Jefferson, Christopher; **Moore, Neil C. A.**; **Nightingale, Peter**; **Petrie, Karen E.**
Implementing logical connectives in constraint programming. (English) Zbl 1210.68103
Artif. Intell. 174, No. 16-17, 1407-1429 (2010).

Summary: Combining constraints using logical connectives such as disjunction is ubiquitous in constraint programming, because it adds considerable expressive power to a constraint language. We explore the solver architecture needed to propagate such combinations of constraints efficiently. In particular we describe two new features named satisfying sets and constraint trees. We also make use of movable triggers [*I. P. Gent*, *C. Jefferson* and *I. Miguel*, "Watched literals for constraint propagation in minion", Lect. Notes Comput. Sci. 4204, 182–197 (2006)], and with these three complementary features we are able to make considerable efficiency gains.

A key reason for the success of Boolean Satisfiability (SAT) solvers is their ability to propagate Or constraints efficiently, making use of movable triggers. We successfully generalise this approach to an Or of an arbitrary set of constraints, maintaining the crucial property that at most two constraints are active at any time, and no computation at all is done on the others. We also give an And propagator within our framework, which may be embedded within the Or. Using this approach, we demonstrate speedups of over 10,000 times in some cases, compared to traditional constraint programming approaches. We also prove that the Or algorithm enforces generalised arc consistency (GAC) when all its child constraints have a GAC propagator, and no variables are shared between children. By extending the Or propagator, we present a propagator for AtLeastK, which expresses that at least k of its child constraints are satisfied in any solution.

Some logical expressions (e.g. exclusive-or) cannot be compactly expressed using And, Or and AtLeastK. Therefore we investigate reification of constraints. We present a fast generic algorithm for reification using satisfying sets and movable triggers.

MSC:

68T20 Problem solving in the context of artificial intelligence (heuristics, search strategies, etc.)

Keywords:

constraint programming; constraint satisfaction problems; propagation algorithms; logical connectives

Software:

Chaff; MINION

Full Text: DOI

References:

[1] Gent, I.P.; Jefferson, C.; Miguel, I., Watched literals for constraint propagation in minion, (), 182-197
[2] Apt, K.R., Principles of constraint programming, (2003), Cambridge University Press · Zbl 1187.68132
[3] ()
[4] Gent, I.P.; Jefferson, C.; Miguel, I., Minion: A fast scalable constraint solver, (), 98-102
[5] Brand, S.; Yap, R.H.C., Towards "propagation = logic + control", (), 102-116 · Zbl 1131.68368
[6] Müller, T.; Würtz, J., Constructive disjunction in oz, (), 113-122
[7] Würtz, J.; Müller, T., Constructive disjunction revisited, (), 377-386
[8] Lagerkvist, M.Z.; Schulte, C., Propagator groups, (), 524-538
[9] Bacchus, F.; Walsh, T., Propagating logical combinations of constraints, (), 35-40
[10] Lhomme, O., An efficient filtering algorithm for disjunction of constraints, (), 904-908
[11] Lhomme, O., Arc-consistency filtering algorithms for logical combinations of constraints, (), 209-224 · Zbl 1094.68648

[12] Moskewicz, M.W.; Madigan, C.F.; Zhao, Y.; Zhang, L.; Malik, S., Chaff: engineering an efficient sat solver, (), 530-535

[13] P.V. Hentenryck, V. Saraswat, Y. Deville, Constraint processing in cc(fd), Technical report, Brown University, 1991. · Zbl 0920.68026

[14] J.C. Régin, A filtering algorithm for constraints of difference in CSPs, in: Proceedings 12th National Conference on Artificial Intelligence (AAAI 94), 1994, pp. 362-367.

[15] Aggoun, A.; Chan, D.; Dufresne, P.; Falvey, E.; Grant, H.; Harvey, W.; Herold, A.; Macartney, G.; Meier, M.; Miller, D.; Mudambi, S.; Novello, S.; Perez, B.; van Rossum, E.; Schimpf, J.; Shen, K.; Tsahageas, P.A.; de Villeneuve, D.H., Eclipse user manual release 5.10, (2006)

[16] Schulte, C., Programming deep concurrent constraint combinators, (), 215-229

[17] Bessiere, C.; Hebrard, E.; Hnich, B.; Walsh, T., The complexity of reasoning with global constraints, Constraints, 12, 2, 239-259, (2007) · Zbl 1124.68103

[18] C. Jefferson, Representations in constraint programming, PhD thesis, University of York, 2007.

[19] Colbourn, C.J.; Dinitz, J.H., Handbook of combinatorial designs, Discrete mathematics and its applications, (2006), Chapman \& Hall/CRC

[20] Daniel, P.; Semple, C., Supertree algorithms for nested taxa, (), 151-171

[21] Gent, I.P.; Prosser, P.; Smith, B.M.; Wei, W., Supertree construction with constraint programming, (), 837-841

[22] Moore, N.C.; Prosser, P., The ultrametric constraint and its application to phylogenetics, Journal of artificial intelligence research, 32, 901-938, (2008) · Zbl 1166.92033

[23] A.M. Frisch, I. Miguel, T. Walsh, Modelling a steel mill slab design problem, in: IJCAI-01 Workshop on Modelling and Solving Problems with Constraints, 2001, pp. 39-45.

[24] Gent, I.P.; Jefferson, C.; Kelsey, T.; Lynce, I.; Miguel, I.; Nightingale, P.; Smith, B.M., Search in the patience game 'black hole', AI communications, 20, 3, 211-226, (2007) · Zbl 1151.90525

[25] Gent, I.P.; Miguel, I.; Nightingale, P., Generalised arc consistency for the alldifferent constraint: an empirical survey, Artificial intelligence, 172, 18, 1973-2000, (2008) · Zbl 1184.68472

[26] Jefferson, C.; Miguel, A.; Miguel, I.; Tarim, A., Modelling and solving English peg solitaire, Computers and operations research, 33, 10, 2935-2959, (2006) · Zbl 1086.90070