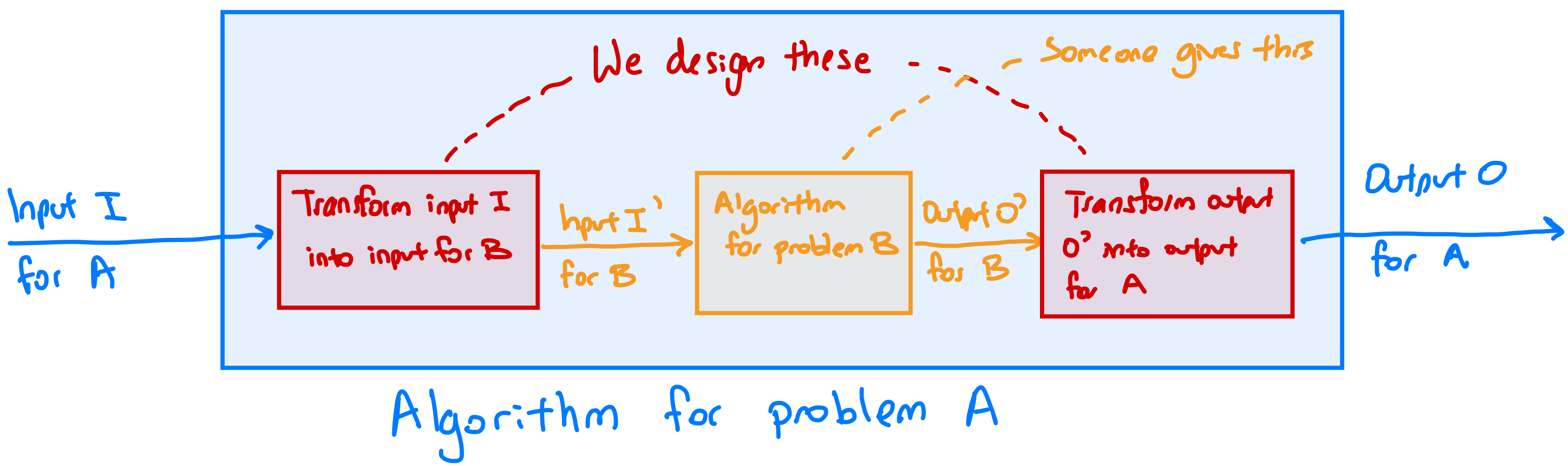
- Many natural applications
  - Bipartite matching
  - Image segmentation
  - Airline scheduling
  - Fair division
  - Auction design
  - ...

- Maxflow-Mincut duality (often) implies interesting duality theorems for these problems

Reductions

Reduction: Problem A reduces to Problem B if there is a polynomial-time algorithm that solves A using any efficient algorithm that solves B.

- Denoted $A \leq_P B$ (i.e. "A is at most as hard as B")
- **View 1:** If B can be solved efficiently then so can A
- **View 2:** If A can't be solved efficiently then neither can B

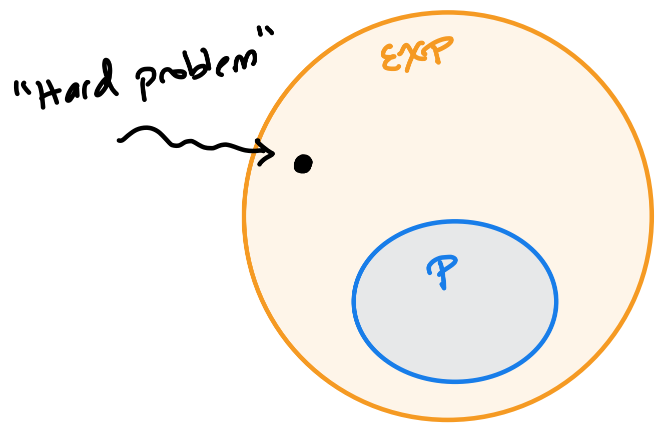


Tractable and Intractable Problems

- **Definition:** $\mathcal{P}$ is the set of decision problems that can be solved in polynomial time

problems with a yes/no answer

- **Definition**: $\mathcal{EXP}$ is the set of decision problems that can be solved in exponential time
- **Theorem:** $\mathcal{P} \neq \mathcal{EXP}$



Allegedly Intractable Problems

INDSET $\equiv_p$ CLIQUE $\equiv_p$ VERTEX COVER

Reduction in both directions ("Equally hard")

