Tractability & Computational Complexity Classes

Plan: Explore the space of computationally hard problems to arrive at a mathematical characterization of a large class of them.

Technique: Compare relative difficulty of different problems.

Loose Definition:

If a problem X is at least as hard as problem Y, it means that if we could solve X, we could also solve Y.

Formal Definition:

$Y \leq_p X$ (Y is polynomial time reducible to X)

if Y can be solved using a polynomial number of standard computational steps plus a polynomial number of calls to a blackbox that solves X.

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

Independent Set

Def.: In a graph $G = (V, E)$, we say that a set of nodes $S \subseteq V$ is "independent" if no two nodes in S are joined by an edge.

Independent Set Problem

- Find the largest independent set in graph G.

Vertex Cover

Def.: In a graph $G = (V, E)$, we say that a set of nodes $S \subseteq V$ is a vertex cover if every edge in E has at least one end in S.

Vertex Cover Problem

- Find the smallest vertex cover set in G.

FACT: Let $G=(V,E)$ be a graph, then S is an independent set if and only if its complement $V-S$ is a vertex cover set.

Proof: A) First suppose that S is an independent set.

Claim: Independent Set $\leq_p$ Vertex Cover

Proof: Having proven that the complement of a vertex cover set is an independent set, given a black box that solves the vertex cover problem, we can decide if G has an independent set of size at least k, by asking the black box

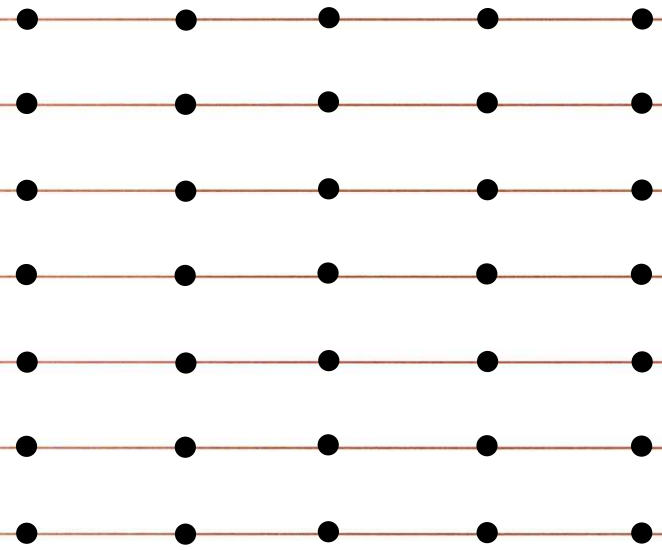
Claim: Vertex Cover $\leq_p$ Independent Set

Proof: Having proven that the complement of an independent set is a vertex cover set, given a black box that solves the indep. set problem, we can decide if G has a vertex cover set of size at most k, by asking the black box

Set Cover Problem

Given a set U of n elements, a collection $S_1, S_2, \ldots, S_m$ of subsets of U, and a number k, does there exist a collection of at most k of these sets whose union is equal to all of U?

Each • represents an element in the set U



Claim: Vertex Cover $\leq_p$ Set Cover

Plan: Given an instance of the vertex cover problem (G, k), we will construct a set of sets such that there is a vertex cover of size k is G iff there are k sets whose union contains all elements of all sets.

Proof:

We need to show that G has a vertex cover of size k, iff the corresponding set cover instance has k sets whose union contains all edges in G.

A) If we have a vertex cover set of size k in G, we can find a collection of k sets whose union contains all edges in G.

Explain how ...

B) If we have k sets whose union contains all edges in G, we can find a vertex cover set of size k in G.

Explain how ...

Reduction Using Gadgets

Some Definitions:

- Given n Boolean variables $x_1 \dots x_n$, a clause is a disjunction of terms $t_1 \vee t_2 \vee \dots \vee t_l$ where $t_i \in \{x_1 \dots x_n, \bar{x}_1 \dots \bar{x}_n\}$

- A truth assignment for X is an assignment of values 0 or 1 to each x_i.

- An assignment satisfies a clause C if it cause C to evaluate to 1.

- An assignment satisfies a collection of clauses if $C_1 \wedge C_2 \ldots C_k$ evaluates to 1.

Problem Statement

Given a set of clauses $C_1 \ldots C_k$ over a set of variables $X = \{x_1, \ldots x_n\}$, does there exist a satisfying truth assignment?

Problem Statement

Given a set of clauses $C_1 \ldots C_k$ each of length 3 over a set of variables $X = \{x_1, \ldots x_n\}$, does there exist a satisfying truth assignment?

Claim: 3-SAT $\leq_p$ Independent Set

Plan: Given an instance of 3-SAT with k clauses, we will construct a graph G that has an indep. set of size k iff the 3-SAT instance is satisfiable.

Claim: The 3_SAT instance is satisfiable iff the graph G has an independent set of size k.

Proof: A) If the 3_SAT instance is satisfiable, then there is at least one node label per triangle that evaluates to 1.
Let S be a set containing one such node from each triangle

B) Suppose G has an independent set S of size at least k.

If x_i appears as a label in S, then

If $\bar{x}_i$ appears as a label in S, then

If neither x_i nor $\bar{x}_i$ appear as a label in S, then

Efficient Certification

A problem has efficient certification if given a solution, the correctness of the solutions can be confirmed in polynomial time.

To show efficient certification, we need

1_ Polynomial length certificate

2_ Polynomial time certifier

Showing Efficient Certification

3-SAT:

Certificate $\underline{t}$ is an assignment of truth values to variables (x_i)

Certifier: Evaluate the clauses. If all of them evaluate to 1, then it answers "yes".

Independent Set:

Certificate $\underline{t}$ is a set of nodes of size at least $\underline{k}$ in G.

Certifier: Check each edge to make sure no edges have both ends in the set.
Check size of the set to be $\geq k$
Check that there no repeating nodes in the set.

Def.: Class NP is the set of all decision problems for which there exists an efficient certifier.

All problems in P are in NP

Are all problems in NP in P ?
i.e. is P = NP ?

If $X \in NP$ and for all $Y \in NP$ $Y \leq_p X$, then X is the hardest problem in NP.

Transitivity in polyn. time reduction

if $Z \leq_p Y$ and $Y \leq_p X$, then $Z \leq_p X$

Basic strategy to prove a problem X is NP-Complete:

Def.: NP-hard is the class of problems that are at least as hard as NP-Complete problems.

1. Given the SAT problem from lecture for a Boolean expression in Conjunctive Normal Form with any number of clauses and any number of literals in each clause. For example,

$$(X_1 \vee \neg X_3) \wedge (X_1 \vee \neg X_2 \vee X_4 \vee X_5) \wedge \ldots$$

Prove that SAT is polynomial time reducible to the 3-SAT problem (in which each clause contains at most 3 literals.)

2. The *Set Packing* problem is as follows. We are given m sets $S_1, S_2, \ldots S_m$ and an integer k. Our goal is to select k of the m sets such that no selected pair have any elements in common. Prove that this problem is **NP**-complete.

3. The *Steiner Tree* problem is as follows. Given an undirected graph $G=(V,E)$ with nonnegative edge costs and whose vertices are partitioned into two sets, R and S, find a tree $T \subseteq G$ such that for every v in R, v is in T with total cost at most C. That is, the tree that contains every vertex in R (and possibly some in S) with a total edge cost of at most C.
Prove that this problem is **NP**-complete.