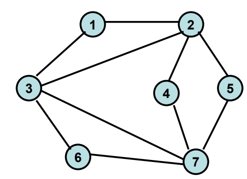
CSE 421 Algorithms

Richard Anderson
Lecture 29
NP-Completeness

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



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

Theorem: 3-SAT $<_P$ IS

- Build a graph that represents the 3-SAT instance
- Vertices y_i, z_i with edges (y_i, z_i)
  - Truth setting
- Vertices u_{j1}, u_{j2}, and u_{j3} with edges (u_{j1}, u_{j2}), (u_{j2}, u_{j3}), (u_{j3}, u_{j1})
  - Truth testing
- Connections between truth setting and truth testing:
  - If $t_{jl} = x_i$, then put in an edge (u_{jl}, z_i)
  - If $t_{jl} = !x_i$, then put in an edge (u_{jl}, y_i)

Example

C_1 = x_1 or x_2 or !x_3

C_2 = x_1 or !x_2 or x_3

C_3 = !x_1 or x_2 or x_3

Thm: 3-SAT instance is satisfiable iff there is an IS of size n + k

What is NP?

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

Certificate examples

- Independent set of size K
  - The Independent Set
- Satifisfiable 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

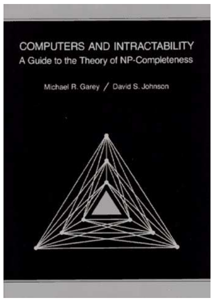
NP-Completeness

- 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

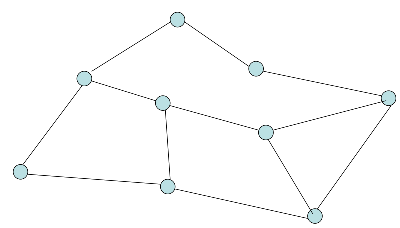
- Jack Edmonds
  - Identified NP
- Steve Cook
  - Cook's Theorem – NP-Completeness
- Dick Karp
  - Identified "standard" collection of NP-Complete Problems
- Leonid Levin
  - Independent discovery of NP-Completeness in USSR

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

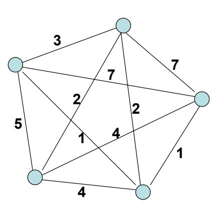
Hamiltonian Circuit Problem

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



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 $<_P$ TSP