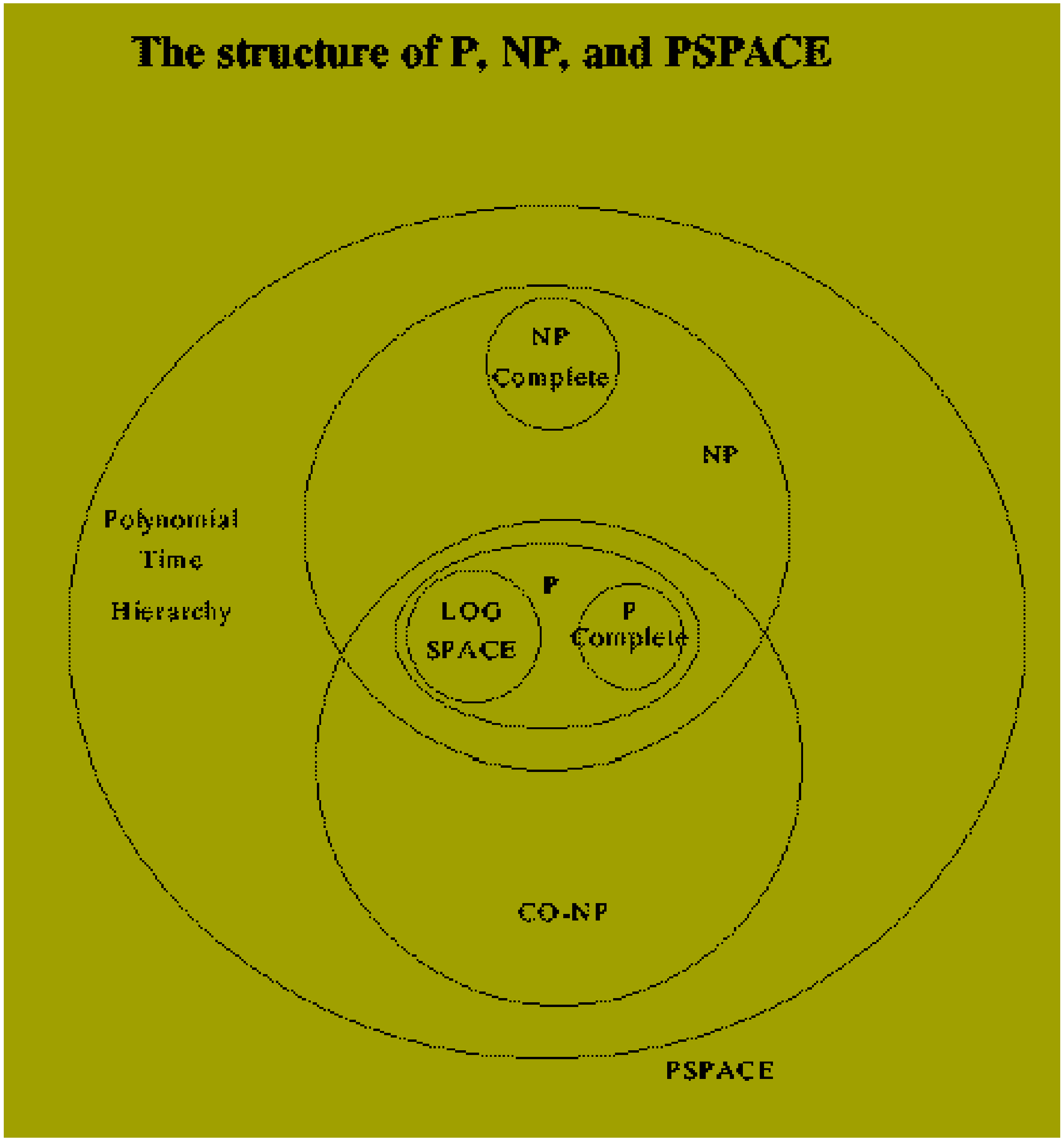
A Brief Review of NP-Completeness



Prof. Santosh Kumar
Dept. of Computer Science
University of Memphis
Fall 2008

The Idea

- Look at the following cartoon:

http://max.cs.kzoo.edu/~kschultz/CS510/ClassPresentations/NPCartoons.html

A Class of Problems

- All possible instances of the problem
- The algorithm you design should be able to solve any instance of the problem
- Hence, a need to prove the correctness (and/or optimality)
- The complexity quoted is usually worst-case

An Example – Vertex Cover

- Problem
  - Given
    - An undirected graph *G = (V,E)*
    - A cost function on vertices $c : V \rightarrow Q^+$
  - Find
    - A minimum cost *vertex cover*, i.e., a set $V' \subseteq V$ such that every edge in *E* has at least one end point in V'.
  - If $\forall v \in V, c(v) = 1$, then the problem is referred to as the *cardinality vertex cover* problem.

