Bitwuzla at the SMT-COMP 2023

Aina Niemetz
Stanford University

Mathias Preiner
Stanford University

***Abstract*—In this paper, we present our Satisfiability Modulo Theories (SMT) Bitwuzla as submitted to SMT-COMP 2023. Bitwuzla is a new solver for the theories of bit-vectors, floating-points, arrays and uninterpreted functions and their combinations (with and without quantifiers). Previous versions of Bitwuzla competing in SMT-COMP were an improved and extended version of a fork of Boolector. The new version of Bitwuzla entering SMT-COMP 2023 is a rewrite from scratch. A comprehensive system description of this new version of Bitwuzla will be presented at CAV 2023.**

I. Introduction

Bitwuzla is a Satisfiability Modulo Theories (SMT) solver for the (quantifier-free and quantified) theories of fixed-size bit-vectors, floating-point arithmetic, arrays and uninterpreted functions and their combinations. Previous versions of Bitwuzla as entered in SMT-COMP from 2020–2022 [5] were an improved and extended version of a fork of Boolector [8]. For this new version of Bitwuzla submitted to SMT-COMP 2023, we discarded the existing code base and rewrote Bitwuzla from scratch. It is written in C++ and inspired by techniques implemented in Boolector and previous versions of Bitwuzla. A comprehensive system description of Bitwuzla will be presented at CAV 2023 [7].

Bitwuzla implements the lazy, abstraction/refinement-based SMT paradigm lemmas on demand, using a bit-vector abstraction similar to [4, 9]. At its core, Bitwuzla maintains a bit-vector theory solver and a solver for each supported theory. Quantifier reasoning is handled by a dedicated quantifiers module, implemented as a theory solver.

The array theory solver implements and extends the array procedure from [4] with support for reasoning over (equalities of) nested arrays. The bit-vector theory solver implements two orthogonal approaches, classic bit-blasting and ternary propagation-based local search [6], and a sequential combination of both. For theory of floating-point arithmetic, Bitwuzla integrates SymFPU [3], a C++ library of bit-vector encodings of floating-point operations. For uninterpreted functions, Bitwuzla implements a lazy form of Ackermann's reduction. The quantifiers module implements model-based quantifier instantiation for all supported theories and their combinations. Bitwuzla further supports incremental solving (including incremental preprocessing), model generation, unsat core and unsat assumptions extraction.

This paper serves as system description for Bitwuzla as entered in the SMT competition 2023 [2]. Bitwuzla is licensed under the MIT license. Source code, releases and more information is available on the Bitwuzla website [1].

II. Configurations

Bitwuzla participates in the single query, incremental, unsat core, and model validation tracks in the logics matching the following regular expression:

```
^(QF_)?(A)?(UF)?(BV|FP|FPLRA)+$
```

III. License

Bitwuzla is licensed under the MIT license. For more details, refer to the actual license text, which is distributed with the source code.

References

[1] Bitwuzla website. https://bitwuzla.github.io, 2021.

[2] SMT-COMP 2023 website. https://www.smt-comp.org/2023, 2023.

[3] M. Brain, F. Schanda, and Y. Sun. Building better bit-blasting for floating-point problems. In *TACAS 2019, Prague, Czech Republic, April 6-11, 2019, Proceedings, Part I*, volume 11427 of *LNCS*, pages 79–98. Springer, 2019.

[4] R. Brummayer and A. Biere. Lemmas on demand for the extensional theory of arrays. *J. Satisf. Boolean Model. Comput.*, 6(1-3):165–201, 2009.

[5] A. Niemetz and M. Preiner. Bitwuzla at the SMT-COMP 2020. *CoRR*, abs/2006.01621, 2020.

[6] A. Niemetz and M. Preiner. Ternary propagation-based local search for more bit-precise reasoning. In *2020 Formal Methods in Computer Aided Design, FMCAD 2020, Haifa, Israel, September 21-24, 2020*, pages 214–224. IEEE, 2020.

[7] A. Niemetz and M. Preiner. Bitwuzla. In *CAV 2023, Paris, France, July 17-21, 2023, Proceedings*, Lecture Notes in Computer Science. Springer, 2023 (to appear).

[8] A. Niemetz, M. Preiner, C. Wolf, and A. Biere. BTOR2 , BtorMC and Boolector 3.0. In *CAV 2018, Oxford, UK, July 14-17, 2018, Proceedings, Part I*, volume 10981 of *Lecture Notes in Computer Science*, pages 587–595. Springer, 2018.

[9] M. Preiner, A. Niemetz, and A. Biere. Lemmas on demand for lambdas. In M. K. Ganai and A. Sen, editors, *Proceedings of the Second International Workshop on Design and Implementation of Formal Tools and Systems, Portland, OR, USA, October 19, 2013*, volume 1130 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013.