CSE 421
Algorithms
Richard Anderson
Lecture 27
NP-Completeness

NP-Complete
NP
P

I can't find an efficient algorithm, but neither can all these famous people.
NP Completeness: The story so far

Circuit Satisfiability is NP-Complete
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: $Y <_P X$
- **Suppose $Y <_P X$. 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, $Y <_P X$
- If X is **NP-Complete**, Z is in NP and $X <_P Z$
  - 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
Circuit SAT

Find a satisfying assignment
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
Populating the NP-Completeness Universe

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

Literal:  A Boolean variable or its negation.

Clause:  A disjunction of literals.

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

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

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

Ex:  \[(\overline{x_1} \lor x_2 \lor x_3) \land (x_1 \lor \overline{x_2} \lor x_3) \land (x_2 \lor x_3) \land (\overline{x_1} \lor \overline{x_2} \lor \overline{x_3})\]

Yes:  $x_1 = \text{true}$, $x_2 = \text{true}$ $x_3 = \text{false}$.
3-SAT is NP-Complete

Theorem. 3-SAT is NP-complete.

Pf. Suffices 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:
  • $x_2 = \neg x_3 \implies$ add 2 clauses: $x_2 \lor \overline{x_3}$, $\overline{x_2} \lor x_3$
  • $x_1 = x_4 \lor x_5 \implies$ add 3 clauses: $x_1 \lor x_4$, $x_1 \lor x_5$, $x_1 \lor x_4 \lor x_5$
  • $x_0 = x_1 \land x_2 \implies$ add 3 clauses: $\overline{x_0} \lor x_1$, $\overline{x_0} \lor x_2$, $x_0 \lor \overline{x_1} \lor \overline{x_2}$

– Hard-coded input values and output value.
  • $x_5 = 0 \implies$ add 1 clause: $\overline{x_5}$
  • $x_0 = 1 \implies$ add 1 clause: $x_0$

– Final step: turn clauses of length $< 3$ into clauses of length exactly 3. •
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 Satisfiability Reduces to Independent Set

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

Pf. Given an instance \( \Phi \) of 3-SAT, we construct an instance \((G, k)\) of INDEPENDENT-SET that has an independent set of size \( k \) iff \( \Phi \) 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.

\[
\Phi = (\overline{x_1} \lor x_2 \lor x_3) \land (x_1 \lor \overline{x_2} \lor x_3) \land (\overline{x_1} \lor x_2 \lor x_4)
\]
Claim. G contains independent set of size \( k = |\Phi| \) iff \( \Phi \) is satisfiable.

Pf. \( \Rightarrow \) Let \( S \) be independent set of size \( k \).
- \( S \) must contain exactly one vertex in each triangle.
- Set these literals to true. and any other variables in a consistent way
- Truth assignment is consistent and all clauses are satisfied.

Pf \( \Leftarrow \) Given satisfying assignment, select one true literal from each triangle. This is an independent set of size \( k \).
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 $n-K$
Find a maximum independent set $S$

Show that $V - S$ is a vertex cover
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)

Find the minimum cost tour
Thm: HC $\leq_p$ TSP
Graph Coloring

• NP-Complete
  – Graph K-coloring
  – Graph 3-coloring

• Polynomial
  – Graph 2-Coloring