23 Reductions, Polynomial Reducibility, Satisfiability Problem

- Reductions
- NP and NP-Completeness

23.1 Reductions

To show that problem X is at least as hard as Y, we can give a **reduction** from Y to X, a procedure that solves Y using the ability to solve X.

To have this quantify efficiency, we want the procedure to be efficient.

Definition 23.1
If any instance of Y can be solved by a polynomial time algorithm that can make polynomially many calls to a procedure for solving instances of X, then we say "Y is polynomial-time reducible to X" and write $Y \leq_P X$.

$Y \leq_P X$ - note that this means that X is at least as hard as Y. Y could be solved in a smaller amount of time than X. If X can be solved in polynomial time, we know that Y can also be solved in polynomial time. If X can only be solved in exponential time, Y could still be solved in polynomial time.

Example 23.2
We showed Bipartite Matching $\leq_P$ Max Flow.

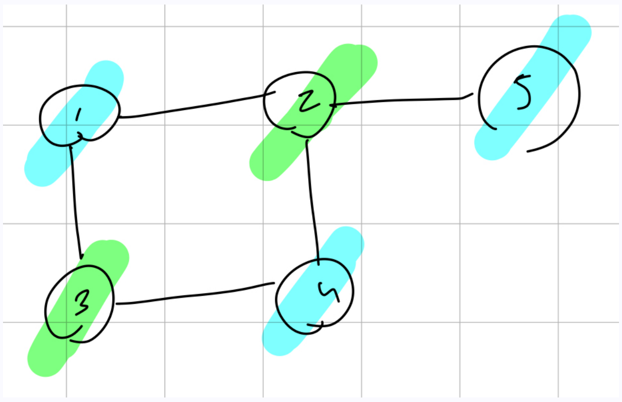
Definition 23.3
An **independent set** in a graph is a subset of vertices, no two of which are adjacent.

The **maximum independent set problem** asks us to find a largest independent set in a given graph.

Definition 23.4
A **vertex cover** of a graph is a subset of vertices such that every edge has at least one end in that subset.

The **vertex cover problem** asks us to find a smallest vertex cover.

Example 23.5
Observe the following example of a graph:



The nodes highlighted blue represent a solution to a maximum independent set, and the ones highlighted green represent a minimum vertex cover.

Lemma 23.6
For any graph $G = (V, E)$, a vertex subset $S \subseteq V$ is an independent set iff $V \setminus S$ is a vertex cover.

Proof. Suppose S is an independent set. Then for any $\{u, v\} \in E$, at most one of u, v is in S, so at least one is in $V \setminus S$, so $V \setminus S$ is a vertex cover.

Suppose $V \setminus S$ is a vertex cover.
Suppose for a contradiction that $u \in S$ and $v \in S$ and $\{u, v\} \in E$. Then, neither end of e is in $V \setminus S$, so this is a contradiction.
So, S is an independent set. □

This shows that Independent Set $\leq_P$ Vertex Cover, and Vertex Cover $\leq_P$ Independent Set.

Now consider the **Set Cover** problem:
Givne a set U and subsets $S_1, S_2, \cdots, S_m \subseteq U$, find a minimal subset of the S_i's so that their union is all of U.

Theorem 23.7
Vertex Cover $\leq_P$ Set Cover

Proof. Given an instance of Vertex Cover (a graph $G = (V, E)$), construct an instance of set cover.

Let U = E. For all $v \in V$, let S_v = set of edges incident on v.

Claim: G can be covered with k vertices iff U can be covered with k S_v's.

- If $\{v_1, \cdots, v_k\}$ is a vertex cover, then $S_{v_1}, \cdots, S_{v_n}$ includes all the edges.
- If $S_{v_1}, \cdots, S_{v_n}$ is a set cover, then $\{v_1, \cdots, v_k\}$ is a vertex cover.

□

23.1.1 Boolean Satisfiability

Definition 23.8
Given a set of Boolean variables $x_1, \cdots, x_n$, a **term** or **literal** is x_i or $\overline{x}_i$.

A **clause** is a disjunction of terms (Ex: $x_2 \vee x_6 \vee \overline{x}_{11}$).

We say a set of clauses is **satisfiable** if there is an assignment of the x_i's to true/false so that all clauses evaluate to true.

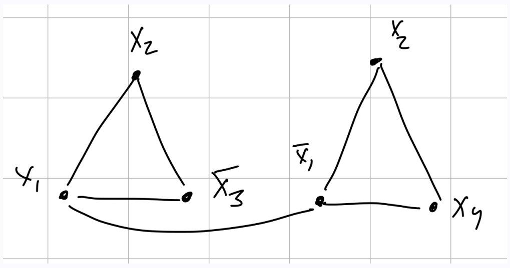
The **Satisfiability** problem (SAT) asks whether a given set of clauses is satisfiable.
In 3SAT, each clause has exactly 3 terms (no variables can be repeated within a clause).

Theorem 23.9
3SAT $\leq_P$ Independent Set

Proof. Given a 3SAT instance, we use "gadgets" to construct a graph that has a big independent set iff the 3SAT instance is satisfiable.

For clause C_j, introduce vertices v_{j_1}, v_{j_2}, v_{j_3}. Connect them with a triangle.
Connect v_{jl} and $v_{j'l'}$ if term l in C_j is the negation of term l' in $C_{j'}$.

Example 23.10
$(x_1 \vee x_2 \vee \overline{x}_3) \wedge (\overline{x}_1 \vee x_2 \vee x_4)$



The smallest independent set in this graph is just the two x_2 nodes.

We draw an edge between literals and their negations, because we can not set them both to be true, and so if we treat this as an independent set problem where if a literal is in our set then it is true, then we can not add the negations of those literals in the set, corresponding to a literal and its negation not being both true.

Claim: formula is satisfiable iff this graph has an independent set of size equal to the number of clauses □