

# CAQE and QuAbS at QBFEval 2020

Leander Tentrup, Saarland University

## 1 Introduction

This paper gives an overview of the submissions to QBFEval 2019. The prenex CNF solver CAQE [7] and the prenex non-CNF solver QuAbS [4]. We refer to [12] for a detailed system description.

### 1.1 CAQE

The solver CAQE (version 4.0) is based on the *clausal abstraction* algorithm [5, 7]<sup>1</sup> for solving QBF. The underlying idea of clausal abstraction is to assign variables, where the assignment order is determined by the quantifier prefix until either all clauses are satisfied or there is a set of clauses that cannot be satisfied at the same time. The effect of assignments, i.e., whether they satisfy a clause, is abstracted into one bit of information per clause, and this information is communicated through the quantifier prefix. The fundamental data structure of the algorithm is an abstraction, a propositional formula for each maximal block of quantifiers, that, given the valuation of outer variables, generates candidate assignments for the variables bound at this quantifier block. In case this candidate is refuted by inner quantifiers, the returned counterexample is excluded in the abstraction. Thus, the clausal abstraction algorithm uses ideas of search-based solving [3], and counterexample guided abstraction refinement (CEGAR) algorithms [2]. A proof-theoretic analysis of the clausal abstraction approach [11] has shown that the

refutation proofs correspond to the (level-ordered)  $Q$ -resolution calculus [6].

The implementation is written in Rust and uses CryptoMiniSat [9] in version 5.0.1 as the underlying SAT solver. The source code of CAQE is available at <https://github.com/ltentrup/caqe>. We submitted three configurations of CAQE, one using only HQSPre [13] as preprocessor, one configuration using both, Bloqqer [1] and HQSPre, and one configuration that produces QDIMACS outputs using the partial assignments from Bloqqer [8].

### 1.2 QuAbS

QuAbS is based on the extension of clausal abstraction to formulas in negation normal form [4, 10]. The implementation is written in C++ and uses CryptoMiniSat [9] in version 5.5 as the underlying SAT solver. The source code of QuAbS is available at <https://github.com/ltentrup/quabs>. In addition to QuAbS, we submitted a configuration of CAQE that transforms the circuit representation and its negation to CNF.

## 2 Acknowledgments

This work was partially supported by the German Research Foundation (DFG) as part of the Collaborative Research Center “Foundations of Perspicuous Software Systems” (TRR 248, 389792660) and by the European Research Council (ERC) Grant OSARES (No. 683300).

---

<sup>1</sup>called *clause selection* in [5]

## References

- [1] Armin Biere, Florian Lonsing, and Martina Seidl. Blocked clause elimination for QBF. In *Proceedings of CADE-23*, volume 6803 of *LNCS*, pages 101–115. Springer, 2011.
- [2] Edmund M. Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veith. Counterexample-guided abstraction refinement. In *Proceedings of CAV*, volume 1855 of *LNCS*, pages 154–169. Springer, 2000.
- [3] Enrico Giunchiglia, Paolo Marin, and Massimo Narizzano. Reasoning with quantified boolean formulas. In *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*, pages 761–780. IOS Press, 2009.
- [4] Jesko Heicking-Harbusch and Leander Tentrup. Solving QBF by abstraction. In *Proceedings of GandALF*, volume 277 of *EPTCS*, pages 88–102, 2018.
- [5] Mikolás Janota and João Marques-Silva. Solving QBF by clause selection. In *Proceedings of IJ-CAI*, pages 325–331. AAAI Press, 2015.
- [6] Hans Kleine Büning, Marek Karpinski, and Andreas Flögel. Resolution for quantified boolean formulas. *Inf. Comput.*, 117(1):12–18, 1995.
- [7] Markus N. Rabe and Leander Tentrup. CAQE: A certifying QBF solver. In *Proceedings of FMCAD*, pages 136–143. IEEE, 2015.
- [8] Martina Seidl and Robert Könighofer. Partial witnesses from preprocessed quantified boolean formulas. In *Proceedings of DATE*, pages 1–6. European Design and Automation Association, 2014.
- [9] Mate Soos, Karsten Nohl, and Claude Castelluccia. Extending SAT solvers to cryptographic problems. In *Proceedings of SAT*, volume 5584 of *LNCS*, pages 244–257. Springer, 2009.
- [10] Leander Tentrup. Non-prenex QBF solving using abstraction. In *Proceedings of SAT*, volume 9710 of *LNCS*, pages 393–401. Springer, 2016.
- [11] Leander Tentrup. On expansion and resolution in CEGAR based QBF solving. In *Proceedings of CAV*, volume 10427 of *LNCS*, pages 475–494. Springer, 2017.
- [12] Leander Tentrup. CAQE and quabs: Abstraction based QBF solvers. *J. Satisf. Boolean Model. Comput.*, 11(1):155–210, 2019.
- [13] Ralf Wimmer, Sven Reimer, Paolo Marin, and Bernd Becker. Hqspre - an effective preprocessor for QBF and DQBF. In *Proceedings of TACAS*, volume 10205 of *LNCS*, pages 373–390, 2017.