Combining inference and search for the propositional satisfiability problem[*]

Lyndon Drake and **Alan Frisch**
{lyndon,frisch}@cs.york.ac.uk
Computer Science Department
University of York
Heslington, York YO10 5DD
United Kingdom

Toby Walsh
tw@4c.ucc.ie
Cork Constraint Computation Centre
University College Cork
Cork
Ireland

Abstract

The most effective complete method for testing propositional satisfiability (SAT) is backtracking search. Recent research suggests that adding more inference to SAT search procedures can improve their performance. This paper presents two ways to combine neighbour resolution (one such inference technique) with search.

Introduction

Davis-Putnam (DP) (Davis & Putnam 1960) was the first practical complete algorithm for solving propositional satisfiability (SAT) problems. DP uses resolution to determine whether a SAT problem instance is satisfiable. However, resolution is generally impractical, as it can use exponential space and time. The most important refinement to DP was DLL (Davis, Logemann, & Loveland 1962), which replaced the resolution in DP with backtracking search. Backtracking search still uses exponential time in the worst case, but only needs linear space. As time is more readily available than space, the use of search was essential.

Since then, the DLL algorithm has been used almost exclusively in complete SAT solvers (Gu *et al.* 1997). However, Rish and Dechter (2000) recently showed that a hybrid complete solver which used ordered resolution along with backtracking search often outperformed pure DLL. Van Gelder and Tsuji (1995) have also shown that another hybrid resolution and search method, `2cl`, outperforms standard DLL. Cha and Iwama (1996) separately described a local search algorithm that used resolution between similar (or neighbouring) clauses to improve performance. We have investigated the use of this neighbour resolution in a complete DLL-based SAT solver.

Neighbour resolution with complete search

We have completed two implementations of neighbour resolution:

During search. In the first, neighbour resolution was carried out during search. This method was not cost-effective: while it sometimes pruned the search tree by one or two orders of magnitude, the time cost of doing the resolutions outweighed the benefits to the search algorithm.

The time cost is primarily due to the expensive algorithm used to identify possible neighbour resolutions during search. We have designed an incrementally updated data structure that we plan to use in an improved implementation.

Before search. The second method emulates neighbour resolution by using binary resolution as a preprocessing step. Preliminary results show that on many problems, emulating neighbour resolution before search provides substantial improvements in performance over pure DLL, both in the number of search nodes explored and in the runtime used.

The emulation of neighbour resolution during search is imperfect. We plan to make the preprocessing implementation a closer emulation by reducing the number of resolvents generated, and by checking subsumptions due to resolutions.

References

Cha, B., and Iwama, K. 1996. Adding new clauses for faster local search. In *Proceedings of AAAI-96*, 332–337.

Davis, M., and Putnam, H. 1960. A computing procedure for quantification theory. *Journal of the ACM* 7:201–215.

Davis, M.; Logemann, G.; and Loveland, D. 1962. A machine program for theorem-proving. *Communications of the ACM* 5:394–397.

Gu, J.; Purdon, P. W.; Franco, J.; and Wah, B. W. 1997. Algorithms for the satisfiability (SAT) problem: A survey. In *Satisfiability Problem: Theory and Applications*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society. 19–152.

Rish, I., and Dechter, R. 2000. Resolution versus search: Two strategies for SAT. In Gent, I.; van Maaren, H.; and Walsh, T., eds., *SAT2000: Highlights of Satisfiability Research in the Year 2000*, volume 63 of *Frontiers in Artificial Intelligence and Applications*. IOS Press. 215–259.

van Gelder, A. 1995. Satisfiability testing with more reasoning and less guessing. In Johnson, D. S., and Trick, M., eds., *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society.

[*] This research is supported by EPSRC grant GR/N16129.