Bitwuzla at the SMT-COMP 2025

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 2025. Bitwuzla is a solver for the theories of bit-vectors, floating-points, arrays and uninterpreted functions and their combinations (with and without quantifiers).**

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. A comprehensive system description of Bitwuzla can be found in [10].

Bitwuzla implements the lazy, abstraction/refinement-based SMT paradigm *lemmas on demand*, using a bit-vector abstraction similar to [5, 12].

Bit-Vectors. The bit-vector theory solver implements two orthogonal approaches, classic *bit-blasting* and *ternary propagation-based local search* [9], and a sequential combination of both. Additionally, Bitwuzla's bit-blasting solver integrates an abstraction-refinement approach for bit-vector arithmetic as described in [11].

Arrays. The array theory solver implements and extends the array procedure from [5] with support for reasoning over (equalities of) nested arrays.

Floating-Point Arithmetic. For the theory of floating-point arithmetic, Bitwuzla implements a technique called *word-blasting* and integrates SymFPU [4], a C++ library of bit-vector encodings of floating-point operations.

Uninterpreted Functions. For uninterpreted functions, Bitwuzla implements a lazy form of Ackermann's reduction called *dynamic Ackermannization* [6].

Quanifiers. Quantifier reasoning is handled by a dedicated quantifiers module, which implements model-based quantifier instantiation (MBQI) [8].

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 2025 [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, model validation, and parallel tracks in the logics matching the following regular expression:

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

Bitwuzla uses CaDiCaL version 2.1.2 [3] as default SAT back end. Our abstraction-refinement approach [11] is enabled by default, with the minimum bit-vector size of arithmetic terms to abstract set to 33. For the parallel track, Bitwuzla uses Gimsatul version 1.1.2 [7] as the SAT back end.

A. Improvements

- Improved preprocessing pipeline (optimized for incremental)
- Improved nested arrays support
- Gimsatul SAT backend for parallel workloads
- Custom memory allocator (mimalloc)

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 2024 website. https://www.smt-comp.org/2024, 2024.
[3] A. Biere, T. Faller, K. Fazekas, M. Fleury, N. Froleyks, and F. Pollitt. Cadical 2.0. In A. Gurfinkel and V. Ganesh, editors, *Computer Aided Verification - 36th International Conference, CAV 2024, Montreal, QC, Canada, July 24-27, 2024, Proceedings, Part I*, volume 14681 of *Lecture Notes in Computer Science*, pages 133–152. Springer, 2024.
[4] 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.
[5] 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.
[6] B. Dutertre and L. de Moura. The Yices SMT Solver. 2006.
[7] M. Fleury and A. Biere. Scalable proof producing multi-threaded SAT solving with gimsatul through sharing instead of copying clauses. *CoRR*, abs/2207.13577, 2022.
[8] Y. Ge and L. M. de Moura. Complete instantiation for quantified formulas in satisfiabiliby modulo theories. In A. Bouajjani and O. Maler, editors, *Computer Aided Verification, 21st International Conference, CAV 2009, Grenoble, France, June 26 - July 2, 2009. Proceedings*, volume 5643 of *Lecture Notes in Computer Science*, pages 306–320. Springer, 2009.
[9] 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.
[10] A. Niemetz and M. Preiner. Bitwuzla. In C. Enea and A. Lal, editors, *Computer Aided Verification - 35th International Conference, CAV 2023, Paris, France, July 17-22, 2023, Proceedings, Part II*, volume 13965 of *Lecture Notes in Computer Science*, pages 3–17. Springer, 2023.
[11] A. Niemetz, M. Preiner, and Y. Zohar. Scalable bit-blasting with abstractions. In *CAV 2024, Montreal, Canada, July 22-27, 2024, Proceedings*, Lecture Notes in Computer Science. Springer, 2024 (to appear).

[12] 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.