Lecture 20
NP-complete problems, reductions

CS 161 Design and Analysis of Algorithms

Ioannis Panageas

Example of a reduction

- The 3-SAT problem is NP-complete
- The K-Graph Independent Set (K-GIS) problem is in NP but we don't know if it is hard
- Now, let's reduce the 3-SAT to K-GIS using a poly-reduction.
- Hard part: find the reduction! how to write 3-SAT as a special case of K-GIS.

The 3-SAT problem

- **SAT** (Satisfiability): given a boolean formula, can you make it TRUE;

$$(x_1 \wedge (x_2 \vee \bar{x}_3)) \wedge ((\bar{x}_2 \wedge \bar{x}_3) \vee \bar{x}_1) \Rightarrow x_1 = 1, x_2 = 0, x_3 = 0$$

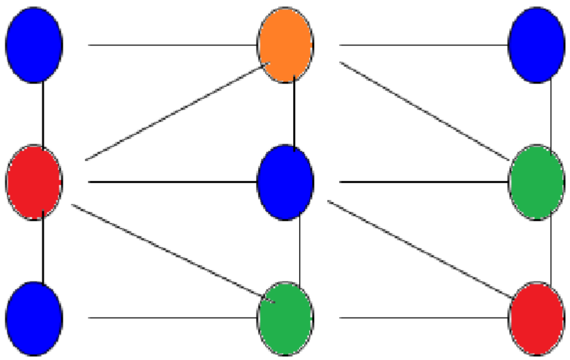
- **3-SAT**: AND clauses, each clause contains 3 variables by OR. For example:

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

- **Cook's Theorem**: 3-SAT is NP-complete

K-Coloring

- Given a graph $G(V, E)$, color the vertices using at most K colors so that all neighboring vertices do not share the same color!
- For example, the following graph can be colored with 4 colors.



- Question: Is K-Coloring NP-complete?

Answer: YES

- First K-Coloring belongs to NP: We can verify in polynomial time if all edges have incident vertices with different colors (in $\Theta(E + V)$ time).
- Then reduce (polynomial reduction) 3-SAT to K-Coloring.

Reduction of 3-SAT to 3-colorability

Goal: We want to solve the 3-SAT problem by making use of an "oracle" that can answer any instance of the 3-colorability problem.

Thought process:

- The input to the 3-SAT problem is a Boolean expression, e.g. $(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee \neg x_4 \vee x_5) \wedge (\neg x_1 \vee x_3 \vee x_5)$.
- The input to the 3-colorability problem is a graph.
- So for the reduction, we have to transform a Boolean expression E into a suitable graph G.

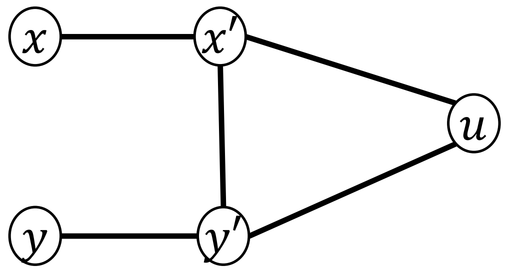
Question: How do we relate a Boolean expression to 3-colorability?

Observation: For a Boolean expression E to be satisfiable, every clause $(x \vee y \vee z)$ in E must evaluate to $true$. [Here, x, y, z are literals.]

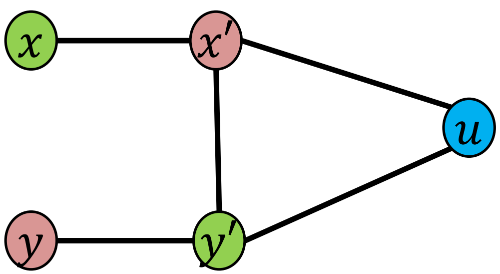
- This means x, y, z cannot all be assigned $false$.

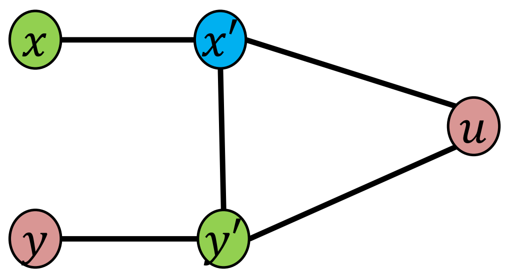
Reduction of 3-SAT to 3-colorability

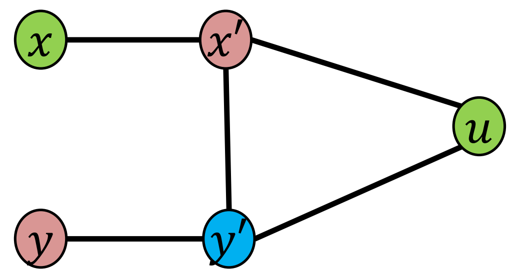
Key Idea 1: Consider a 3-coloring of the following graph:



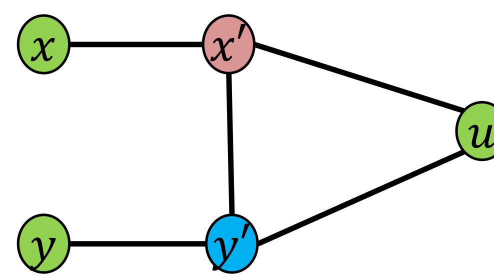
If vertices x, y have distinct colors, then the color of the "output vertex" u can be chosen to be any of the three colors.







