CSE 417
Algorithms and Complexity
Autumn 2020
Lecture 28
NP-Completeness
Announcements

• Homework 9, Deadline, Sunday December 13
• Exam practice problems on course homepage
• Final Exam: Monday, December 14
  – 24 hour take home exam
  – Target: 2 to 4 hours of work time
• Approximate grade weighting
  – 75% HW, 25% Final
NP Completeness

I can't find an efficient algorithm, I guess I'm just too dumb.

I can't find an efficient algorithm, but neither can all these famous people.
The Universe

NP-Complete

NP

P
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”

• Decision Problems
  – Theory developed in terms of yes/no problems
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
• Satisfiable 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
Certifiers and Certificates: 3-Satisfiability

SAT: Does a given CNF formula have a satisfying formula

Certificate: An assignment of truth values to the n boolean variables

Certifier: Check that each clause has at least one true literal,

Instance s

\[
\left( x_1 \lor x_2 \lor x_3 \right) \land \left( x_1 \lor x_2 \lor x_3 \right) \land \left( x_1 \lor x_2 \lor x_4 \right) \land \left( x_1 \lor x_3 \lor x_4 \right)
\]

Certificate t

\[x_1 = 1, \ x_2 = 1, \ x_3 = 0, \ x_4 = 1\]
Certifiers and Certificates: Hamiltonian Cycle

HAM-CYCLE. Given an undirected graph $G = (V, E)$, does there exist a simple cycle $C$ that visits every node?

Certificate. A permutation of the $n$ nodes.

Certifier. Check that the permutation contains each node in $V$ exactly once, and that there is an edge between each pair of adjacent nodes in the permutation.
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: \( Y \leq_p X \)
Composability Lemma

- If $X <_P Y$ and $Y <_P Z$ then $X <_P Z$
Lemmas

• Suppose $Y \leq_p X$. If $X$ can be solved in polynomial time, then $Y$ can be solved in polynomial time.

• Suppose $Y \leq_p X$. 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, \( Y \preceq_p X \)

• X is a “hardest” problem in NP

• If X is NP-Complete, Z is in NP and \( X \preceq_p Z \)
  – Then Z is NP-Complete
Cook’s Theorem

- The Circuit Satisfiability Problem is NP-Complete
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
Garey and Johnson

COMPUTERS AND INTRACTABILITY
A Guide to the Theory of NP-Completeness

Michael R. Garey / David S. Johnson
History

Jack Edmonds
- Identified NP

Steve Cook
- Cook’s Theorem – NP-Completeness

Dick Karp
- Identified the “standard” collection of NP-Complete Problems

Leonid Levin
- Independent discovery of NP-Completeness in USSR
P vs. NP Question

P

NP

NP-Complete

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

Abstract: A large class of computational problems involve the determination of properties of graphs, digraphs, integers, arrays of integers, finite families of finite sets, boolean formulas and elements of other countable domains. Through simple encodings from such domains into the set of words over a finite alphabet these problems can be converted into language recognition problems, and we can inquire into their computational complexity. It is reasonable to consider such a problem satisfactorily solved when an algorithm for its solution is found which terminates within a number of steps bounded by a polynomial in the length of the input. We show that a large number of classic unsolved problems of covering, matching, packing, routing, assignment and sequencing are equivalent, in the sense that either each of them possesses a polynomial-bounded algorithm or none of them does.

1. INTRODUCTION

All the general methods presently known for computing the chromatic number of a graph, deciding whether a graph has a Hamilton circuit, or solving a system of linear inequalities in which the variables are constrained to be 0 or 1, require a combinatorial search for which the worst case time requirement grows exponentially with the length of the input. In this paper we give theorems which strongly suggest, but do not imply, that these problems, as well as many others, will remain intractable perpetually.

This research was partially supported by National Science Foundation Grant GJ-474.
Reducibility Among Combinatorial Problems

FIGURE 1 - Complete Problems
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:

\[
\left( \overline{x_1} \lor x_2 \lor x_3 \right) \land \left( x_1 \lor \overline{x_2} \lor x_3 \right) \land \left( x_2 \lor x_3 \right) \land \left( \overline{x_1} \lor \overline{x_2} \lor \overline{x_3} \right)
\]

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:
  
  - \( x_2 = \neg x_3 \Rightarrow \) add 2 clauses: \( x_2 \lor \bar{x}_3, \bar{x}_2 \lor x_3 \)
  
  - \( x_1 = x_4 \lor x_5 \Rightarrow \) add 3 clauses: \( x_1 \lor x_4, x_1 \lor x_5, x_1 \lor x_4 \lor x_5 \)
  
  - \( x_0 = x_1 \land x_2 \Rightarrow \) add 3 clauses: \( \bar{x}_0 \lor x_1, \bar{x}_0 \lor x_2, x_0 \lor \bar{x}_1 \lor \bar{x}_2 \)

- Hard-coded input values and output value.
  
  - \( x_5 = 0 \Rightarrow \) add 1 clause: \( \bar{x}_5 \)
  
  - \( x_0 = 1 \Rightarrow \) 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 $3\text{-SAT}$, we construct an instance $(G, k)$ of $\text{INDEPENDENT-SET}$ 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.

\[
\Phi = (\overline{x_1} \lor x_2 \lor x_3) \land (x_1 \lor \overline{x_2} \lor x_3) \land (\overline{x_1} \lor x_2 \lor x_4)
\]
3 Satisfiability Reduces to Independent Set

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 $k$. □

$$\Phi = (\overline{x_1} \lor x_2 \lor x_3) \land (x_1 \lor \overline{x_2} \lor x_3) \land (\overline{x_1} \lor x_2 \lor x_4)$$
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 \(n - K\)
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
Clause Gadget

\[ x_1 \lor x_2 \lor x_3 \]

X_1 Group

X_2 Group

X_3 Group
Reduce Hamiltonian Circuit to Hamiltonian Path

$G_2$ has a Hamiltonian Path iff $G_1$ has a Hamiltonian Circuit
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
Matching

Two dimensional matching

Three dimensional matching (3DM)
3-SAT $\leq_p$ 3DM

Truth Setting Gadget
3-SAT $\leq_p$ 3DM

Clause gadget for ($\overline{X}$ OR Y OR Z)

Garbage Collection Gadget (Many copies)
Graph Coloring

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

• Polynomial
  – Graph 2-Coloring