Ben-Jose SAT solving software library.

Jose Luis Quiroga. 2014.

https://github.com/joseluisquiroga/ben-jose

JoseLuisQuirogaBeltran@gmail.com

Abstract. The software library Ben-Jose for solving instances of the satisfiability problem in CNF form is presented. Ben-Jose implements a trainable strategy that extends the traditional DPLL+BCP+CDCL resolution based approach, first introduced by Joao Marquez da Silva [01] and latter refined by others [14][15], with an original technique to check if the structure of a sub-formula of the solving SAT instance has previously been found to be unsatisfiable, in order to skip the search on it whenever found again. The technique introduces an original stabilization procedure (as in [16]) for the structure of the sub-formulas that is tightly coupled with the work done by BPC (each BCP step groups some variables) and has linear complexity.

Introduction.

The SAT problem is the canonical decision problem by excellence [02][03][04]. It lays at the heart of the P vs NP question and its importance cannot be overstated [05]. People have proved both that “NP != P” and that “NP==P” [06] [07], and filed patent applications for optimal SAT solvers based on resolution [08].

Exponential lower bounds of resolution (RES) proof size for the pigeon hole principle (PHP) have been proven [09] [10], and polynomial size proofs for extended resolution (ER) have also been proven for the PHP [11] [12] [13] instances of the SAT problem. That work suggests that solvers based on RES [01] [14] [15] need “something else” in order to be “faster” [17][18].

Based on the notion that theorems are proved with lemmas and the observation that the structure of PHP(n+1, n) can be matched with several substructures of PHP(n+2, n+1), the software library presented in this work (Ben-Jose) extends RES by learning the structure of unsatisfiable sub-formulas (proved unsatisfiable during its execution) and matching them against future structures of sub-formulas found during its execution, in order to skip the search, and directly backtrack on them, when ever a match (subsumed isomorphism) is found.

This technique is here after called Backtrack Driven by Unsatisfiable Substructure Training (BDUST). BDUST introduces an original stabilization procedure (as in [16]) for the unsatisfiable sub-formulas found, uses the work done by BPC (each BCP step groups some variables), and has linear complexity with respect to the size of the sub-formulas. The structures learned with BDUST have the advantage that can be used also with different instances than the one they were found on. That is why BDUST is “training” and not “learning”.

This form of extending RES is not ER, as the system presented by Tseitin [19], because the subsumed isomorphism detection technique is not RES based, and the power and complexity of the resulting proof system has not been formally studied. The empirical results on the classical problems of PHP and isPrime (based on the Braun multiplier) are presented here.

References

[01] Joao Marques da Silva. Search Algorithms for Satisfiability problems in combinatorial switching circuits, 1995. A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Electrical Engineering). The University of Michigan.

[02] Armin Biere. Handbook of Satisfiability, 2009.

[03] Daniel Kroening, Ofer Strichman. Decision Procedures: An Algorithmic Point of View, 2008.

[04] Victor W. Marek. Introduction to Mathematics of Satisfiability, 2009.

[05] Stephen Cook. The importance of the P versus NP Question, 2003. JACM 50, 1 (50th Anniversary Issue), pp 27-29.

[06] GJ Woeginger. The P-versus-NP page, 2014. http://www.win.tue.nl/~gwoegi/P-versus-NP.htm. .

[07] Matthias Muller. Polynomial SAT-Solver, 2013. http://vixra.org/pdf/1212.0109v2.pdf. (this one completely fools me).

[08] J. L. Quiroga. Optimal circuit verification method, 2001. http://www.google.com/patents/US20040250223. (shame on him).

[09] A Haken. The intractability of resolution, 1985. Theoretical computer science, 39, pp. 297-308..

[10] Samuel R. Buss, Gyorgy Turan. Resolution Proofs of Generalized Pigeonhole Principles, 1988. .

[11] S. A. Cook. A short proof of the pigeon hole principle using extended resolution, 1976. SIGACT News 8(4):28-32..

[12] S. A. Cook, R. A. Reckhow. The relative efficiency of propositional proof systems, 1979. Journal of Symbolic logic. 44, pp. 25-38.

[13] Matti Jarvisalo. Impact of Restricted Branching on Clause Learning SAT solving, 2007. Research Reports 107. pp.28. Helsinki University of Technology Laboratory for Theoretical Computer Science.

[14] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, Sharad Malik. Chaff: Engineering an Efficient SAT Solver, 2001. DAC 01. Proceedings of the 38th annual Design Automation Conference.

[15] Niklas Een, Niklas Sorensson. An Extensible SAT-solver, 2004. Chalmers University of Technology, Sweden.

[16] Oliver Bastert. Stabilization Procedures and Applications, 2002. Zentrum Mathematik. Technische Universitat Muchen.

[17] Heidi E. Dixon, Matthew L. Ginsberg, David Hofer, Eugene M. Luks, Andrew J. Parkes. Generalizing Boolean Satisfiability I, II and III: Background and Survey of Existing Work, Theory and Implementation, 2004. Journal of Artificial Intelligence Research 21, 22 and 23.

[18] Gilles Audemard, George Katsirelos, Laurent Simon. A Restriction of Extended Resolution for Clause Learning SAT Solvers, 2010. www.aaai.org.

[19] G. Tseitin. On the complexity of proofs in propositional logics, 1983. Automation of Reasoning: Classical Papers in Computational Logic 1967-1970. volume 2. Springer- Verlag.