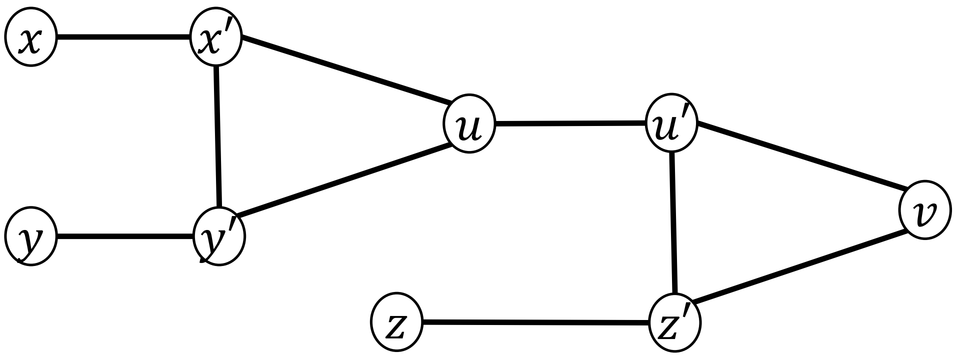
If vertices x, y have the same color, then the color of the "output vertex" u must also be that same color.



Reduction of 3-SAT to 3-colorability

Let's now consider the satisfiability of a single clause $(x \vee y \vee z)$.

Key Idea 2: Consider a 3-coloring of the following "combined graph", using three colors **T**, **F**, **N** (for "true", "false", "neutral").



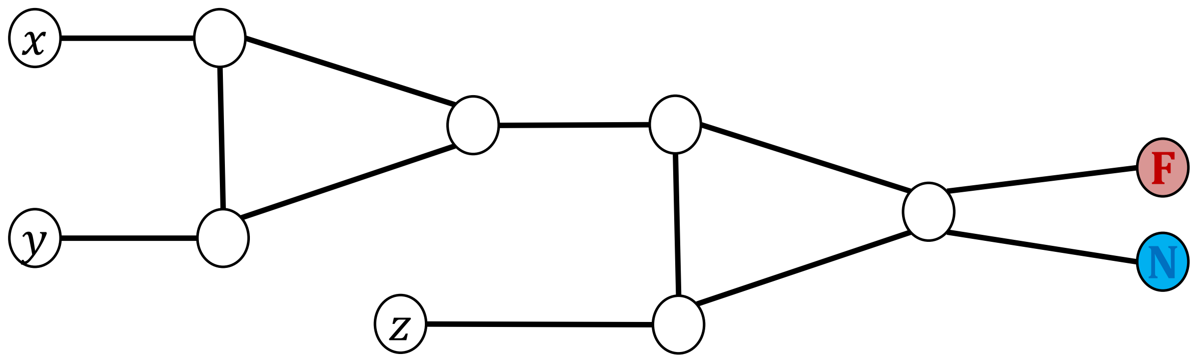
Color each of the vertices x, y, z either **T** or **F**, depending on whether we assign the corresponding variable to be $true$ or $false$.

Key Observation 1: As long as x, y, z are not all colored **F**, then we can always choose the final "output vertex" v to have color **T**.

Key Observation 2: If all three x, y, z are colored **F**, then the final "output vertex" v must have color **F**.

Reduction of 3-SAT to 3-colorability

Key Idea 3: Consider the following "**gadget graph**":



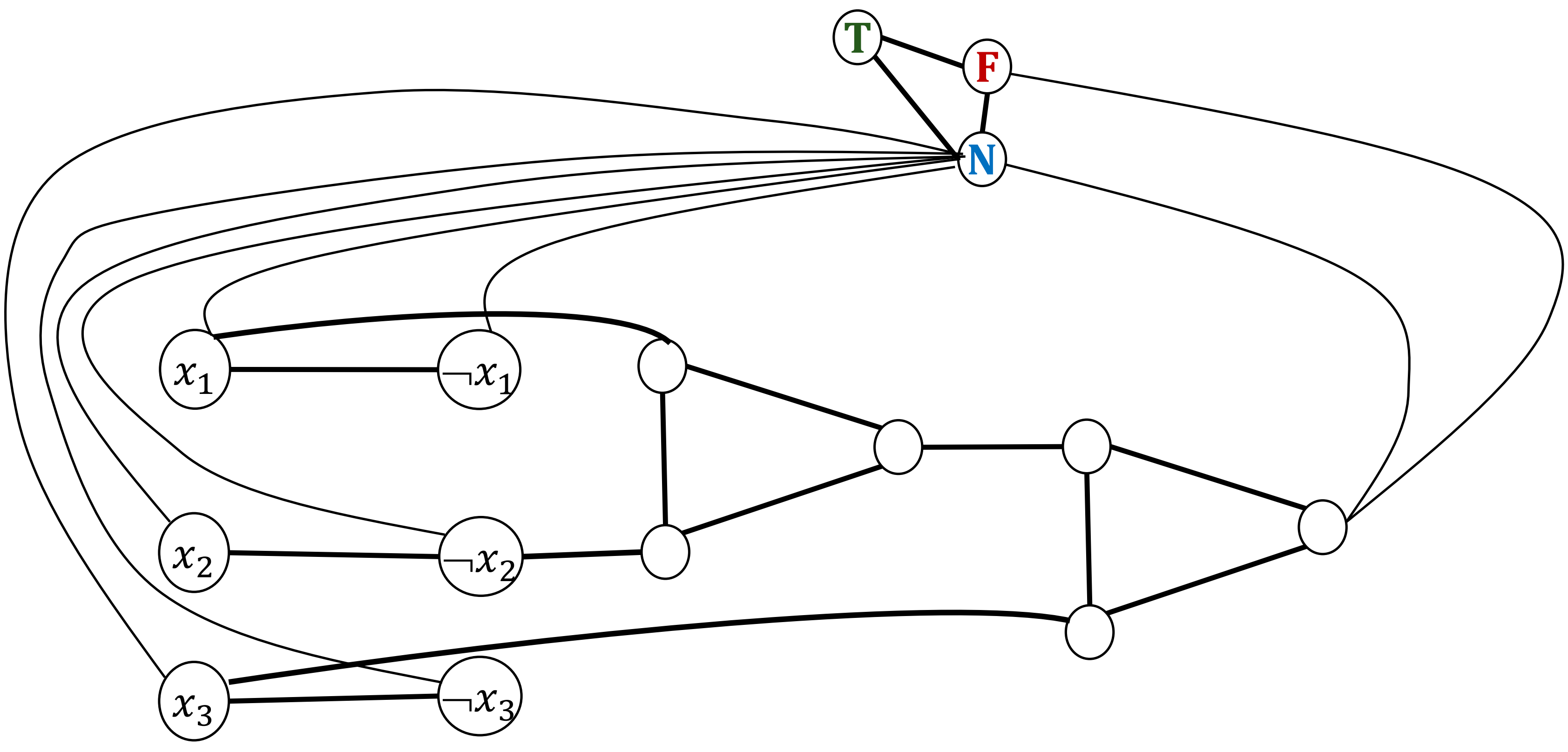
Let each clause $(x \vee y \vee z)$ be associated to a gadget graph.