Assume: INDSET $\notin$ P

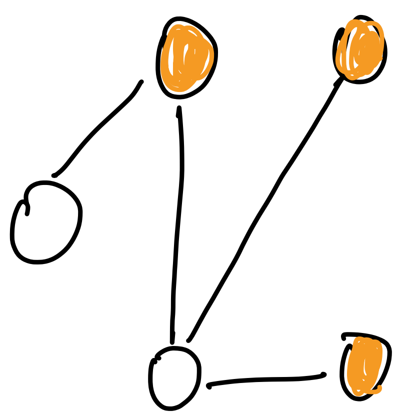
INDSET

Given an undirected graph G and number k, decide if G has an independent set of size $\geq k$,

INDSET

$S \subseteq V$

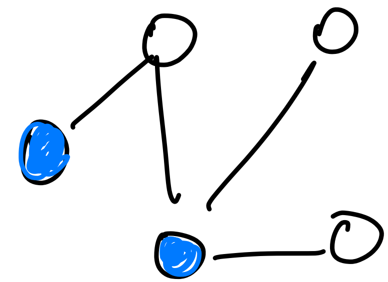
s.t. $\forall (u,v) \in E$ $u \notin S$ or $v \notin S$



VERTEX COVER

$S \subseteq V$ s.t. $\forall (u,v) \in E$

either $u \in S$ or $v \in S$



VERTEX COVER

Given an undirected G and k, decide if G has a vertex cover of size $\leq k$,

Allegedly Intractable Problems

INDSET $\equiv_p$ CLIQUE $\equiv_p$ VERTEX COVER

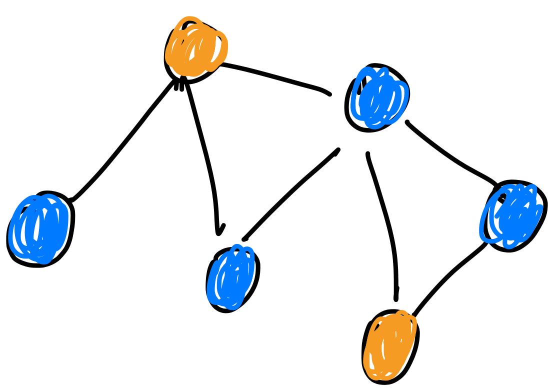
Reduction in both directions ("Equally hard")

Show INDSET $\leq_p$ VERTEX COVER

Key Claim: If S is an independent set in G then V\S is a vertex cover and vice versa. (Immediately gives a reduction.)

Proof:

● = independent set S ● = complement of S



- no edges btw nodes in S
- ⇒ every edge has at least endpoint in V\S
- ⇒ V\S is vertex cover

Allegedly Intractable Problems

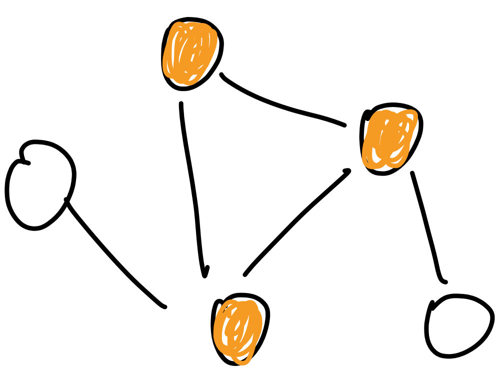
INDSET $\equiv_p$ CLIQUE $\equiv_p$ VERTEX COVER

Reduction in both directions ("Equally hard")

CLIQUE

CLIQUE: $S \subseteq V$
s.t. $\forall u,v \in S$, $(u,v) \in E$

Given G and k, is there a clique of size $\geq k$?

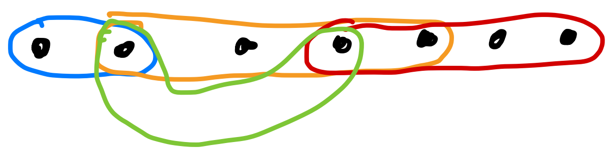


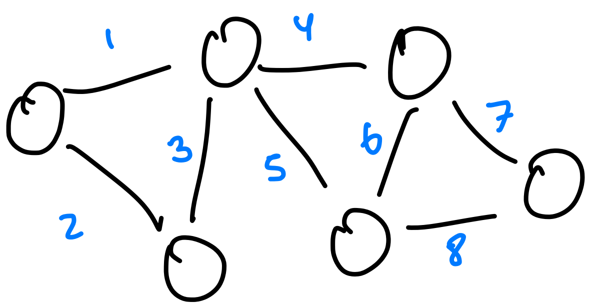
Allegedly Intractable Problems

VERTEX COVER $\le_p$ SET COVER

SET COVER: Given a family $\mathcal{F}$ of sets $S \subseteq \{1, \dots, n\}$

decide: $\exists C \subseteq \mathcal{F}$ such that (1) $\bigcup_{S \in C} S = \{1, \dots, n\}$ (2) $|C| \le k$





