

On Exploiting Structures for Constraint Solving

Sathiamoorthy Subbarayan

January 31, 2008

Abstract

The constraint satisfaction problem (CSP) instances modeling problems occurring in practice often have structures which can be exploited to speed-up the constraint solving process. This thesis explores the exploitation of three such structures.

The three structures explored in this thesis are:

1. *Tree-like relationships*: Several CSP instances have tree-like relationships between the variables in them. Such a relationship can be observed by using the tree-width parameter. Given a CSP instance with low tree-width, implying close to tree-like relationship, a tree-decomposition of the instance could be used to speed-up the constraint solving process.
2. *Compact BDD representation*: For some finite-domain constraint problem instances, the solutions of each constraint in those instances might be represented by a compact Binary Decision Diagram (BDD). In which case, it is advantageous to represent the constraints using BDDs. This might lead to speeding-up the propagation and other reasoning steps in a constraint solver.
3. *Abundance of infrequent variables*: In some constraint problem instances, the modeling process might introduce a lot of variables to simplify the encoding of a problem instance. Such introduction of new variables might lead to abundance of variables occurring in only a few constraints. In such cases, the existential quantification of those infrequent variables in a preprocessing step might reduce the effort required to determine the satisfiability of the instance.

This thesis explores toward exploiting these three ideas on several problem instances. Suggestions are made toward incorporation of these ideas and those suggestions are supported by empirical data. Since these kind of structures won't be found in randomly generated CSP instances, the experiments in this thesis were done on instances either mostly from practice or on those close to those occurring in practice. For most of the experiments reported in this thesis, the source code or the executable code used, along with the input data are publicly available for further research.

Some highlights of the results in this thesis are:

1. Using the tree-decomposition of configuration problems lead to up to 96% reduction in space required for BDD-based interactive configuration, while still guaranteeing reasonable response time.
2. Exporting the idea of mixing tree-decomposition with BDDs to the *knowledge compilation* tasks lead to up to 1000 times lesser space usage than the earlier methods.
3. A new linear-time method for finding a *minimal-reason* for a variable implication by a BDD, while the earlier method in the worst case requires at least quadratic-time.
4. Existential quantification (by using variable elimination resolution) of infrequent variables in CNF SAT problem instances lead to a significant reduction in the time taken for solving several CNF SAT instances. Subsequently, an extension of this idea has been used in the winning SAT solvers of both the last two SAT solver competitions (during 2005 and 2007).
5. Finally, a new method for *hypertree decomposition* lead to significant improvement over the earlier methods for hypertree decomposition of CSPs.