- The three literals x, y, z in $(x \vee y \vee z)$ shall correspond to the "input vertices" of this gadget graph.
- The final "output vertex" of this gadget graph shall be connected to two other vertices with colors **F** and **N** respectively.

Key Observation: This gadget graph has a 3-coloring if and only if the vertices ⓧ, ⓨ, ⓩ do not all have color **F**.

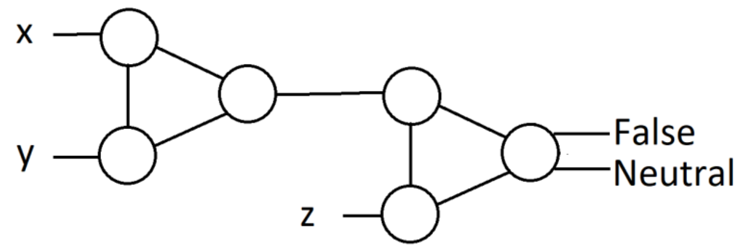
Reduction of 3-SAT to 3-colorability

Example: The Boolean expression "$(x_1 \lor \neg x_2 \lor x_3)$" is transformed to the following graph:



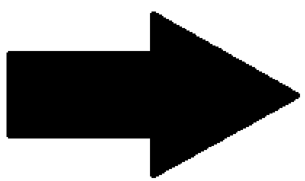
Reduction of 3-SAT to 3-Coloring

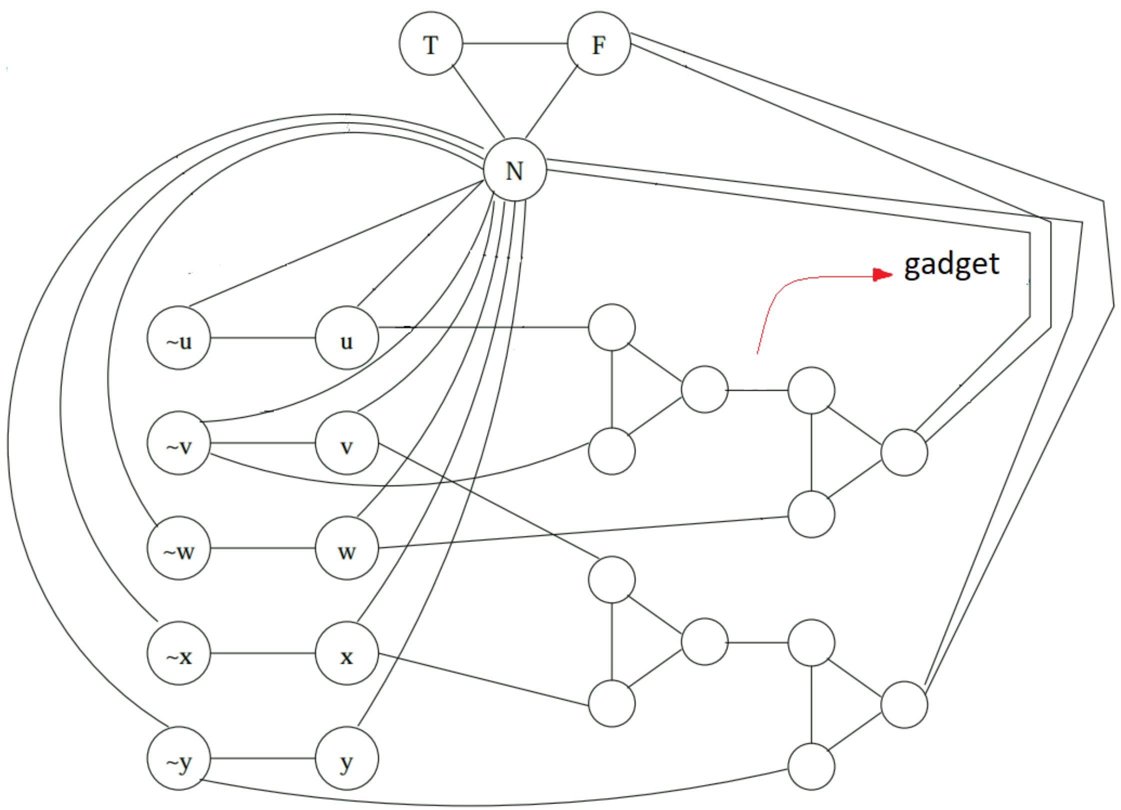
- Gadget graph for $(x \vee y \vee z)$:



- Example:

$(u \vee \bar{v} \vee w) \wedge (v \vee x \vee \bar{y})$





Reduction of 3-SAT to 3-Coloring

- Observe that the reduction is polynomial!