- Given $G = (V, E)$ and k for VC
- Number edges $1, \dots, n$
- For every node $v \in V$ let $S_v = \{u \text{ s.t. } (u,v) \in E\}$
- Let $\mathcal{F} = \{S_v\}_{v \in V}$

Allegedly Intractable Problems

VERTEX COVER $\leq_p$ ILP (0/1 Integer Linear Programming)

ILP:

$$\max_x \; c^T x \quad \text{s.t.} \quad Ax \leq b, \quad x \geq 0, \quad \forall i \; x_i \in \{0,1\}$$

Given an ILP (A, b, c) and V decide if there exists a feasible x s.t. $c^T x \geq V$

Given G, k for VC

Write the ILP

$$\min_x \sum_{u \in V} x_u$$

$$\forall (u,v) \in E \quad x_u + x_v \geq 1$$

$$x_u \in \{0,1\}$$

Allegedly Intractable Problems

3-SAT $\leq_p$ INDSET

A 3-SAT formula is a Boolean formula of the form

vars $x_1 \dots x_n$ formula $(x_3 \vee x_7 \vee x_{11}) \wedge (x_5 \vee \overline{x_7} \vee \overline{x_{11}})$

$\wedge \underbrace{(x_2 \vee x_8 \vee \overline{x_{11}})}_{\text{clause}} \wedge \dots$

