Claim: G contains an ind. set of size K
$\iff$ the formula f is satisfiable

Proof: 1) suppose f is satisfiable. Pick any satisfying assignment. Each clause in f has ≥ 1 TRUE literal.
Thus we can choose a subset S of K vertices in G that contains exactly one vertex per group such that the corresponding K literals are all TRUE. The set S is an ind. set because it does not contain both endpoints of any edge of a group, nor of any edge that connects inconsistent literals

2) suppose G contains an ind. set of size K. Each vertex in S must be in a different group. Assign TRUE to each literal of S. Since inconsistent literals are connected by an edge, this assignment is consistent. Since S contains 1 vertex per group, each clause in f contains (at least) one TRUE litual
$\Rightarrow f$ is satisfiable

Exercises: (easy)

Clique : compute the largest complete subgraph in G

Show that Clique is NP-hard

Definition: a vertex cover of a graph is a set of vertices that includes at least one endpoint of every edge of the graph

Vertex Cover : compute the smallest vertex cover in G

Show that Vertex Cover is NP-hard

Approximation Algorithms

... for NP-hard problems.

Assumption: $P \neq NP$

Optimization problems:

$$\Pi : \underbrace{I}_{\text{inputs}} \times \underbrace{S}_{\text{solutions}}$$

$$c : S \to \mathbb{R}^+$$

$$\forall i \in I \quad S(i) = \{ s \in S : i \Pi s \}$$

"feasible solutions"

$$s^* \in S(i) \quad \text{and} \quad c(s^*) = \begin{matrix}\min \\ \max\end{matrix} \; c(S(i))$$

Approximation:

$s \in S(i)$

ok if $s \neq s^*$, but I want:

1) guarantee on the quality of s

2) guarantee on the complexity: polynomial-time algorithm

Definition: let Π be an optimization problem, and let A_Π be an algorithm for Π that returns, $\forall i \in I$, $A_\Pi(i) \in S(i)$.
We say that A_Π has an <u>approximation factor</u> of $\rho(n)$ if $\forall i \in I$ s.t. $|i| = n$ we have

min: $$\frac{c(A_\Pi(i))}{c(S^*(i))} \leq \rho(n)$$

max: $$\frac{c(S^*(i))}{c(A_\Pi(i))} \leq \rho(n)$$

$c : S \to \mathbb{R}^+ \Rightarrow$ ≥ 1

Goal: $\rho(n) = 1 + \varepsilon$ with ε as small as possible

We'll get $\varepsilon = 1$ for vertex cover $\rightarrow$ "2-approx."

$\varepsilon = \log_2 n$ for set cover

Much stronger approximation: $\rho(n) = 1+\varepsilon \quad \forall \varepsilon > 0$

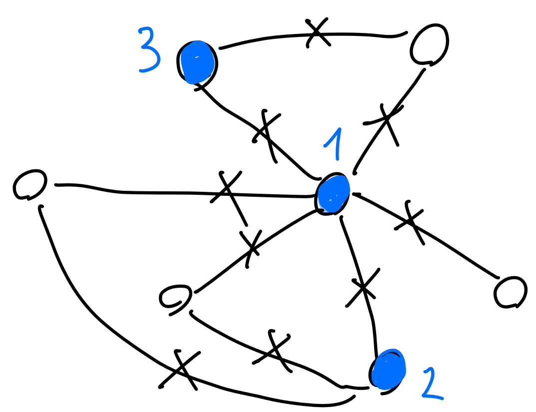
Definition: an <u>approximation scheme</u> for Π is an algorithm with 2 inputs $A_\Pi(i,\varepsilon)$ that $\forall\, \varepsilon > 0$ is a $(1+\varepsilon)$-approximation

Definition: an approximation scheme is <u>polynomial</u> (<u>PTAS</u>) if $A_\Pi(i,\varepsilon)$ is polynomial in $|i|$ $\forall\, \varepsilon$ fixed.

<u>Approximation</u> <u>algorithms for Vertex Cover</u>

very first algorithm you can think of?

- select the vertex with highest degree
- "remove" the covered edges
- repeat



Unfortunately, for this algorithm $\rho(n) = \Omega(\log n)$

How to prove a LB on $\rho(n)$? It's enough to show one "bad" input instance

Exercise: show a lower bound on $\rho(n)$ for this algorithm

the higher the better (log n is difficult)

another algorithm:

- choose any edge
- add its endpoints to the solution
- "remove" the covered edges
- repeat

Approx-Vertex-Cover (G)

$V' = \emptyset$

$E' = E$

while $E' \neq \emptyset$ do

let (u, v) be an arbitrary edge of E'

$V' = V' \cup \{u, v\}$

$E' = E' \setminus \{$all (u, z) or (v, w) edges$\}$

return V'

Complexity: $O(n+m)$

Analysis: we'll show $\frac{|V'|}{|V^*|} \leq 2$

A = set of selected edges

A is a <u>matching</u>: $\forall e, e' \in A \Rightarrow e \cap e' = \emptyset$

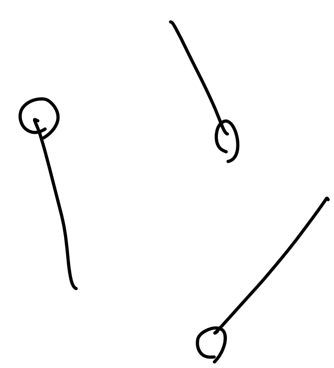
i.e. no vertices in common

Approx-vertex-cover selects a <u>maximal</u> matching

$\forall$ edge y, $A \cup y$ is <u>not</u> a matching

1) $|V^*|$ vs. $|A|$?

A is a matching $\Rightarrow$
in V^* there must be ≥ 1
vertex $\forall$ edge of A



$|V^*| \geq |A|$

2) $|V'|$ vs. $|A|$?

$|V'| = 2|A|$ by construction

$$\overset{1)+2)}{\Longrightarrow} \quad \cancel{|V'| = 2|A| \leq 2|V^*|}$$

$$\Longrightarrow \quad \frac{|V'|}{|V^*|} \leq 2$$

Approx-vertex-cover is a 2-approximate algorithm for Vertex Cover

Exercise: show that the approximation factor of Approx-vertex-cover is <u>exactly</u> 2