W04

NP-Completeness

Introduction

Polynomial-time: Tractable, or easy

Examples

Shortest vs. Longest simple paths

Euler tour vs. Hamiltonian cycle

2-CNF satisfiability vs. 3-CNF satisfiability

P, NP, NPC, NP-Hard

P: lovable in polynomial time

NP: verifiable in polynomial time

Nondeterministic polynomial time

NPC: NP and as hard as any problem in NP

NPC Key concepts

Decision problems vs. optimization problems

Reductions

A first NPC problem

Polynomial time

Why?

Few practical problems with extreme high order

Improvement comes soon

Cross models

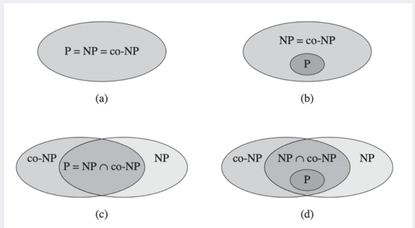
Closure properties

Abstract

Encodings

A formal-languages framework

Polynomial time verification



NP-completeness and reducibility

Reducibility

Not more than a polynomial factor harder

NPC

A language $L \subseteq \{0,1\}^*$ is ***NP-complete*** if

1. $L \in$ NP, and
2. $L' \leq_P L$ for every $L' \in$ NP.

Circuit satisfiability

Given a boolean combinational circuit composed of AND OR NOT gates, is it satisfiable?

NPC proofs

1. Prove $L \in$ NP.
2. Select a known NP-complete language L'.
3. Describe an algorithm that computes a function f mapping every instance $x \in \{0,1\}^*$ of L' to an instance $f(x)$ of L.
4. Prove that the function f satisfies $x \in L'$ if and only if $f(x) \in L$ for all $x \in \{0,1\}^*$.
5. Prove that the algorithm computing f runs in polynomial time.

Formula satisfiability

3-CNF satisfiability

DNF

NPC problems