NP-Completeness

Andreas Klappenecker

[partially based on slides by Jennifer Welch]
Definition of NP-Complete

L is NP-complete if and only if
(1) L is in NP and
(2) for all L' in NP, \( L' \leq_p L \).

In other words, L is at least as hard as every language in NP.
Implication of NP-Completeness

**Theorem** Suppose L is NP-complete.

(a) If there is a poly time algorithm for L, then $P = NP$.

(b) If there is no poly time algorithm for L, then there is no poly time algorithm for any NP-complete language.
Proving NP-Completeness

(a) Use a direct approach and prove that

(1) L is in NP

(2) every other language in NP is polynomially reducible to L

(b) Find an NP-complete problem and use reduction.

Approach (a) is for larger-than-life people, (b) is for mere mortals.
Proving NP-Completeness by Reduction

To show \( L \) is NP-complete:

1. Show \( L \) is in NP.

2.a) Choose an appropriate known NP-complete language \( L' \).

2.b) Show \( L' \leq_p L \).

This works, since every language \( L'' \) in NP is polynomially reducible to \( L' \), and \( L' \leq_p L \). By transitivity, \( L'' \leq_p L \).
SAT
First NP-Complete Problem

How do we get started? Need to show via brute force that some problem is NP-complete.

• Logic problem "satisfiability" (or SAT).

• Given a boolean expression (collection of boolean variables connected with ANDs and ORs), is it satisfiable, i.e., is there a way to assign truth values to the variables so that the expression evaluates to TRUE?
Conjunctive Normal Form (CNF)

**Boolean variable**: Indeterminate with values T or F. Example: x, y

**Literal**: Variable or negation of a variable. Example: x, ¬x

**Clause**: Disjunction (OR) of several literals. Example: x ∨ ¬y ∨ z ∨ w

**CNF formula**: Conjunction (AND) of several clauses. Example: (x ∨ y) ∧ (z ∨ ¬w ∨ ¬x)
Satisfiable CNF Formula

- Is \((x \lor \neg y)\) satisfiable?
  - yes: set \(x = T\) and \(y = F\) to get overall \(T\)
- Is \(x \land \neg x\) satisfiable?
  - no: both \(x = T\) and \(x = F\) result in overall \(F\)
- Is \((x \lor y) \land (z \lor w \lor x)\) satisfiable?
  - yes: \(x = T\), \(y = T\), \(z = F\), \(w = T\) result in overall \(T\)
- If formula has \(n\) variables, then there are \(2^n\) different truth assignments.
Definition of SAT

SAT = all (and only) strings that encode satisfiable CNF formulas.
SAT is NP-Complete

• Cook's Theorem: SAT is NP-complete.

• Proof ideas:

• (1) SAT is in NP: Given a candidate solution (a truth assignment) for a CNF formula, verify in polynomial time (by plugging in the truth values and evaluating the expression) whether it satisfies the formula (makes it true).
SAT is NP-Complete

• How to show that every language in NP is polynomially reducible to SAT?

• Key idea: the common thread among all the languages in NP is that each one is solved by some nondeterministic Turing machine (a formal model of computation) in polynomial time.

• Given a description of a poly time TM, construct in poly time, a CNF formula that simulates the computation of the TM.