

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}>

Propositional satisfiability (SAT) is the archetypal NP-complete problem [1]. A great deal of recent SAT research, particularly on the performance of SAT solvers, has been driven by structured instances, which are obtained by mapping another problem class into SAT. It is possible to efficiently solve a wide range of problems by mapping them into SAT and solving the SAT representation of the problem.

Currently, the most effective complete method for solving SAT instances is backtracking search. Logical inference can be used to reduce the size of the search space a SAT solver must explore. Unfortunately, inference often consumes so much time and memory that these costs outweigh the benefit gained by pruning the search space. The challenge is to limit this overhead while still pruning enough of the search space to make the inference worthwhile.

Existing techniques for combining inference with search include unit propagation [2], conflict learning [3], and resolution-based methods [4, 5]. Some ways to reduce the overhead of inference include: using data structures that make fast inference operations possible; allowing inference only during preprocessing; and heuristics for determining when inference is worthwhile. We are working on explaining why some inference techniques are more beneficial than others, and using that information to select other beneficial inference techniques.

References

1. S.A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third ACM Symposium on Theory of Computing*, pages 151–158, 1971.
2. Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5:394–397, 1962.
3. João P. Marques Silva and Karem A. Sakallah. Conflict analysis in search algorithms for satisfiability. In *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, Nov 1996.
4. 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.
5. Allen van Gelder. Satisfiability testing with more reasoning and less guessing. In D. S. Johnson and M. Trick, editors, *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, 1995.