Claim 1: ϕ is satisfiable implies constructed Graph is 3-colorable.

Proof:

- If x_i variable is assigned True, color vertex x_i T and $\bar{x}_i$ F.
- For each clause $(x \vee y \vee z)$ at least one of x, y, z is colored T. Graph gadget for clause $(x \vee y \vee z)$ can be 3-colored such that output is color is T.
- Therefore, no two neighboring vertices have the same color and we used colors T, F, N.

Reduction of 3-SAT to 3-Coloring

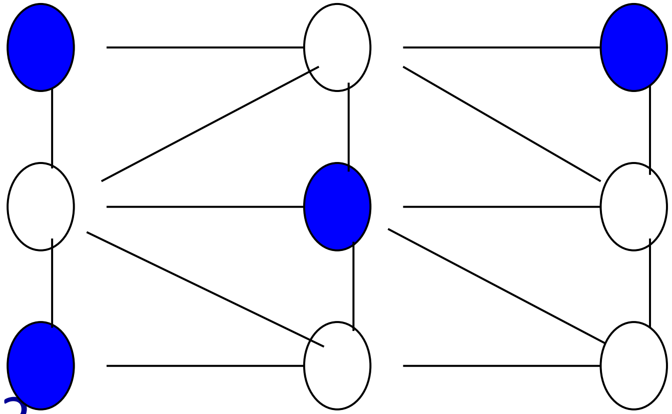
Claim 2: Constructed Graph is 3-colorable (T, F, N) implies ϕ is satisfiable.

Proof:

- Nodes True, False, Neutral use colors T, F, N(need all three)
- If x_i is colored T then set variable x_i to be True, this is a truth assignment.
- Consider any clause $(x \vee y \vee z)$. It cannot be that all x, y, z are False. If so, the output of Graph gadget for $(x \vee y \vee z)$ has to be colored F but output is connected to nodes Neutral and False!

K-Graph Independent Set (K-IS)

- Set of K nodes, all pairs are NOT adjacent to each other
- For example, the following blue nodes are 4-IS (K=4)



- Question: Is K-IS NP-complete?

Answer: YES

- First K-IS belongs to NP: We can verify in polynomial time if a set of K nodes are not adjacent to each other (in $\Theta(K^2)$ time).
- Then reduce (polynomial reduction) 3-SAT to K-IS.

Reduction of 3-SAT to K-IS

Given a formula ϕ with n literals and m clauses that we want to check if it satisfiable.

Construct a graph $G(V, E)$ as follows:

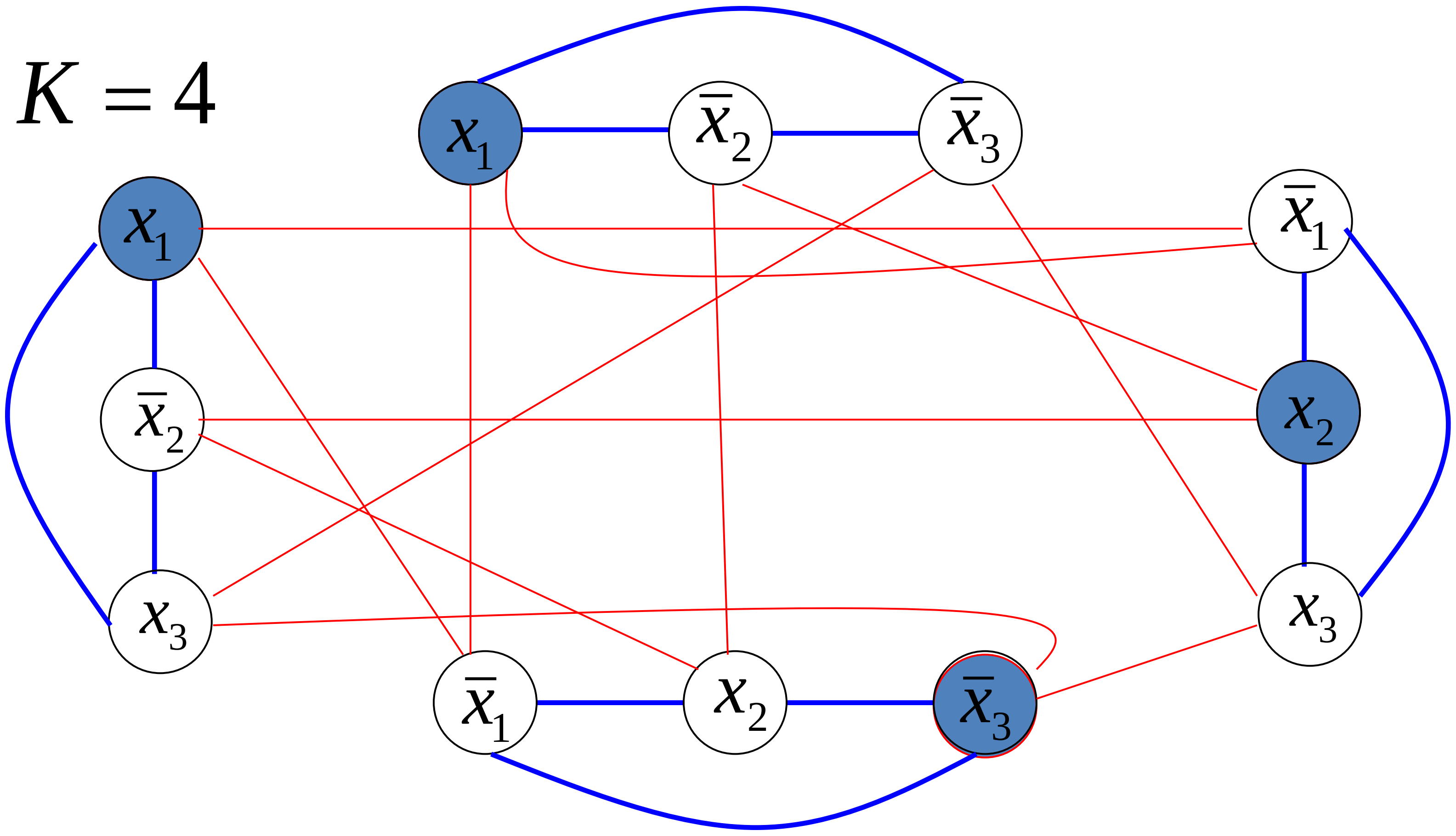
- For each clause $(x \vee y \vee z)$ in ϕ, create three new vertices, one for each variable, and link all the vertices $(x, y), (x, z), (y, z)$.
- Link each vertex (literal) x_i with all its the corresponding negations.
- The construction can happen in polynomial time since $|\mathrm{V}| = 3m$, $|\mathrm{E}| \leq 3m + 2n^2$

