Automatic Generation of Implied Constraints: Project Description

Alan M. Frisch* and Toby Walsh†
Artificial Intelligence Group
Dept. of Computer Science, Univ. of York
York, YO10 5DD, United Kingdom

Constraint satisfaction is a highly successful technology for tackling a wide variety of search problems including resource allocation, transportation and scheduling. Several recent studies show that implied constraints added by hand to a problem representation can lead to significant reductions in search (e.g. [8]). The aim of this project is to develop, analyse and evaluate methods for generating implied constraints automatically. One of the key ideas will be to combine theorem proving techniques (like ordered resolution) with constraint satisfaction algorithms.

As an example of the value of implied constraints, consider colouring the nodes in a graph so that adjacent nodes have different colours. This constraint satisfaction problem models a variety of assignment problems like exam and classroom timetabling. If we have a near k-clique in which all but one pair of k nodes are connected and $k-1$ colours available, then we can infer an additional constraint that the two unconnected nodes must take the same colour. Adding this implied constraint explicitly to the problem representation could prevent a backtracking algorithm like forward checking from exploring an exponential number of partial colourings for the k nodes in the near k-clique.

Though those who formulate problems for constraint solvers appreciate the importance of adding implied constraints, there has been little research on how to generate such implied constraints automatically outside of highly focused domains like planning (see, for example, [4]). One exception is [5], which generalises resolution to multi-valued clauses (in which variables can take more than just the two values *True* and *False*), and proves that implied constraints generated by the closure of this operation will eliminate search. In practice, we cannot expect to eliminate search completely as the closure can be exponentially large to compute. One of the objectives of this research is to identify how much of the closure to generate to reduce the total time needed to solve a problem.

Many inference techniques can be viewed as methods for generating restricted classes of implied constraints. For example, consistency techniques like arc-consistency generate implied unary constraints. As a second example, nogood recording techniques [2] generate implied constraints from dead-ends in search. In the area of operations research cutting planes are highly effective at strengthening linear relaxations and at pruning search [7]. Cutting planes are linear inequalities implied by the original set of inequalities. Hooker has shown [7] that resolution (and its generalisation to multi-valued clauses) is a general method for generating cutting planes analogous to more traditional techniques like Chvátal's method. As a final example, in propositional satisfiability inference techniques such as directional resolution [3] can be highly competitive with more traditional branching techniques like the Davis-Putnam procedure [1]. Such methods generate resolvents which can again be viewed as implied constraints.

The aims of this project are to be achieved through four major objectives. Though each primarily focuses on the use of implied constraints for systematic search, we will also consider their use in local search.

The first objective is to develop theorem proving techniques for generating implied constraints automatically. We will encode the initial problem representation into a propositional

*Email: frisch@cs.york.ac.uk, tel: +44 1904 432745.
†Email: tw@cs.york.ac.uk, tel: +44 1904 432793.

theory and attempt to determine the extent to which standard theorem proving technologies, such as resolution, can generate constraints that have previously proven useful when generated by hand.

As theorem proving techniques are likely to generate both useful and unuseful constraints, the second objective is to develop heuristics for identifying which of the generated constraints to retain and which to discard. We will attempt to determine whether measures such as its "constrainedness" [6], size or tightness can guide this decision. It is known that an ordered form of resolution [5] can be used to generate constraints sufficient to give backtrack free search for a fixed variable ordering. But which of these implied constraints is necessary? Can we give similar results for dynamic variable orderings?

The first two objectives focus on the generation of constraints prior to the start of search. The third objective considers algorithms that interleave the generation of implied constraints with the search process itself. Consistency techniques, such as arc-consistency, can be viewed as generating unary constraints during search. The great success of these techniques in practice motivates us to consider whether other methods of deriving implied constraints during search can also be effective. A critical question to be addressed is how to partition effort between inference (generation of implied constraints) and search.

The fourth and final objective is to apply these results to a closely related domain, propositional satisfiability (SAT) and some extensions of SAT. In recent years, there has been considerable interest in encoding constraint satisfaction and other problems like planning into SAT, and then using either an efficient complete procedure like Davis-Putnam or a fast semi-decision procedures like GSAT or WalkSAT. The success of such an approach often depends critically on the implied constraints included in the encoding [4]. Are the techniques developed for generating implied constraints for constraint satisfaction problems useful for generating implied constraints when encoding problems into SAT?

The research methodology followed in this project is that of theoretical analysis backed up by large scale empirical tests. We will therefore prototype the various technologies being explored, and test these implementations on the wide variety of problems found in CSPLib.

We expect automated methods for generating implied constraints to become a vital component of the next generation of constraint satisfaction toolkits, and we hope to contribute to this development. Our results will be available at http://www.cs.york.ac.uk/aig/projects/implied.

References

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

[2] R. Dechter. Enhancement schemes for constraint processing: Backjumping, learning and cutset decomposition. *Artificial Intelligence*, 41(3):273–312, 1990.

[3] R. Dechter and I. Rish. Directional resolution: The Davis-Putnam procedure, revisited. In *Proceedings of KR-94*, pages 134–145, 1994. A longer version is available from `http://www.ics.uci.edu/~irinar`.

[4] M. Ernst, T.D. Millstein and D.S. Weld. Automatic SAT-Compilation of Planning Problems. In *Proceedings of the 14th IJCAI*, pages 1169-1177. International Joint Conference on Artificial Intelligence, 1997.

[5] A.M. Frisch. Solving Constraint Satisfaction Problems with NB-Resolution. In S. Muggleton, D. Michie and Luc De Raedt, editors, *Machine Intelligence 16*, 2000. Electronic Transactions in Artificial Intelligence. Available from `http://www.cs.york.ac.uk/~frisch/papers/mi16.ps`.

[6] I.P. Gent, E. MacIntyre, P. Prosser, and T. Walsh. The constrainedness of search. In *Proceedings of the 13th National Conference on AI*, pages 246–252. American Association for Artificial Intelligence, 1996.

[7] J.N. Hooker. Constraint satisfaction methods for generating valid cuts. In D. L. Woodruf, editor, *Advances in Computational and Stochastic Optimization, Logic Programming, and Heuristic Search*, pages 1–30. Kluwer, 1997.

[8] B.M. Smith, K. Stergiou, and T. Walsh. Modelling the Golomb ruler problem. In *Proceedings of the IJCAI-99 Workshop on Non-Binary Constraints*. International Joint Conference on Artificial Intelligence, 1999. Also available as APES report, APES-11-1999 from http://apes.cs.strath.ac.uk/reports/apes-11-1999.ps.gz.