Where we left off last time:

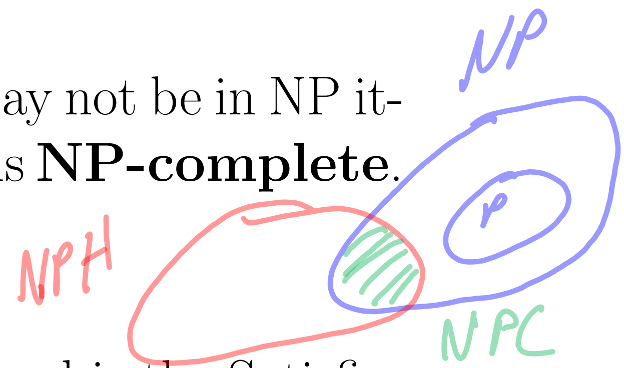
- We now have a method of showing that one problem Π' at worst as hard as another problem Π.

- If we can prove the reduction in both directions (i.e. show $\Pi' \leq \Pi$ and $\Pi \leq \Pi'$), then a polynomial time algorithm for one problem would imply a polynomial time algorithm for the other as well.

- Suppose there was a problem Π such that for any Π' in NP, it was true that $\Pi' \leq \Pi$? Then to show that a problem Π' in NP was hard, we would only need to show $\Pi \leq \Pi'$ (as $\Pi' \leq \Pi$ would be implied by Π' being in NP).

If a problem Π satisfies $\Pi' \leq \Pi$ for any Π' in NP, then we say that Π is **NP-hard**.

A problem that is NP-hard may or may not be in NP itself. If it is in NP, then we say that it is **NP-complete**.



The first problem shown to be NP-hard is the Satisfiability problem (SAT):

- Given: A set of n boolean variables, and a formula ϕ with m clauses:

$$(x_1 \vee x_2 \vee x_5) \wedge (x_3 \vee \overline{x_5})$$

$x_1 = T$

$x_5 = F$

- Determine if there exists T/F assignments to the variables so that ϕ is satisfied.

In 1971, Cook proved that SAT is NP-hard. Note that SAT is in NP (given an assignment, we can determine if it is a satisfying assignment in polynomial time). Therefore SAT is also NP-complete.

Thanks to Cook, we can prove other problems Π' in NP are NP-complete by showing SAT $\leq \Pi'$. Once this is done, we can show other problems Π'' in NP are NP-complete by reducing from either.

Recall that the Clique problem asks if there is a clique of size at least k in an input graph G, where a clique is a subset of a graph such that all pairs of vertices in the subset have an edge connecting them.

Next, we will show that SAT $\leq$ Clique. This will imply that Clique is NP-complete.

Last time, we showed Clique $\leq$ Independent Set. By the transitivity of reductions, we will have that SAT $\leq$ Clique also implies that Independent Set is NP-complete.

To show that SAT $\leq$ Clique, we need to convert a the input to SAT (a formula ϕ) into the input for Clique (a graph $G = (V, E)$ and an integer k) such that ϕ is satisfiable iff G has a clique of size at least k.

