# Introduction To NP-Completeness

Guest Lecture for CS 350

Bart Massey <bart@cs.pdx.edu>
PSU CS Department
February 19, 2001

## Overview

- Computational Complexity
- NP-Completeness
- Applications

## Order Analysis and Computational Complexity

- Motivation: estimate problem difficulty
- Two approximations: Consider only
  - *growth rate* of difficulty with *instance size*
  - *polynomial part* of growth rate
- Both approximations questionable
  - cryptography: difficulty of fixed-size instances
  - linear-time register allocation: huge constants

## Two Theses

**Polytime Thesis**: Any realistic $O(n^k)$ problem is $O(n^3)$ problem

**Church-Turing Thesis**: The same problems are $O(n^k)$ problems on any computer (QP? Open)

## Problem Descriptions

Need uniform notation for

- Distinguishing problem from class
- Distinguishing instance from problem
- Formulating size of instance

## Problem Description Notation [Garey-Johnson]

Key Elements:

- Name: Identifies problem
- Instance: Lists all data comprising problem instance
- Question: Formulates yes/no "decision" question

Example:

**EVEN SET**

INSTANCE: A set $S$ of integers.

QUESTION: Are all integers in $S$ even?

## Instance Size

Size of instance is minimum number of bits needed to represent instance.

What is size of EVEN SET instance?

What about EVEN ELEMENT?

## Problems And Classes

Problem $p$ is *in* class $\mathcal{C}$ when exists $\mathcal{C}$ algorithm for solving $p$ instances.

$p$ is *hard for* $\mathcal{C}$ when $\mathcal{C}$ algorithm for $p$ also solves all instances in $\mathcal{C}$.

$p$ is *complete for* $\mathcal{C}$ when $p$ is in $\mathcal{C}$ and $\mathcal{C}$ hard.

## The Class NP

Decision problem is in P if can answer yes/no in polytime.

Consider class of problems that can *check* yes answer in polytime.

- Same? No one knows
- Call this class NP

## Nondeterministic Polynomial

Why NP? Because

- If you *guess* an answer (nondeterminism)
- You can *check* it in polytime

E.g. SAT, graph coloring, bin packing

$$P \stackrel{?}{=} NP$$

- Most think $P \neq NP$
- After many years
  - no proof
  - no algorithm
- Assume $P \neq NP$ for this lecture

## Decision vs. Optimization Problems

- Many problems call for value, not decision
- Optimization problems call for best value
- Trick
  - Make optimization target part of instance
  - Binary search

# From Optimization To Decision

**MAXIMAL BOUNDED SUBSET CONSTRUCTION**

INSTANCE: Set $A$ of positive integers, bound $B$

QUESTION: What is the largest subset $A'$ of $A$ such that

$$s = \sum_{e \in A'} e \leq B$$

**MAXIMAL BOUNDED SUBSET**

INSTANCE: Set $A$ of positive integers, bound $B$, target $K$.

QUESTION: Is there a subset $A'$ of $A$ such that

$$s = \sum_{e \in A'} e \leq B$$

and $s \geq K$?

## Coclasses and Coproblems

- Note: NP decision problem is P check for ‘yes’ answer
- P check for ‘no’ answer?
  - These are co-NP problems
  - E.g. UNSAT, No-Coloring, No-Packing
  - Believed harder than NP
  - But $NP \stackrel{?}{=} co - NP$ open
  - Note $P = co - P$, so $P = NP$ would imply $NP = co - NP$

## NP Complete

A problem is NP Complete if it is

- In NP: easy to check, but important
- Hard for NP: how to tell?

## Many-One Reductions

If you have an NP-hard problem $p$, another problem $q$ is NP-hard if

- Each $p$ instance transformable to $q$ instance in polytime
- $q$ instance yes exactly when $p$ instance yes

If $q$ is polytime solvable and $p \leq_m q$ ($p$ is reducible to $q$)

- $p$ is polytime solvable
- All NPC problems polytime solvable

## Many-One Reduction: Integer Knapsack

**SUBSET SUM**

INSTANCE: Set $A$ of positive integers, target $K$.