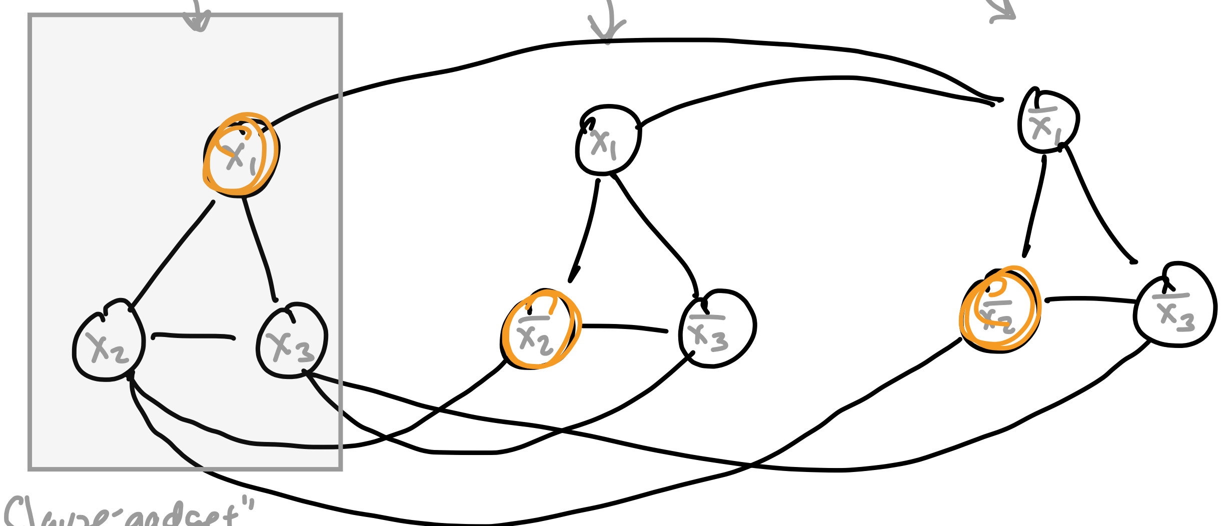
__Assumption:__ 3-SAT $\notin$ P

Allegedly Intractable Problems

3-SAT $\leq_p$ INDSET

Given $F(x) = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$

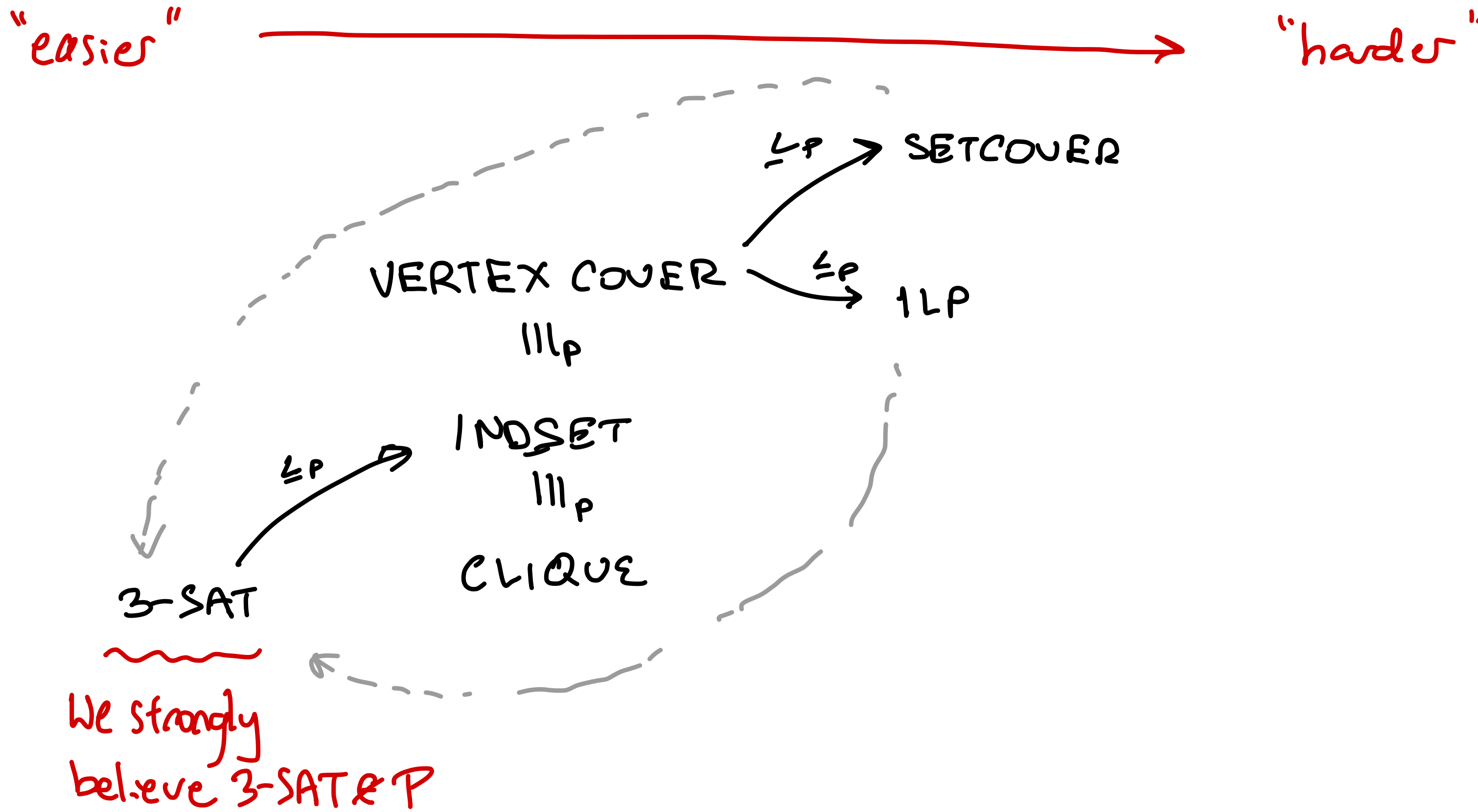
Want a graph G and number k s.t. G has an ind. set of size $\geq k$ iff F is satisfiable



"Clause-gadget"

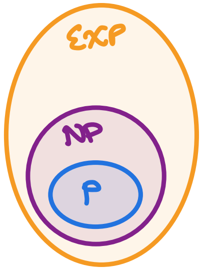
Claim: F is satisfiable if and only if G has an independent set of size $\geq m$

Allegedly Intractable Problems



Note: Reductions are transitive

The Class NP



- **Definition:** $\mathcal{NP}$ is the class of problems for which there is an efficient verifier for solutions
  - An algorithm V is an efficient verifier for problem A if
    - (1) V takes as input I and a solution S
    - (2) V is a polynomial-time algorithm
    - (3) $I \in A$ if and only if there exists a polynomial-size solution S such that $V(I, S) = \text{YES}$