- ϕ is satisfiable if and only if there exists an IS of size m!

Reduction of 3-SAT to K-IS

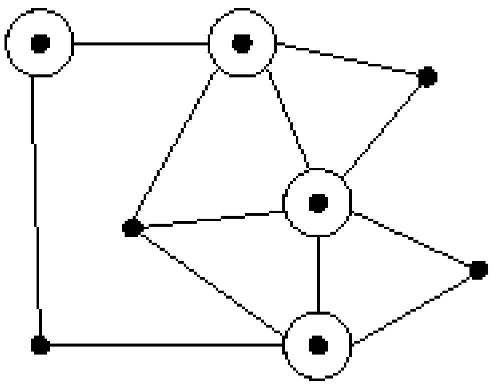
$$\phi \coloneqq (x_1 \vee \bar{x}_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2 \vee \bar{x}_3) \wedge (x_1 \vee \bar{x}_2 \vee x_3)$$

$K = 4$



Vertex Cover (VC)

- Vertex Cover (VC): is there a subset of at most k vertices, such that it connect to all edges?



e.g. in this graph, 4 of the 8 vertices is enough to cover

- Question: VC is NP Complete?
  - Answer: YES
    - First, it belongs in NP (why?)
    - Then Reduce 3-SAT to VC (or there is something simpler?)

Reduction of K-IS to Vertex Cover (VC)

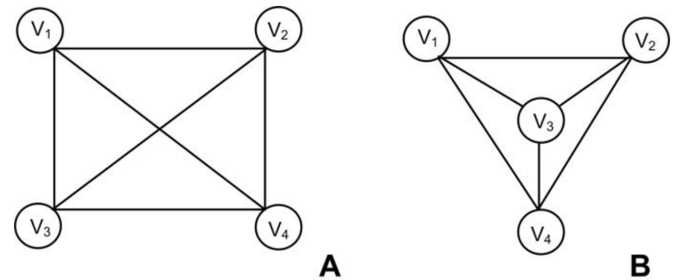
- Given a graph $G(V, E)$, with $|\mathrm{V}| = \mathrm{n}$, suppose there exists an Independent Set of size k.

- Lemma: If $G(V, E)$, is a graph, then set of vertices S is an *independent set* if and only if $V - S$ is a *vertex cover*.

Proof: Let S be an independent set, and $e = (u, v)$ be some edge. Only one of u, v can be in S. Hence, at least one of u, v is in $V - S$. So, $V - S$ is a vertex cover. The other direction is similar.

CLIQUE

- K-clique: k vertices, all vertices are adjacent to each other
  - E.g. both of these are 4-CLIQUE

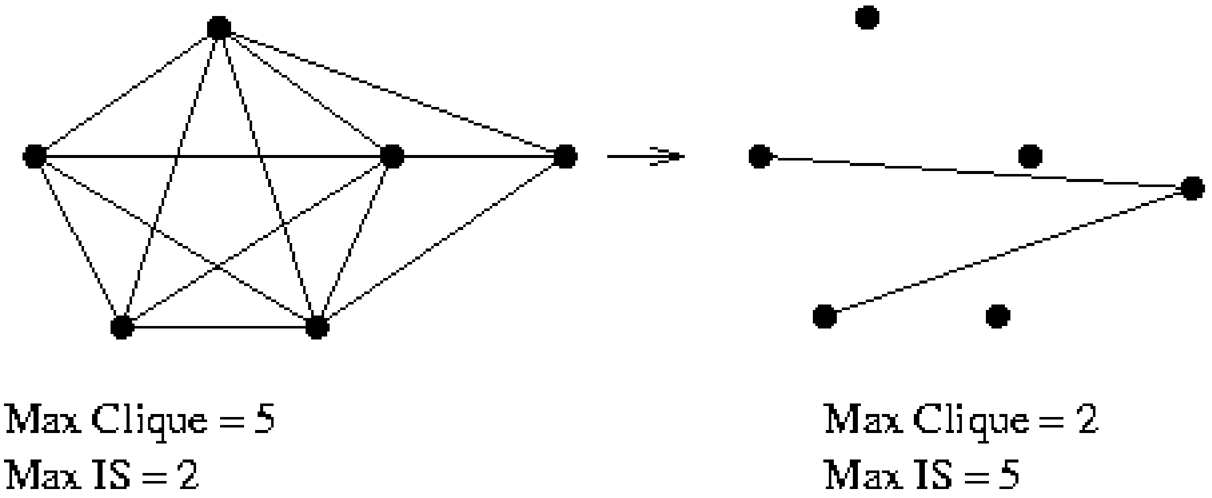


- CLIQUE Problem: in a graph, does k-clique exists?

