CSE 421
Algorithms
Richard Anderson
Lecture 27
NP-Completeness

NP Completeness: The story so far

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 iff and only if the circuit is satisfiable
Today

There are a whole bunch of other important problems which are NP-Complete

Satisfiability

Literal: A Boolean variable or its negation. \( x_i \) or \( \overline{x_i} \)
Clause: A disjunction of literals. \( C_i = x_i \lor \overline{x_i} \lor x_j \)

Conjunctive normal form: A propositional formula \( \Phi \) that is the conjunction of clauses.
\( \Phi = C_1 \land C_2 \land C_3 \land C_4 \)

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

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

Example:
\( (\overline{x_2} \lor x_5 \lor x_6) \land (x_1 \lor \overline{x_3} \lor x_4) \land (x_3 \lor \overline{x_5}) \land (\overline{x_4} \lor \overline{x_3}) \)

Yes: \( x_1 = \text{true}, x_2 = \text{true}, x_3 = \text{false} \).

3-SAT is NP-Complete

Theorem. 3-SAT is NP-complete.

Proof. Sufficient to show that CIRCUIT-SAT \( \leq_p \) 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:
  - \( x_i = \overline{x_i} \Rightarrow \text{add 2 clauses}: x_i \lor x_j, x_i \lor x_k \)
  - \( x_i = x_j \Rightarrow \text{add 3 clauses}: x_i \lor x_j, x_j \lor x_k, x_i \lor x_k \)
  - \( x_i = x_j \Rightarrow \text{add 3 clauses}: x_i \lor x_j, x_i \lor x_k, x_i \lor x_k \)
- Hard-coded input values and output value.
  - \( x_0 = \text{false} \Rightarrow \text{add 1 clause}: \overline{x_0} \)
  - \( x_0 = \text{true} \Rightarrow \text{add 1 clause}: x_0 \)
- Final step: turn clauses of length < 3 into clauses of length exactly 3.

3 Satisfiability Reduces to Independent Set

Claim. 3-SAT \( \leq_p \) 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.

\[ \Phi = (x_1 \lor x_2 \lor x_3) \land (x_4 \lor x_5 \lor x_6) \land (x_7 \lor x_8 \lor x_9) \]

k = 3
3 Satisfiability Reduces to Independent Set

Claim. \( G \) contains independent set of size \( k = |\phi| \) if \( \phi \) is satisfiable.

**Pt.** Let \( S \) be independent set of size \( k \).

- \( S \) must contain exactly one vertex in each triangle.
- Set these literals to true.
- Truth assignment is consistent and all clauses are satisfied.

**Pf.** Given satisfying assignment, select one true literal from each triangle. This is an independent set of size \( k \).

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 \( S \).

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

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 $\leq_p$ TSP

Graph Coloring

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