

# Ultimate Eliminator: a Quantifier Upgrade for SMT Solvers at SMT-COMP 2020

Max Barth, Daniel Dietsch, Leonard Fichtner, Matthias Heizmann, and  
Andreas Podelski

University of Freiburg, Germany

## Abstract

Ultimate Eliminator is a tool for eliminating quantifiers in SMT formulas. At the SMT-COMP 2020 we participate with a user interface that takes SMT-LIB 2.6 compatible input, tries to eliminate quantifiers in this input and passes the modified input to the MathSAT SMT solver.

## 1 Overview

Quantified formulas are notoriously difficult to solve and several state-of-the-art SMT solvers support only quantifier-free formulas. We found out that, perhaps surprisingly, the quantifier elimination algorithms that are implemented in the Ultimate software verification framework can often find quantifier-free formulas that are logically equivalent to formulas in SMT-LIB benchmarks. Hence, these quantifier elimination algorithms empower existing SMT solvers without support for quantifiers to solve quantified formulas. E.g., a formula of the form  $\exists x. \varphi(x) \wedge x = t$  is transformed to  $\varphi(t)$  and the formula  $\forall a. \text{select}(a, k) = \text{select}(a, i)$  is transformed to  $k = i$ .

## 2 Quantifier Elimination in Ultimate

A key algorithm [2] of several software verifiers [3] in the Ultimate framework<sup>1</sup> does an iterative application of the strongest post predicate transformer to a sequence of statements resp. an iterative application of the weakest precondition predicate transformer. Both variants of the algorithm produce quantified formulas and the handling of these quantified formulas often was a bottleneck for the overall software verification approach. The performance of the tool improved significantly when the developers started to apply quantifier elimination techniques to every intermediate result. As a consequence, in the last years an increasing number of quantifier elimination techniques was implemented into the Ultimate framework.

## 3 Ultimate Eliminator

Ultimate Eliminator<sup>2</sup> is a user interface that takes SMT-LIB 2.6 compatible input tries to eliminate quantifiers in this input and passes the modified input to a user defined SMT-LIB 2.6 compatible solver. Ultimate Eliminator is implemented in Java as a plug-in of the Ultimate framework. The source code is available in a public repository<sup>3</sup>.

---

<sup>1</sup><https://ultimate.informatik.uni-freiburg.de/>

<sup>2</sup><https://ultimate.informatik.uni-freiburg.de/eliminator/>

<sup>3</sup><https://github.com/ultimate-pa/ultimate/>

## 4 SV-COMP 2020 Submissions

We submitted three SMT solvers to the competition. Each of them wraps an SMT solver that does not support quantifiers. For each logic that the wrapped solver (to the best of our knowledge) supports, we participated in the corresponding logic with quantifiers.

Our `ULTIMATEELIMINATOR+MATHSAT-5.6.3.S` submission is based on Ultimate 0.1.25-36245e4 and wraps the version 5.6.3 of the MathSAT SMT solver<sup>4</sup>[1]. We call MathSAT without any additional arguments.

Our wrapper tool participates in the Single Query Track, in the Incremental Track, and in the Unsat Core Track. We did not participate in other tracks because the other tracks use only quantifier-free benchmarks.

## References

- [1] Alessandro Cimatti, Alberto Griggio, Bastiaan Joost Schaafsma, and Roberto Sebastiani. The `mathsat5` SMT solver. In *TACAS*, volume 7795 of *Lecture Notes in Computer Science*, pages 93–107. Springer, 2013.
- [2] Daniel Dietsch, Matthias Heizmann, Betim Musa, Alexander Nutz, and Andreas Podelski. Craig vs. newton in software model checking. In *ESEC/SIGSOFT FSE*, pages 487–497. ACM, 2017.
- [3] Matthias Heizmann, Yu-Fang Chen, Daniel Dietsch, Marius Greitschus, Jochen Hoenicke, Yong Li, Alexander Nutz, Betim Musa, Christian Schilling, Tanja Schindler, and Andreas Podelski. Ultimate automizer and the search for perfect interpolants - (competition contribution). In *TACAS (2)*, volume 10806 of *Lecture Notes in Computer Science*, pages 447–451. Springer, 2018.

---

<sup>4</sup><http://mathsat.fbk.eu/>