Solving Some NP-complete Problem Instances by Reductions

Aleksandar Zeljić
Predrag Janičić

4^{th} Workshop on Formal and Automated Theorem Proving and Aplications

Contents

Motivation

- Not many NP-complete problems have tuned solvers.
- Reductions give a way of solving various problems with a single problem-specific solver.
- While reductions typically require polynomial time, their use rarely yields practically usable solutions.
- Use of reductions could produce solvers that might be better suited for some problem instances.

Research goal

- Discover classes of some NP-complete problem instances that can be practically solved by reduction to another NP-complete problem.
- If several such classes were discovered it should be possible to automatically determine which reduction should be applied.
- It would also be interesting to determine if "the hardest (random) instances" of an NP-complete problem remain "the hardest instances" after reduction.
- So far our focus was on SAT and k-clique problems.

Reduction of SAT problem to k-clique problem

- Karp, 1972. "Reducibility among combinatorial problems"
- Input is a propositional formula in CNF.
- For each literal a node is created, marked with the literal and clause number it belongs to.
- An edge is added between every two nodes that don't belong to the same clause and aren't marked with a literal and its negation.
- Resulting graph now contains a clique with size equal to the number of clauses in the formula iff input formula is SAT.

Reduction of k-clique problem to SAT problem

- Input is a graph with n nodes and number k representing the size of clique.
- Nodes in graph are marked by numbers $1, \dots, n$. In order to represent one node we will use $\lceil \log(n-1) \rceil$ boolean variables.
- There are several approaches to represent k-clique, the simplest being an array of k nodes, so for representation of k-clique we need a total of $k * \lceil \log(n-1) \rceil$ boolean variables.

Reduction of problem k-clique to problem SAT

- Clauses are added to the formula to ensure the following conditions are met:
    - Values assigned to the to nodes must represent an existing node.
    - No node identifier may be repeated among the k members of the clique or it wouldn't be a k-clique.
    - Every edge that doesn't exist in graph makes a pair of identifiers that must not appear among clique nodes.
- Resulting formula is SAT iff a k-clique exists in the graph.

Preliminary observations

- Reduction to k-clique problem gives very poor results for SAT instances.
- Reductions to SAT give results comparable to k-clique solver.