CSE 421 Algorithms

Richard Anderson

Lecture 27

NP Completeness

Announcements

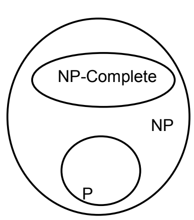
- Final Exam
  - Monday, December 11, 2:30-4:20 pm
    - Closed book, closed notes
  - Practice final and answer key available
- HW 9, due Friday, 1:30 pm
- This week's topic
  - NP-completeness
  - Reading: 8.1-8.8: Skim the chapter, and pay more attention to particular points emphasized in class
 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

Algorithms vs. Lower bounds

- Algorithmic Theory
  - What we can compute
    - I can solve problem X with resources R
  - Proofs are almost always to give an algorithm that meets the resource bounds
- Lower bounds
  - How do we show that something can't be done?

Theory of NP Completeness

The Universe



Polynomial Time

- P: Class of problems that can be solved in polynomial time
  - Corresponds with problems that can be solved efficiently in practice
  - Right class to work with "theoretically"

What is NP?

- Problems solvable in non-deterministic polynomial time . . .
- Problems where “yes” instances have polynomial time checkable certificates

Decision Problems

- Theory developed in terms of yes/no problems
  - Independent set
    - Given a graph G and an integer K, does G have an independent set of size at least K
  - Vertex cover
    - Given a graph G and an integer K, does the graph have a vertex cover of size at most K.

Certificate examples

- Independent set of size K
  - The Independent Set
- Satififiable formula
  - Truth assignment to the variables
- Hamiltonian Circuit Problem
  - A cycle including all of the vertices
- K-coloring a graph
  - Assignment of colors to the vertices

Polynomial time reductions

- 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
  - Notations: $Y <_P X$

Lemma

- Suppose $Y <_P X$. If X can be solved in polynomial time, then Y can be solved in polynomial time.

Lemma

- Suppose $Y <_P X$. If Y cannot be solved in polynomial time, then X cannot be solved in polynomial time.

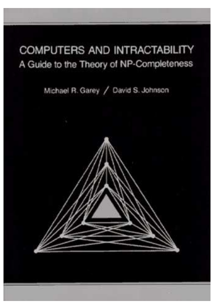
NP-Completeness

- A problem X is NP-complete if
  - X is in NP
  - For every Y in NP, $Y <_P X$
- X is a "hardest" problem in NP
- 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

Garey and Johnson



History

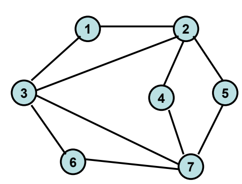
- Jack Edmonds
  - Identified NP
- Steve Cook
  - Cook's Theorem – NP-Completeness
- Dick Karp
  - Identified "standard" collection of NP-Complete Problems
- Leonid Levin
  - Independent discovery of NP-Completeness in USSR

Populating the NP-Completeness Universe

- Circuit Sat $<_P$ 3-SAT
- 3-SAT $<_P$ Independent Set
- Independent Set $<_P$ Vertex Cover
- 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

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