CSE 417
Algorithms and Complexity
Autumn 2020
Lecture 29
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
• Possibly some extra office hours
• Approximate grade weighting
  – 75% HW, 25% Final
Today

There are a whole bunch of other important problems which are NP-Complete
Reducibility Among Combinatorial Problems

FIGURE 1 - Complete Problems
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
NP-Completeness Proofs

• Prove that problem X is NP-Complete
• Show that X is in NP (usually easy)
• Pick a known NP complete problem Y
• Show $Y \leq_{p} X$

• Types of NP completeness proofs
  – Easy modifications from other NP complete problems
  – Complicated gadget constructions from 3-SAT
Satisfiability

Literal: A Boolean variable or its negation. $x_i$ or $\overline{x_i}$

Clause: A disjunction of literals. $C_j = x_1 \lor \overline{x_2} \lor x_3$

 Conjunctive normal form: A propositional formula $\Phi$ that is the conjunction of clauses. $\Phi = C_1 \land C_2 \land C_3 \land C_4$

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 = $ true, $x_2 = $ true $x_3 = $ false.
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: $x_2 \lor \neg x_3$, $\neg x_2 \lor x_3$
  - $x_1 = x_4 \lor x_5 \Rightarrow$ 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: $\neg x_0 \lor x_1$, $\neg x_0 \lor x_2$, $\neg x_0 \lor \neg x_1 \lor \neg x_2$

- Hard-coded input values and output value.
  - $x_5 = 0 \Rightarrow$ add 1 clause: $\neg x_5$
  - $x_0 = 1 \Rightarrow$ 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-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.

\[
\Phi = (\overline{x_1} \lor x_2 \lor x_3) \land (x_1 \lor \overline{x_2} \lor x_3) \land (\overline{x_1} \lor x_2 \lor x_4)
\]
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.
   – 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

- 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

$x_1 \lor \overline{x_2} \lor x_3$

$X_1$ 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: $HC \prec_p TSP$
Matching

Two dimensional matching

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

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

Clause gadget for (X OR Y OR Z)

Garbage Collection Gadget
(Many copies)
Graph Coloring

- NP-Complete
  - Graph K-coloring
  - Graph 3-coloring

- Polynomial
  - Graph 2-Coloring