QUESTION: Is there a subset $A'$ of $A$ such that

$$\sum_{e \in A'} e = K$$

- SUBSET SUM is NPC
- Can reduce SUBSET SUM to MBS
- Therefore MBS is NPC

## SAT Is NP Complete

Can prove problems NP hard (thus NPC) by many-one reduction

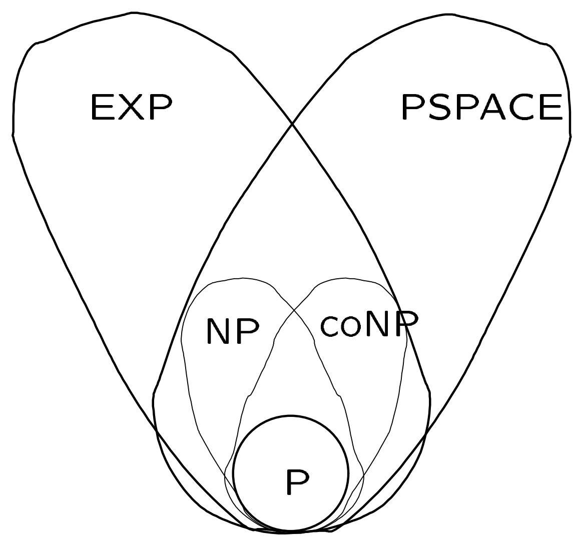
But need base case

- CLR: CIRCUIT-SAT is NP-hard by definition
- Cook's Theorem: SAT is NP-hard by definition

Proof Idea: Construct circuit (resp. formula) for "Nondeterministic Turing Machine." Show any NP-hard problem polytime-solvable by NTM

## P, NP, and Beyond

Consider P, NP, co-NP. Can define more complex classes (w/ co-classes): EXP, PSPACE, oracle classes. Little known: mainly $P \subset EXP$



## Reminder: Why We Care

Big excursion into theory. Why?

- Should do one of
  - Give P algorithm for your problem
  - Prove your problem NPC
  - Prove your problem above NP

## Techniques For NP Hardness Proof

- Restriction: $q$ contains NP-hard $p$ as special case
  - e.g. MBS restricts to SUBSET SUM
- local replacement
- Component design
- Direct proof (never)

## Instance Generalization

- Often given single instance: constant time!
- Still interested in instance “hardness”
- Generalize instance and find problem class
  - No “right” generalization
  - Answers wrong question
  - Still very useful

# LOGS BOX STACKING

Given instance with

- 14 logs of given length
- 3-D box of given length

formulate

## LOGS BOX STACKING

INSTANCE: $n$ logs of length $l_1 \ldots l_n$ and unit width and height, 3-dimensional box with sides $d_1$, $d_2$, $d_3$ and $\prod_i d_i = \sum_j l_j$ and

$$\exists q \in \mathcal{R} \ . \ \forall i \ . \ l_i | q$$

QUESTION: Is there a packing of the logs into the box?

## Restriction To 1D

### LOGS BOX STACKING

INSTANCE: Set L of $n$ logs of integer length $l_1 \ldots l_n$ and unit width, 2-dimensional box with sides 2 and $d$ such that $2d = \sum_j l_j$.

QUESTION: Is there a packing of the logs into the box?

## Proving LOGS BOX STACKING NPC

Two things to prove

- LOGS BOX STACKING in NP? Yes
- LOGS BOX STACKING NP-hard? Yes, by reduction from PARTITION [G&J SP12]

**PARTITION**

INSTANCE: Finite set $A$, a size $s(a) \in \mathcal{Z}^+$ for each $a \in A$.

QUESTION: Is there a subset $A' \subseteq A$ such that

$$\sum_{a \in A'} s(a) = \sum_{a \in A - A'} s(a)$$

## NP Hardness Proof For LOGS BOX STACKING

Consider PARTITION instance with $A$. Make LOGS BOX STACKING instance with $2 \times \sum A/2$ box and logs from $A$.