CMPSCI 611: Advanced Algorithms
Lecture 18: NP-Completeness

Andrew McGregor
Problem 1: Clique

Definition
A clique of size $k$ in a graph $G$ is a completely connected subgraph of $G$ with $k$ vertices.

- **Input:** Given graph $G = (V, E)$ and integer $k$.
- **Question:** Does $G$ contain a clique of size $k$?
Problem 2: 3-SAT

▶ Input: A boolean formula $\phi(x_1, \ldots, x_n)$ in *conjunctive normal form* with $m$ clauses and 3 literals per clause, e.g.,

$$(x_1 \lor \bar{x}_2 \lor x_3) \land (\bar{x}_1 \lor x_2 \lor \bar{x}_3)$$

where $\bar{x}_i$ is “not $x_i$”, $\land$ is “and”, $\lor$ is “or.” We call $x_i$ and $\bar{x}_i$ *literals*.

▶ Question: Is there a setting of each $x_i$ to TRUE or FALSE such that the formula is satisfied.
Theorem
There exists a polynomial time algorithm for 3-SAT iff there exists a polynomial time algorithm for Clique.

This will follow because:
1. Clique is “in NP”
2. 3-SAT is “NP-complete”
3. There exists a “polynomial time reduction” from 3-SAT to Clique.

It is widely believed that 3-SAT can’t be solved in poly-time. Hence, the belief that Clique can’t be solved in poly-time is just as strong.
P and NP Definitions

Definition
Π is a decision problem if it only has a “yes” or “no” answer.

Definition (P)
Π ∈ P iff there exists a polynomial time algorithm A such that:

\[(X \text{ is a “yes” instance of } \Pi) \iff (A(X) = “yes”)\]

Definition (NP)
Π ∈ NP iff there exists a polynomial time algorithm A such that:

\[(X \text{ is a “yes” instance of } \Pi) \implies (\exists Y : |Y| = \text{poly}(|X|), A(X, Y) = “yes”)\]
\[(X \text{ is a “no” instance of } \Pi) \implies (\forall Y : |Y| = \text{poly}(|X|), A(X, Y) = “yes”)\]

We call Y a witness.
Example: Clique

- **Input:** Given graph $G = (V, E)$ and integer $k$.
- **Question:** Does $G$ contain a clique of size $k$?

**Lemma**

*Clique is in NP.*

**Proof.**

1. Suppose the witness $Y$ encodes a set of $k$ nodes in $V$ and $A(G, Y)$ checks if the induced graph on $Y$, $G[Y]$ is a clique.
2. $A$ is a polynomial time algorithm.
3. If there exists a clique of size $k$, there exists $Y$ of size $k$ such that $A(G, Y)$ outputs “yes”
4. If there doesn’t exist a clique of size $k$, there doesn’t exist $Y$ of size $k$ such that $A(G, Y)$ outputs “yes”
Polynomial Time Reduction

Definition
Given two decision problems $\Pi_1, \Pi_2$ we say $\Pi_2$ is polynomial time reducible to $\Pi_1$ iff there exists a polynomial time algorithm $f$ that transforms any instance $X$ of $\Pi_2$ to an instance $f(X)$ of $\Pi_1$ such that:

$$(X \text{ is a “yes” instance of } \Pi_2) \iff (f(X) \text{ is a “yes” instance of } \Pi_1)$$

We write $\Pi_2 \leq_p \Pi_1$ to denote “$\Pi_2$ is polynomial time reducible to $\Pi_1$”.

Easy Example: Independent-Set is polynomial time reducible to Clique
NP-Completeness

Definition
A decision problem $\Pi$ is NP-Hard iff for all $\Pi' \in NP$, $\Pi' \leq_P \Pi$.

Definition
A decision problem $\Pi$ is NP-Complete iff both of the following conditions are satisfied:
1. $\Pi$ is NP-Hard
2. $\Pi \in NP$

Note: If $\Pi$ is NP-Complete and $\Pi \in P$ then $P = NP$

Theorem (Cook 1971)
3-SAT is NP-Complete.
Clique is NP-Complete

**Theorem**

*Clique is NP-Complete*

**Proof.**

1. We’ve already shown Clique $\in NP$
2. Because 3-SAT is NP-complete, it suffices to show 3-SAT $\leq P$ Clique
3. Given formula 3-SAT

$$\phi = (l_{1,1} \lor l_{1,2} \lor l_{1,3}) \land (l_{2,1} \lor l_{2,2} \lor l_{2,3}) \land \ldots \land (l_{m,1} \lor l_{m,2} \lor l_{m,3})$$

in poly-time we can construct $G_{\phi} = (V_{\phi}, E_{\phi})$:

$$V_{\phi} = \{l_{i,j} : i \in [m], j \in [3]\}$$

$$E_{\phi} = \{(l_{i,j}, l_{k,l}) : i, k \in [m], j \in [3], i \neq k, l_{i,j} \neq \overline{l}_{k,l}\}$$

4. **Claim:** $\phi$ is satisfiable iff $G_{\phi}$ has a clique of size $m$
ϕ is satisfiable iff \( G_ϕ \) has a clique of size \( m \)

Suppose \( ϕ \) is satisfiable:

1. In a satisfying assignment, at least one literal is true in each clause
2. Pick one true literal per clause: let \( Y \) be set of corresponding nodes
3. \( G_ϕ[Y] \) is a clique because \( x_k \) and \( \bar{x}_k \) can’t both be in \( Y \) for any \( k \)

Suppose \( G_ϕ \) has a clique of size \( m \):

1. Let \( Y \) be the clique of size \( m \)
2. For each clause:
   - Exactly one node \( l \) from \( i \)-th clause is in \( Y \)
   - Set \( x_k = \text{TRUE} \) if \( l = x_k \) and set \( x_k = \text{FALSE} \) if \( l = \bar{x}_k \)
3. We can’t set \( x_k \) to be true and false because literals \( x_k \) and \( \bar{x}_k \) can’t both be in \( Y \)