The Art of Reduction

- Convert one problem instance to another in polynomial-time
  - A scheduling problem to a flow problem
  - A coverage problem to a satisfiability problem
- What's the use?
  - Can use an algorithm designed for one problem class to solve another problem class
- Reduction requires proving an ***if and only if*** relation

Reduction for Decision Problems

- Problem class ***A*** reduces to problem class ***B*** if
  - Given an instance of ***A*** an instance of ***B*** can be constructed in polynomial time
  - If the answer to the ***B*** instance is **no**, then answer to the instance of ***A*** must also be **no**
  - Similar relation should hold for the answer **yes**
- Typical Claim: *Problem instance of* **B** *produces a "yes" answer iff the corresponding instance of* **A** *has a "yes" answer.*

Reduction for Optimization Problems

- Problem class ***A*** (with objective *f(x)*) reduces to problem class ***B*** (with objective *g(y)*) if
  - Given an instance of ***A**,* an instance of ***B*** can be constructed in polynomial time
- Typical Claim: *Given an instance of **A**, an objective value of g(y) is achieved iff the corresponding instance of **A** achieves an objective value of f(x) = h(g(y)).*
  - Given an instance y *of* **A**, construct an instance *x of* ***B***, such that *f(x)* = *h(g(y))* and vice versa

What if the Problem is NP-Hard?

- First establish that your problem is NP-Hard by reducing a known NP-Complete problem to it
  - This is the other way round
  - Your claim is that if someone were to solve my problem efficiently, then they will have solved all known NP-Complete problems that have remained open

Proving NP-Completeness

- For a given problem class Π,
  1. Show $\Pi \in$ NP
  2. Select a known NPC problem, say Δ
  3. Construct a reduction (*iff*) from Δ to Π and prove both directions of *iff*
     - Come up with a rule s.t. every problem instance of Δ should transform to some instance of Π
  4. Prove that the transformation (the steps in your rule) is polynomial-time

Examples

- Dominating Set
- Vertex Cover
- Independent Set
- See http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15451-f99/www/lectures/lect1019

Dominating Set

- Problem
  - Given
    - An undirected graph $G = (V,E)$
    - A positive integer $K \leq |V|$
  - Find
    - Whether there exists a *dominating set* of size $\leq K$
      - A subset $D \subseteq V$ is said to be a *dominating set*, if every vertex in $V \backslash D$ is a direct edge with a vertex in D
  - What is correctness here?
  - And, performance?

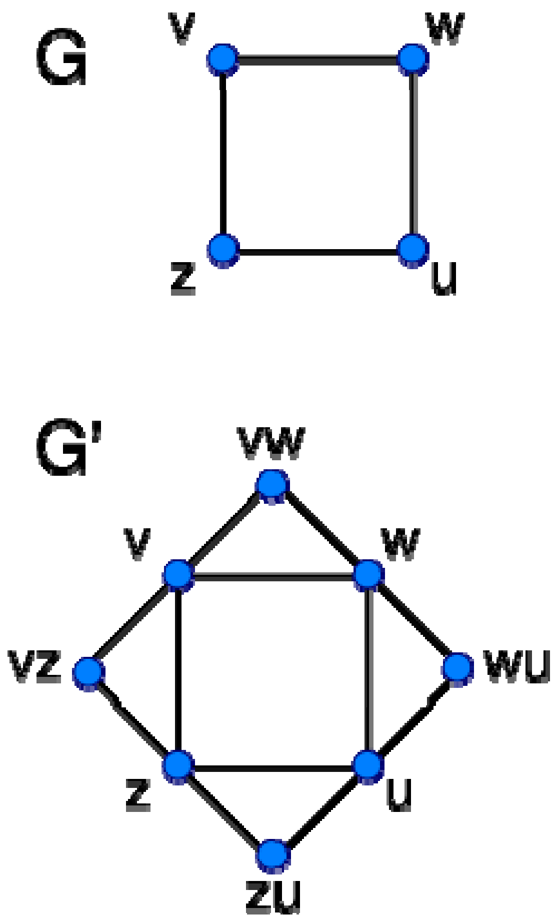
Proving NP Completeness

- Why is it in *NP*?
- Which *NPC* problem do reduce from?
- Construct a polynomial time reduction
- Prove that it is indeed a reduction

Vertex Cover

- Problem
  - Given
    - An undirected graph $G = (V,E)$
    - A positive integer $K \leq |V|$
  - Find
    - Whether there exists a *vertex cover* of size $\leq K$
      - A subset $C \subseteq V$ is said to be a *vertex cover* if every edge in E has at least one end point in C'.

The Reduction



Correctness of Reduction

- Assume *C* is a vertex cover in *G*
  - Prove that *C* will also be a dominating set in the transformed graph *G*'
- Assume *D* is a dominating set in a transformed graph *G*'
  - Prove that in the original graph *G*, *D* can be converted to a vertex cover preserving its cardinality

NP Completeness of Vertex Cover

- Can use Independent Set
  - An independent set in a graph is a set of nodes no two of which have an edge.
- Key Claims
  - If C is a vertex cover in G, then $V \setminus C$ is an independent set in G
  - Conversely, if S is an independent set, then $V \setminus S$ is a vertex cover in G
- What else needs to be proved?

NP Completeness of Independent Set

- Can use *k*-Clique
  - *k*-Clique: Does *G* have a clique of size ≤ *k*
- Key Claim:
  - A graph *G* has a clique of size *k* if and only if its complement has an independent set of size *k*

Beyond NP-Completeness Proof

- Proving NP-Completeness is only the beginning
  - Find special cases that can be solved optimally
    - Follow the steps outlined previously or other techniques
  - Design approximation algorithms for the general case
    - Prove approximation bounds (how?)
    - Establish that the approximation factor established is the best possible for your algorithm
      - Come up with some examples where the approximation factor is tight
    - Establish that your approximation algorithm is the best possible for the bounding scheme being used
    - Establish that no better approximation algorithm is possible.

Your First Presentation (10/1)

- Select a known NP-Complete (or Hard) problem
- Come up with at least two networking applications for it
- Proof of its NP-Completeness