CSE 421
Algorithms

Richard Anderson

Lecture 26

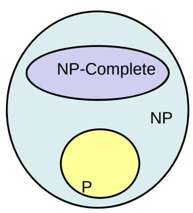
NP-Completeness

Announcements

- Final Exam
  - Monday, March 16, 2:30-4:20 pm
    - Closed book, closed notes
  - Practice final and answer key available
- HW 9, due Friday, March 13, 1:30 pm
- RJA Office Hours, Thursday, 11 am
- This week's topic
  - NP-completeness
  - Reading: 8.1-8.8: Skim the chapter, and pay more attention to particular points emphasized in class
  - It will be on the final

Highlights from Monday

- P and NP
- Certificates
- Decision Problems
- Polynomial time reduction
  - $Y <_P X$
- NP Completeness Definition
  - Z in NP, for all Y in NP, $Y <_P Z$
- NP Completeness Proof:
  - A is NPC, B in NP, $A <_P B$, then B is NPC



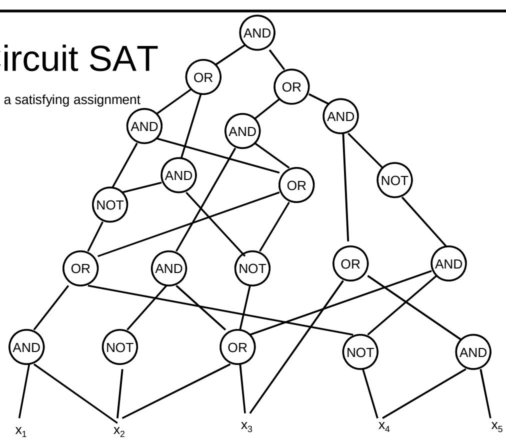
Cook's Theorem

- 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

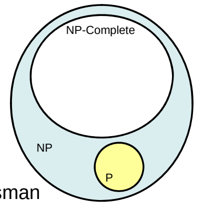
Find a satisfying assignment



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

Populating the NP-Completeness Universe



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

Satisfiability

- Given a boolean formula, does there exist a truth assignment to the variables to make the expression true

Definitions

- Boolean variable: $x_1, \ldots, x_n$
- Term: x_i or its negation $!x_i$
- Clause: disjunction of terms
  - t_1 or t_2 or … t_j
- Problem:
  - Given a collection of clauses $C_1, \ldots, C_k$, does there exist a truth assignment that makes all the clauses true
  - $(x_1$ or $!x_2)$, $(!x_1$ or $!x_3)$, $(x_2$ or $!x_3)$

3-SAT

- Each clause has exactly 3 terms
- Variables $x_1, \ldots, x_n$
- Clauses $C_1, \ldots, C_k$
  - $C_j = (t_{j1}$ or t_{j2} or $t_{j3})$
- Fact: Every instance of SAT can be converted in polynomial time to an equivalent instance of 3-SAT

Find a satisfying truth assignment

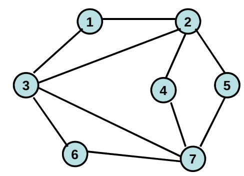
(x || y || z) && (!x || !y || !z) && (!x || y) && (x || !y) && (y || !z) && (!y || z)

Theorem: CircuitSat $<_P$ 3-SAT

Theorem: 3-SAT $<_P$ IndSet

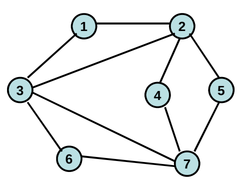
Sample Problems

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



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



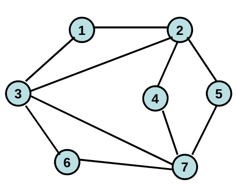
IS $<_P$ VC

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

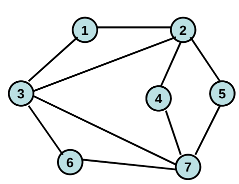
- To reduce IS to VC, we show that we can determine if a graph has an independent set of size K by testing for a Vertex cover of size n - K

IS $<_P$ VC

Find a maximum independent set S

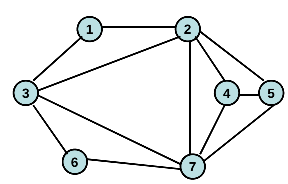


Show that V-S is a vertex cover



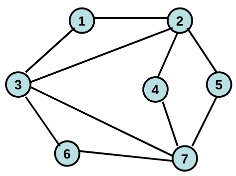
Clique

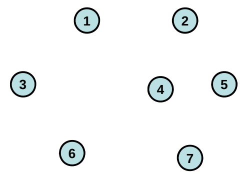
- Clique
  - Graph G = (V, E), a subset S of the vertices is a clique if there is an edge between every pair of vertices in S



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





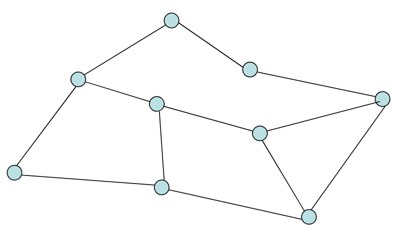
IS $<_P$ Clique

- 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

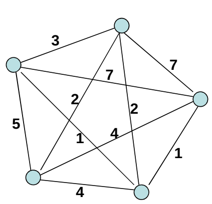


Thm: Hamiltonian Circuit is NP Complete

- 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 $<_P$ TSP