# Blocked Clause Elimination and its Extensions

Marijn J. H. Heule*

joint work with Matti Järvisalo and Armin Biere

## Abstract

Boolean satisfiability (SAT) has become a core technology in many application domains, such as planning and formal verification, and continues to find various new application domains today. The SAT-based approach divides into three steps: encoding, preprocessing, and search. It is often argued that by encoding arbitrary Boolean formulas in conjunctive normal form (CNF), structural properties of the original problem are not reflected in the CNF. This should result in the fact that CNF-level preprocessing and SAT solver techniques have an inherent disadvantage compared to related techniques applicable on the level of more structural SAT instance representations such as Boolean circuits. Motivated by this, various simplification techniques and intricate CNF encodings for circuit-level SAT instance descriptions have been proposed. On the other hand, based on the highly efficient CNF-level clause learning SAT solvers, there is also strong support for the claim that CNF is sufficient as an input format for SAT solvers.

In this work we study the effect of a CNF-level simplification technique called blocked clause elimination (BCE) [4].We show that BCE is surprisingly effective both in theory and in practice on CNF formulas resulting from a standard CNF encoding for circuits [3]: without explicit knowledge of the underlying circuit structure, it achieves the same level of simplification as a combination of circuit-level simplifications and previously suggested polarity-based CNF encodings.

Encouraged by the results of BCE, we developed five extensions of this clause elimination technique. This work is motivated on one hand by the possibilities of lifting SAT solving efficiency further by integrating additional simplification techniques to the solving process before and/or during search, and on the other by understanding the relationships between these different simplification techniques. As extensions of BCE we introduce novel elimination procedures based on adding *hidden* [1] and *asymmetric* and *covered* [2] literals to clauses and then check whether these extended clauses are blocked.

*Supported by the Dutch Organization for Scientific Research (NWO) under grant 617.023.611

We analyze the resulting six clause elimination procedures from various perspectives. One property is *effectiveness* (or *size reduction*), i.e., the ability to remove clauses and thus reduce the size of the CNF formula. Notice that reducing the size of a CNF by preprocessing does not necessarily lead to faster running times. The second property, *confluence*, implies that a procedure has a unique fixpoint. Both explicit BCE and the variant which adds covered literals have this property. The third is *logical equivalence* w.r.t. the original CNF, i.e. preserving the set of satisfying assignments.

We also address the problem of reconstructing original solutions to CNF formulas after applying a preprocessing. For many real application scenarios of SAT it is important to be able to extract a full satisfying assignment for original SAT instances from a satisfying assignment for the instances after preprocessing. For instance, when applying BCE, a solution to the original CNF is not directly available in general. We show how such full solutions can be efficiently reconstructed from solutions to the conjunctive normal form (CNF) formulas resulting from applying a blocked clause elimination and its hidden and asymmetric variants.

Finally, we describe how the explicit clause elimination procedures can be implemented efficiently. Also, we evaluate the practical effectiveness of selected procedures, investigating both the CNF size reduction and resulting solving times.

## References


[1] Marijn Heule, Matti Järvisalo, and Armin Biere. Clause elimination procedures for CNF formulas. In Christian Fermüller and Andrei Voronkov, editors, *Proceedings of the 17th International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR-17)*, volume 6397 of *Lecture Notes in Computer Science*, pages 357–371. Springer, 2010.

[2] Marijn Heule, Matti Järvisalo, and Armin Biere. Covered clause elimination. In *17th International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR-17)*, 2010. Accepted as short paper.

[3] Matti Järvisalo, Armin Biere, and Marijn Heule. Blocked clause elimination. In Javier Esparza and Rupak Majumdar, editors, *Proceedings of the 16th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2010)*, volume 6015 of *Lecture Notes in Computer Science*, pages 129–144. Springer, 2010.

[4] Oliver Kullmann. On a generalization of extended resolution. *Discrete Applied Mathematics*, 96–97:149–176, 1999.