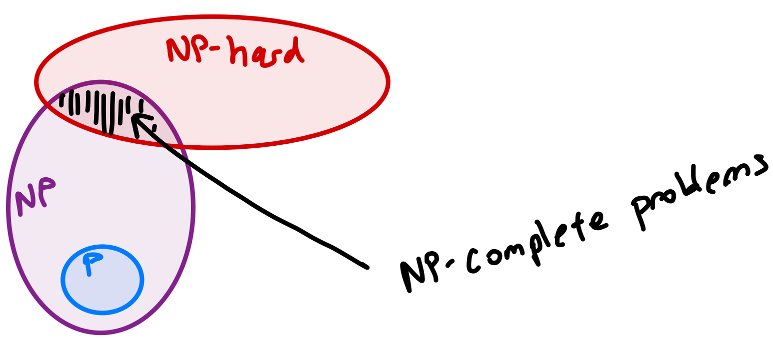
- $\mathcal{P}$ = easy to solve, $\mathcal{NP}$ = easy to check solution

- Natural hard optimization problems are in $\mathcal{N}P$
  - 3-SAT, Vertex-Cover, Independent-Set…

Does $\mathcal{P} = \mathcal{NP}$?

- We do not know, but we believe it very strongly!
  - One of the Millenium Problems

- If we believe $\mathcal{P} \neq \mathcal{NP}$ what does that tell us about problems we care about?
  - **Def:** B is $\mathcal{NP}$-hard if for $A \in \mathcal{NP}$, $A \leq_P B$
  - **Def:** B is $\mathcal{NP}$-complete if $B \in \mathcal{NP}$ and B is $\mathcal{NP}$-hard
  - If B is $\mathcal{NP}$-hard and $B \in \mathcal{P}$ then $\mathcal{P} = \mathcal{NP}$



What problems are $\mathcal{NP}$-complete?

- The Circuit Satisfiability Problem (CKT-SAT)
  - **Input:** Circuit C with n wires and AND/OR/NOT gates
  - **Output:** Decide if there exists x such that $C(x) = 1$

Sort of Cook '71, Levin '73

- **Thm:** CIRCUIT-SAT is $\mathcal{NP}$-complete

What problems are $\mathcal{NP}$-complete?

($\Rightarrow$ 3SAT is NPC)

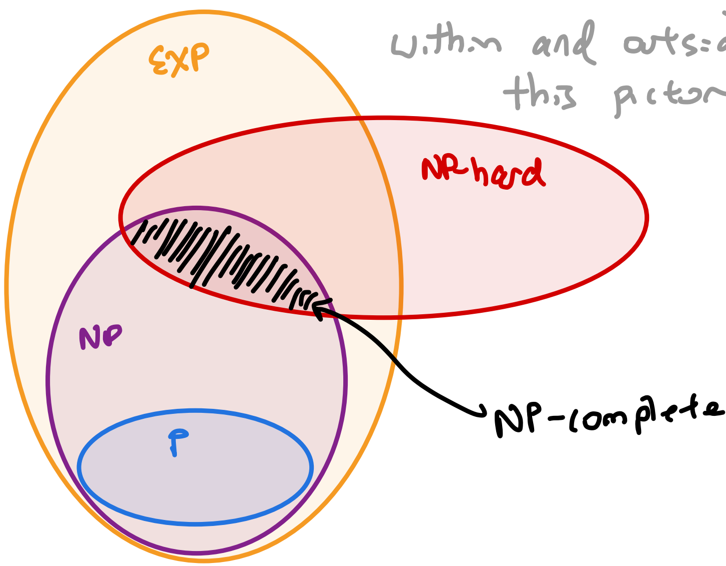
- **Thm (Cook '71, Levin '73):** CKT-SAT $\leq_P$ 3-SAT

What problems are $\mathcal{NP}$-complete?

($\Rightarrow$ 3-SAT is NPC)

- **Thm (Cook '71, Levin '73):** CKT-SAT $\leq_P$ 3-SAT
  - Now we know IND-SET, CLIQUE, VERTEX-COVER, SET-COVER, IP, and 3-SAT are all $\mathcal{NP}$-complete
  - There are thousands more known $\mathcal{NP}$-complete problems in essentially every area within CS

Lots of interesting classes within and outside of this picture.



What problems are $\mathcal{NP}$-complete?

- **Thm (Cook '71, Levin '73):** CKT-SAT $\leq_P$ 3-SAT
  - Now we know IND-SET, CLIQUE, VERTEX-COVER, SET-COVER, IP, and 3-SAT are all $\mathcal{NP}$-complete
  - There are thousands more known $\mathcal{NP}$-complete problems in essentially every area within CS