- Question: CLIQUE is NP-Complete?
  - Answer: YES
    - First, it belongs in NP (why?)
    - Then, reduce Independent set to CLIQUE

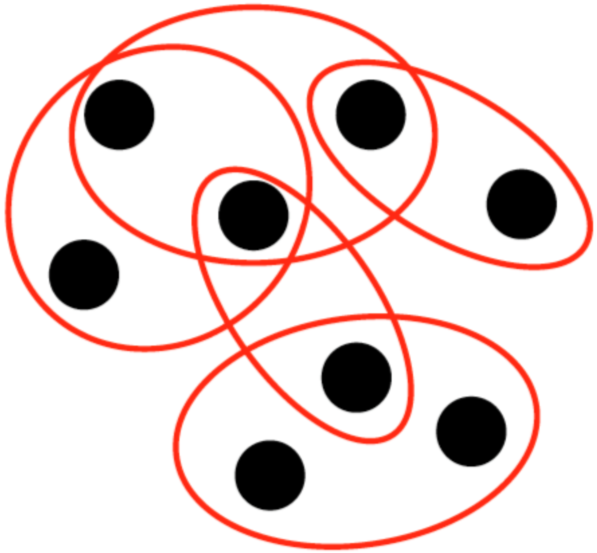
Reduction of IS to CLIQUE

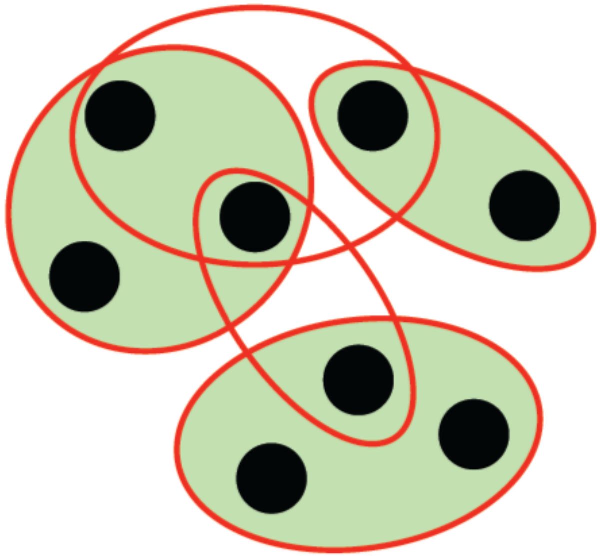
- Reduce Independent set (IS) to CLIQUE
  - Complement a graph!
  - CLIQUE become IS, IS become CLIQUE
  - (most reduction are complicated, this is exceptionally simple…)



Set Cover

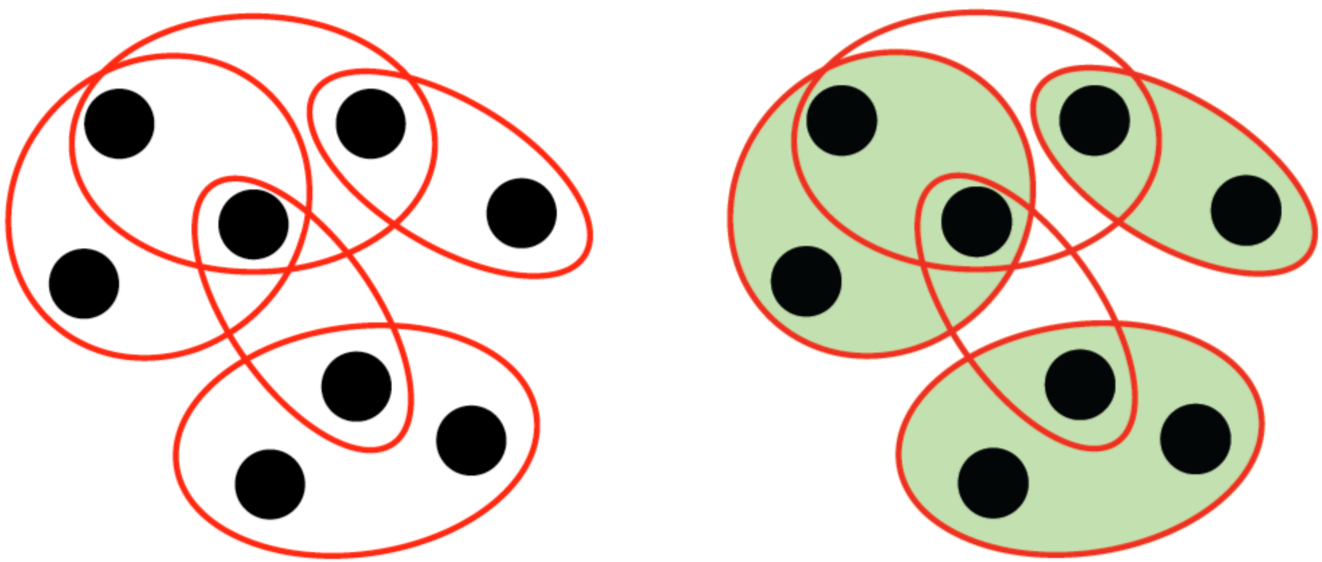
- Set Cover: Given a set of U of elements and collection of set $S_1\ S_2\ S_3\ \ldots\ S_m$ of subset of U. Is there a collection of at most k set, whose Union is U?





