Background

- **P**: Class of problems that can be solved in polynomial time
- **NP**: Class of problems that can be solved in non-deterministic polynomial time
- **Y is Polynomial Time Reducible to X**
  - Solve problem Y with a polynomial number of computation steps and a polynomial number of calls to a black box that solves X
  - Notation: \( Y \leq_p X \)
- **Suppose \( Y \leq_p X \). If X can be solved in polynomial time, then Y can be solved in polynomial time**
- **A problem X is NP-complete if**
  - X is in NP
  - For every Y in NP, \( Y \leq_p X \)
  - If X is NP-Complete, Z is in NP and X \( \leq_p Z \)
  - Then Z is NP-Complete

Cook’s Theorem

- The Circuit Satisfiability Problem is NP-Complete

  - Circuit Satisfiability
    - Given a boolean circuit, determine if there is an assignment of boolean values to the input to make the output true

Proof of Cook’s Theorem

- Reduce an arbitrary problem Y in NP to X
- Let A be a non-deterministic polynomial time algorithm for Y
- Convert A to a circuit, so that Y is a Yes instance if and only if the circuit is satisfiable
There are a whole bunch of other important problems which are NP-Complete

Today

Populating the NP-Completeness Universe

- Circuit Sat $\leq$ 3-SAT
- 3-SAT $\leq$ Independent Set
- 3-SAT $\leq$ Vertex Cover
- Independent Set $\leq$ Clique
- 3-SAT $\leq$ Hamiltonian Circuit
- Hamiltonian Circuit $\leq$ Traveling Salesman
- 3-SAT $\leq$ Integer Linear Programming
- 3-SAT $\leq$ Graph Coloring
- 3-SAT $\leq$ Subset Sum
- Subset Sum $\leq$ Scheduling with Release times and deadlines

Satisfiability

Literal: A Boolean variable or its negation.

$X \lor \neg X$

Clause: A disjunction of literals.

$X \lor Y \lor Z$

Conjunctive normal form: A propositional formula $\Phi$ that is the conjunction of clauses.

$C_1 \land C_2 \land \ldots \land C_n$

SAT: Given CNF formula $\Phi$, does it have a satisfying truth assignment?

3-SAT: SAT where each clause contains exactly 3 literals.

Example:

$(x_1 \lor x_2 \lor x_3) \land (x_1 \lor x_2 \lor x_4) \land (x_5 \lor x_2 \lor x_3) \land (x_5 \lor x_2 \lor x_4) \land (x_5 \lor x_2 \lor x_3)$

Yes: $x_1 = true$, $x_2 = true$, $x_3 = false$

Independent Set

- Independent Set
- Graph $G = (V, E)$, a subset $S$ of the vertices is independent if there are no edges between vertices in $S$

3-SAT is NP-Complete

Theorem. 3-SAT is NP-complete.

Proof. Sufficient to show that CIRCUIT-SAT $\leq$ 3-SAT since 3-SAT is in NP.

- Let $K$ be any circuit.
  - Create a 3-SAT variable $X_i$ for each circuit element $i$.
  - Make circuit compute correct values at each node:
    - $x_i = x_j \Rightarrow$ add 2 clauses:
      - $X_i \lor \neg X_j$
      - $X_j \lor \neg X_i$
    - $x_i = x_j \land x_k \Rightarrow$ add 3 clauses:
      - $X_i \lor \neg X_j \lor \neg X_k$
      - $X_j \lor \neg X_i \lor \neg X_k$
      - $X_k \lor \neg X_i \lor \neg X_j$
    - Hard-coded input values and output value:
      - $x_0 = 0 \Rightarrow$ add 1 clause:
        - $\neg X_0$
      - $x_0 = 1 \Rightarrow$ add 1 clause:
        - $X_0$
    - Final step: turn clauses of length < 3 into clauses of length exactly 3.
3 Satisfiability Reduces to Independent Set

Claim: \(3\text{-SAT} \leq \text{INDEPENDENT-SET}\).

Proof: Given an instance \(\Phi\) of 3-SAT, we construct an instance \((G, k)\) of INDEPENDENT-SET that has an independent set of size \(k\) if \(\Phi\) is satisfiable.

Construction:
- \(G\) contains 3 vertices for each clause, one for each literal.
- Connect 3 literals in a clause in a triangle.
- Connect literal to each of its negations.

\[\Phi = (x_1 \lor x_2 \lor x_3) \land (x_1 \lor x_2 \lor x_4) \land (x_1 \lor x_2 \lor x_5)\]

\(k = 3\)

Vertex Cover

- Vertex Cover
  - Graph \(G = (V, E)\), a subset \(S\) of the vertices is a vertex cover if every edge in \(E\) has at least one endpoint in \(S\).

\[
\begin{align*}
&\{1, 2, 3, 4, 5, 6, 7\} \\
&\{1, 2, 3, 4, 5, 6, 7\} \\
&\{1, 2, 3, 4, 5, 6, 7\}
\end{align*}
\]

IS \(\leq_p\) VC

- Lemma: A set \(S\) is independent iff \(V-S\) is a vertex cover

- To reduce IS to VC, we show that we can determine if a graph has an independent set of size \(K\) by testing for a Vertex cover of size \(n - K\).

Clique

- Clique
  - Graph \(G = (V, E)\), a subset \(S\) of the vertices is a clique if there is an edge between every pair of vertices in \(S\).
Complement of a Graph

- Defn: $G' = (V, E')$ is the complement of $G = (V, E)$ if $(u, v)$ is in $E'$ iff $(u, v)$ is not in $E$.

IS $\leq_p$ Clique

- Lemma: $S$ is Independent in $G$ iff $S$ is a Clique in the complement of $G$.
- To reduce IS to Clique, we compute the complement of the graph. The complement has a clique of size $K$ iff the original graph has an independent set of size $K$.

Hamiltonian Circuit Problem

- Hamiltonian Circuit – a simple cycle including all the vertices of the graph.

Thm: Hamiltonian Circuit is NP Complete

- Reduction from 3-SAT

Clause Gadget

- $x_1 \lor x_2 \lor x_3$

Reduce Hamiltonian Circuit to Hamiltonian Path

$G_2$ has a Hamiltonian Path iff $G_1$ has a Hamiltonian Circuit.
Traveling Salesman Problem

- Given a complete graph with edge weights, determine the shortest tour that includes all of the vertices (visit each vertex exactly once, and get back to the starting point).

Thm: HC \textless_p TSP

3-SAT \textless_p 3DM

Matching

- Two dimensional matching
- Three dimensional matching (3DM)

Graph Coloring

- NP-Complete
  - Graph K-coloring
  - Graph 3-coloring
- Polynomial
  - Graph 2-Coloring

3-SAT \textless_p 3DM

Clause gadget for (X OR Y OR Z)

Garbage Collection Gadget
(Many copies)