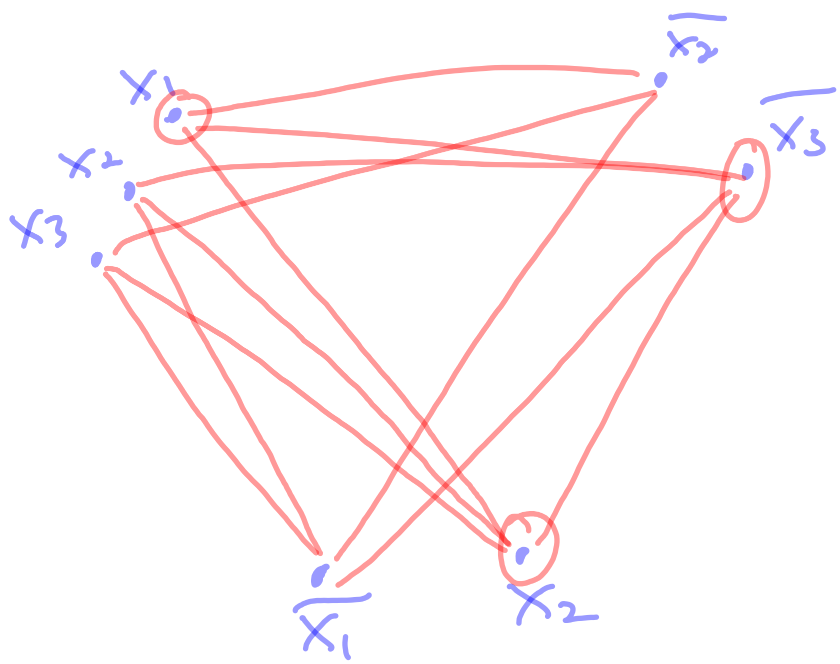
Intuitively, if ϕ is satisfiable then G should have a large clique, and if ϕ is not satisfiable then G should not have a large clique. But we don't know if ϕ is satisfiable or not. Regardless, how can we construct G and choose k so that this property holds?

Notation: We call x_i or $\overline{x_i}$ a *literal*.

Construction of the graph G:

- For each literal t occurring in ϕ, create a vertex v_t.
- Add the edge $\{v_t, v_{t'}\}$ iff:
  - t and t' are not in the same clause, and
  - t is not the negation of t'.

$\phi : (x_1 \vee x_2 \vee x_3) \wedge (\overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_2)$



$x_1 = T$

$x_2 = T$

$x_3 = F$

Claim: ϕ is satisfiable iff G has a clique of size at least m.

$\Rightarrow$ Suppose ϕ is satisfiable. Then there is a literal in each clause that is true.

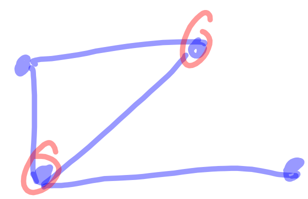
Arbitrarily pick one true literal from each clause. Their corresponding vertices in G will form a clique of size m.

$\Leftarrow$ Suppose there is a clique of size m in G. Set the literals for each of the corresponding vertices in the clique to be true.

This is a valid assignment, because literals x_i and $\bar{x}_i$ cannot both be in the clique, so the truth assignment is valid. Each clause has a true literal, so ϕ is satisfied.

So now we have three known NP-complete problems: SAT, Clique, and Independent Set (IS).

Given a graph $G = (V, E)$, a *vertex cover* of G is a subset $C \subseteq V$ of the vertices such that for each edge $\{u, v\} \in E$, either u or v is in C.



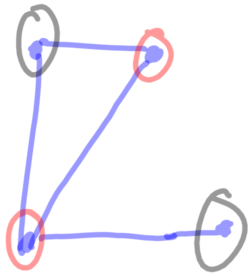
The Vertex Cover problem (VC) takes a graph G and an integer k as input and seeks to determine if there is a vertex cover of G of size at most k.

We will show IS $\leq$ VC, which will imply that VC is NP-complete.

To show IS $\leq$ VC, we will need to take any input to IS (a graph $G = (V, E)$ and integer k) and construct an input to VC ($G' = (V', E')$ and integer k') such that G has an IS of size at least k iff G' has a VC of size at most k'.

Intuitively, if G ~~should~~ has a large IS then G' should have a small VC, and if G does not have a large IS then G' should not have a small VC. But we don't know the size of the largest IS of G. Regardless, how can we construct G' and choose k' so that this property holds?

What is the relationship between an IS and a VC?



The complement of a VC is an IS.

Construction: $V' = V$, $E' = E$, $k' = |V| - k$.

Reasoning: S is an IS of G iff $V \setminus S$ is a VC of G'.