Reduction of VC to Set Cover

- Question: Set Cover is NP-Complete?
  - Answer: YES
    - First, show that is NP (Easy)
    - Then, prove that vertex cover can reduce to set cover.



Reduction of VC to Set Cover

- Let $G = (V, E)$ and k be an instance of vertex cover
- Now,
  - $U = E$ (set of edges)
  - Create set of S_1, S_2, S_3 ... .
    - S_1 = all edges adjacent to node 1
    - S_2 = all edges adjacent to node 2
    - Etc
- Conclusion: If G has a vertex cover of size $\leq k$, then U has a set cover $\leq k$.

Subset Sum

- Subset Sum: (Recall the Reformulation of the partition problem!) Given a set S of integers and a target integer t, does there exist $S' \subseteq S$ with $\sum_{x \in S'} x = t$.

- *Recall that Subset Sum is reduced to Knapsack!*

- Question: Subset Sum is NP-Complete?

Answer: YES

- First, it belongs in NP (why?).
- Then, reduce VC to Subset Sum.

Reduction of VC to Subset Sum

- Let $G = (V, E)$, with $|\mathrm{V}| = n$, $|\mathrm{E}| = m$ and and assume that has a VC of size k. Number the vertices from 0 to $n - 1$ and the edges from 0 to $m - 1$.

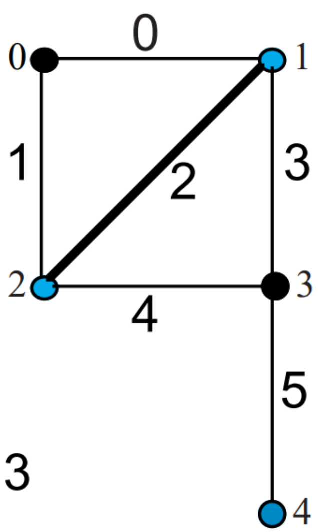
- Let $S = \{x_0, \dots, x_{n-1}\} \cup \{y_0, \dots, y_{m-1}\}$. Each x_i consists of $m + 1$ digits (in base 10) and can be written as $x_{i,m}x_{i,m-1}\dots x_{i,0}$. The digit $x_{i,m}$ is always 1. Each remaining $x_{i,j}$ is 1 if vertex i is an endpoint of edge j, 0 otherwise.

- Each y_i has $i + 1$ digits: a 1 followed by i 0's. Finally, let t be the base 10 representation of the integer k followed by m 2's.

Reduction of VC to Subset Sum

The reduction on an example

Vertex Cover instance



k = 3

Subset Sum instance

$$x_0 = 1000011$$
$$x_1 = 1001101$$
$$x_2 = 1010110$$
$$x_3 = 1111000$$
$$x_4 = 1100000$$
$$y_0 = 1$$
$$y_1 = 10$$
$$y_2 = 100$$
$$y_3 = 1000$$
$$y_4 = 10000$$
$$y_5 = 100000$$
$$t = 3222222$$

$k = 3$

Reduction of VC to Subset Sum

Graph has VC of size k implies that there is a subset of sum k.

Proof.
Assume the graph has a VC V_0 of size k. Let
$S_0 = \{x_i \mid i \in V_0\} \cup \{y_i \mid only\ one\ endpoint\ of\ edge\ i \in S_0\}$.

Since there are three 1's in positions 0 through $m-1$, there will be no carries from those positions. The choice of S_0 items guarantees each of these digit positions has sum 2, as required by t. Since $|V_0| = k$, the x_i's in S_0 will contribute exactly k 1's in position m for a total of k.

