

Elimination Techniques in Modern Propositional Logic Reasoning (Abstract of Invited Talk)

Norbert Manthey

nmanthey@conp-solutions.com

The satisfiability testing (SAT) problem is one of the most relevant problems of computer science, as SAT is the representative problem for the complexity class \mathcal{NP} [3]. Due to the numerous improvements to SAT solvers, many industrial problems are successfully reduced to SAT [3]. The highly optimized and specialized SAT solvers make these improvements accessible, and with such systems solving problems via SAT became effective.

Many recent improvements in SAT solvers are related to data structures, search heuristics or problem simplifications. However, the major reasoning techniques in propositional logic is resolution on clauses, used in unit propagation, variable elimination as well as clause learning [6, 7, 13]. State-of-the-art SAT solvers primarily use this technique to guide their search [1].

Both from a reasoning strength, as well as from an empirical analysis point of view, these systems still benefit from further problem simplifications, specifically variable elimination, where elimination is not only performed on pure clauses, but also on XOR constraints as well as cardinality constraints [2, 4, 8, 14]. For the two more expressive constraint types, constraints can even be extracted from formulas in CNF.

The relations between formulas F before and after F' a simplification have been described in [11]. For applied SAT solving, not only performance matters, but also the ability of constructing models for the original formula F based on a model for F' . All above mentioned elimination techniques have this property.

Further simplification techniques rely on removing clauses, e.g. blocked clause elimination [9]. From a proof complexity point of view, the counter technique – adding blocked clauses [10] – can lead to a much more powerful reasoning than resolution, namely introducing fresh variables via extended resolution [5]. Attempts on introducing fresh variables automatically exist [12], but are currently more used during encoding a problem into CNF than as a reasoning technique during search. Again, a model for the original formula can always be constructed based on a model of the simplified formula.

References

1. *Proceedings of SAT Competition 2017: Solver and Benchmark Descriptions*, volume B-2017-1 of *Publication series B*. University of Helsinki, Helsinki, Finland, 2017.
2. A. Balint and N. Manthey. Boosting the performance of SLS and CDCL solvers by preprocessor tuning. In *POS-13*, volume 29 of *EPiC Series*, pages 1–14. EasyChair, 2014.

Copyright © 2017 by the paper's authors

In: P. Koopmann, S. Rudolph, R. Schmidt, C. Wernhard (eds.): *SOQE 2017 – Proceedings of the Workshop on Second-Order Quantifier Elimination and Related Topics, Dresden, Germany, December 6–8, 2017*, published at <http://ceur-ws.org>.

3. A. Biere, M. J. H. Heule, H. van Maaren, and T. Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.
4. A. Biere, D. Le Berre, E. Lonca, and N. Manthey. Detecting cardinality constraints in CNF. In *SAT 2014*, volume 8561 of *LNCS*, pages 285–301, 2014.
5. S. A. Cook. A short proof of the pigeon hole principle using extended resolution. *SIGACT News*, 8(4):28–32, Oct. 1976.
6. M. Davis, G. Logemann, and D. Loveland. A machine program for theorem-proving. *Communications of the ACM*, 5(7):394–397, 1962.
7. M. Davis and H. Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7(3):201–215, 1960.
8. N. Eén and A. Biere. Effective preprocessing in SAT through variable and clause elimination. In *SAT 2005*, volume 3569 of *LNCS*, pages 61–75, 2005.
9. M. Järvisalo, A. Biere, and M. J. Heule. Blocked clause elimination. In *TACAS*, volume 6015 of *LNCS*, pages 129–144, 2010.
10. O. Kullmann. On a generalization of extended resolution. *Discrete Applied Mathematics*, 96-97(1):149–176, 1999.
11. N. Manthey. *Towards Next Generation Sequential and Parallel SAT Solvers*. PhD thesis, TU Dresden, 2014.
12. N. Manthey, M. J. Heule, and A. Biere. Automated reencoding of Boolean formulas. In *Hardware and Software: Verification and Testing*, volume 7857 of *LNCS*, pages 102–117, 2013.
13. J. P. Marques-Silva and K. A. Sakallah. GRASP – a new search algorithm for satisfiability. ICCAD '96, pages 220–227. IEEE Computer Society, 1996.
14. M. Soos, K. Nohl, and C. Castelluccia. Extending SAT solvers to cryptographic problems. In *SAT 2009*, volume 5584 of *LNCS*, pages 244–257, 2009.