

Automatic Generation of Implied Clauses for SAT

Lyndon Drake, Alan Frisch, and Toby Walsh

Artificial Intelligence Group, Department of Computer Science, University of York,
York YO10 5DD, United Kingdom,
{lyndon,frisch,tw}@cs.york.ac.uk,
<http://www.cs.york.ac.uk/~{lyndon,frisch,tw}>

Davis-Putnam (DP) [3] 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 [2], 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 change to search was a big improvement.

Since then, the DLL algorithm has been used almost exclusively in complete SAT solvers [4]. However, Rish and Dechter [5] recently showed that a hybrid complete solver which used ordered resolution along with backtracking search often outperformed pure DLL. Cha and Iwama [1] separately described a local search algorithm that used resolution between similar (or neighbouring) clauses to improve performance. We have investigated the use of this neighbourhood resolution in a complete SAT solver.

Preliminary results show that on certain problems, using neighbourhood resolution in conjunction with search can provide substantial improvements in performance over pure DLL, both in the number of search nodes explored and in the runtime used. Further work on neighbourhood resolution is planned to improve its performance and to identify suitable problem classes.

References

1. Byungki Cha and Kazuo Iwama. Adding new clauses for faster local search. In *Proceedings of AAAI-96*, pages 332–337, 1996.
2. Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5:394–397, 1962.
3. Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7:201–215, 1960.
4. Jun Gu, Paul W. Purdon, John Franco, and Benjamin W. Wah. Algorithms for the satisfiability (SAT) problem: A survey. In *Satisfiability Problem: Theory and Applications*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, pages 19–152. American Mathematical Society, 1997.
5. Irina Rish and Rina Dechter. Resolution versus search: Two strategies for SAT. In Ian Gent, Hans van Maaren, and Toby Walsh, editors, *SAT2000: Highlights of Satisfiability Research in the Year 2000*, volume 63 of *Frontiers in Artificial Intelligence and Applications*, pages 215–259. IOS Press, 2000.