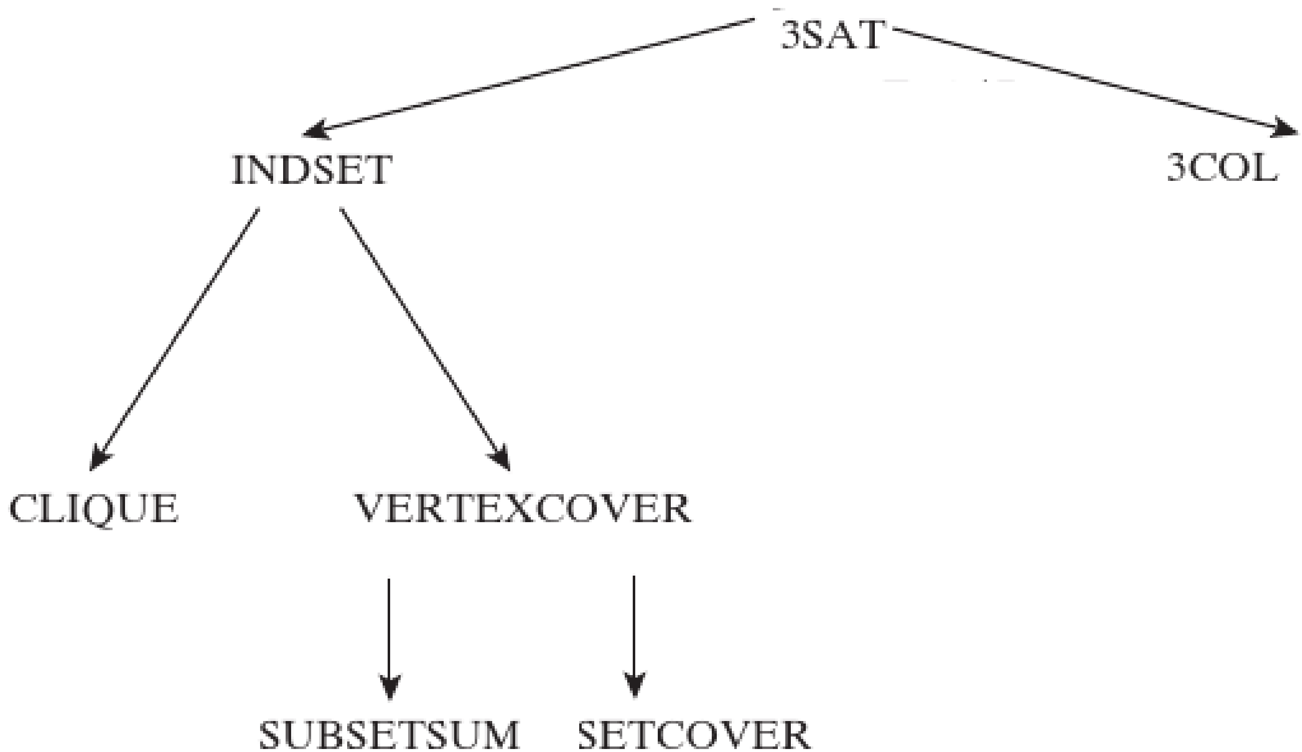
Reduction of VC to Subset Sum

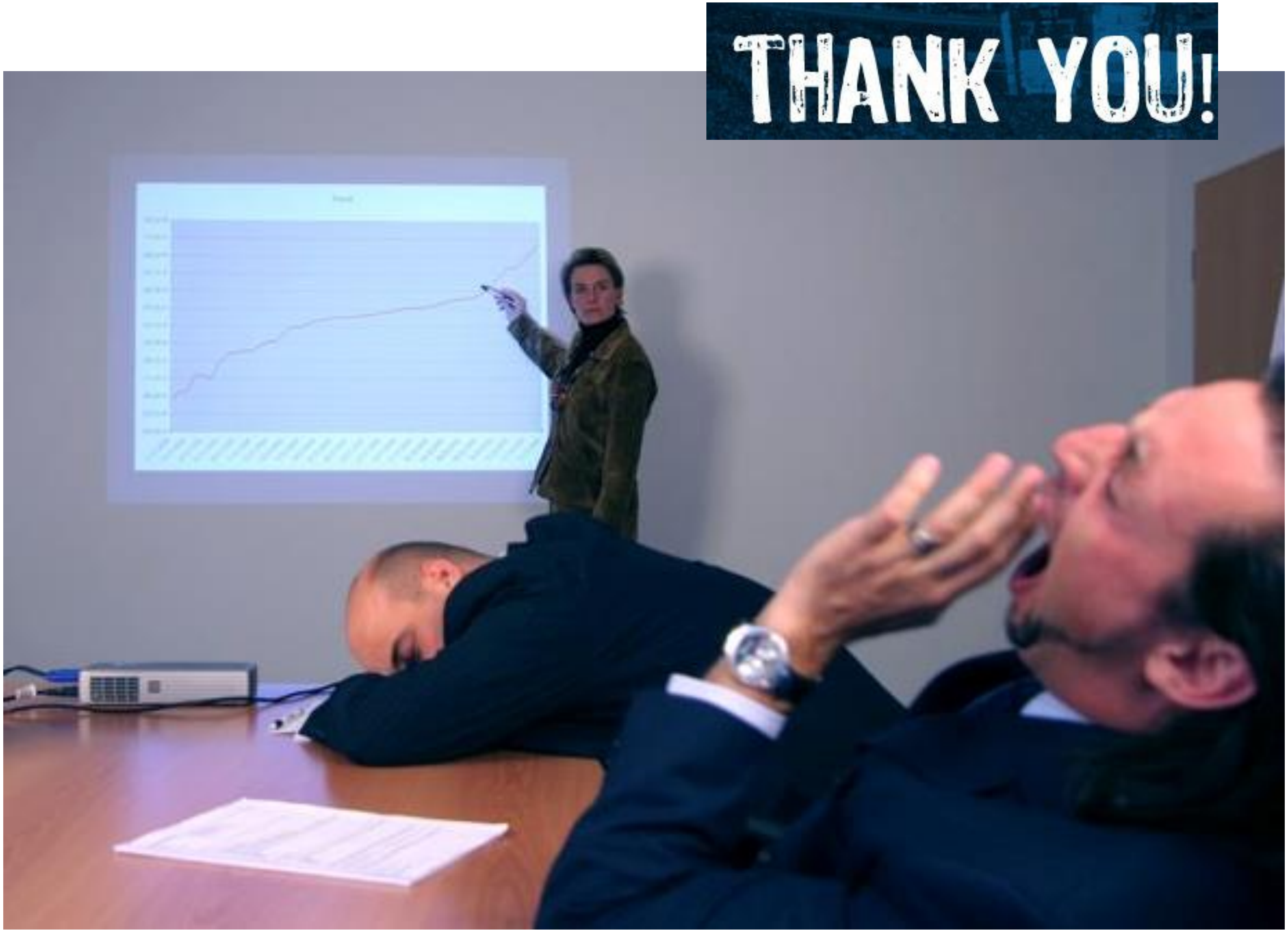
There is a subset of sum k imples the graph has VC of size k

Proof.
Assume S_0 is a set of numbers with sum k. Let V_0 be the set of all vertices i for which $x_i \in S_0$.

Since there are no carries in the lowest m digits, there must be exactly k vertices in V_0 (to get t to start with k) and each edge must have at least one endpoint in V_0 (observe that if edge i has no endpoints in V_0 then S_0 has only a single 1 among all the i-th digits and the sum of S_0 cannot have a 2 in that position).

Web of reductions of the Lecture





This is the last lecture of CS161!