Bitwuzla: A New SMT Solver For Bit-Precise Reasoning

Aina Niemetz and **Mathias Preiner**

CENTAUR Meeting, August 17, 2023

Stanford University

CENTAUR
Stanford | Center for Automated Reasoning

Bitwuzla

A New SMT Solver

. . . for **quantified** and **quantifier-free** theories of

- fixed-size bit-vectors
- floating-point arithmetic
- arrays
- uninterpreted functions

and their combinations.

- **Pronounced as** “**bitvootslah**”
- Derived from an Austrian dialect expression for **someone who tinkers with bits**.

Successor of Boolector

Boolector

- ► An award-winning SMT solver, but ...
  - ○ Specialized, tight integration of **bit-vectors with arrays**
  - ○ **Monolithic** C code base, **rigid** architecture

- ► Cumbersome to maintain, adding new features **difficult**

Bitwuzla

- Started as an **improved and extended** fork of Boolector in 2018
  - ► **No** official release, limitations of Boolector remained
- In 2022, code base discarded and **rewritten from scratch**
- Written in C++, **inspired** by techniques in Boolector

- ► Bitwuzla considered **superior successor** of Boolector

Selected Features

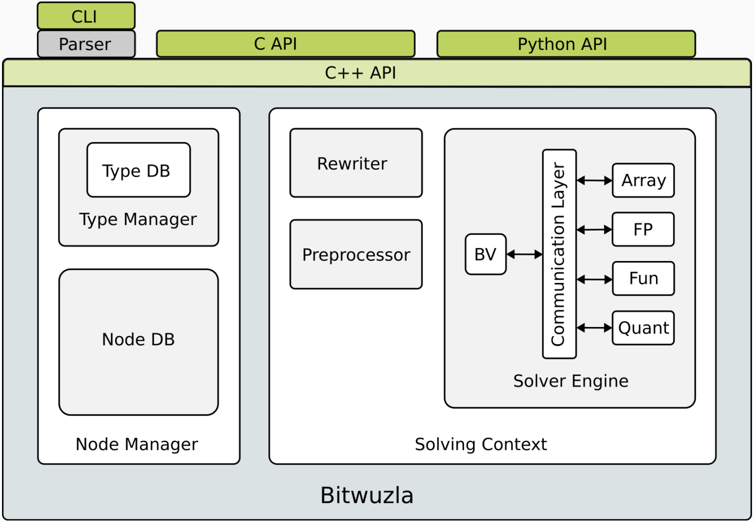
Theories

- **Focus**: Theories primarily used in **hardware verification**
- Arrays, bit-vectors, floating-point arith., uninterpreted functions
- Quantifiers in combination with **all** supported theories

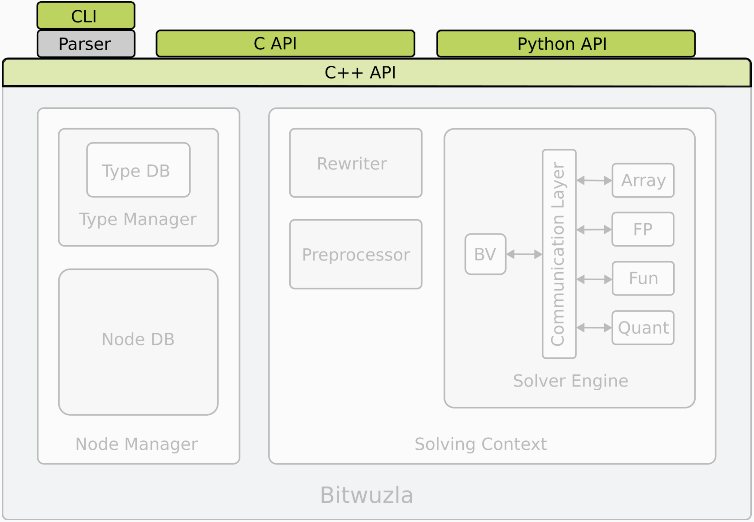
User-Facing

- **Full incremental** support
- Seamless interaction between **multiple solver instances**
- Models, unsat cores/assumptions
- Comprehensive and easy-to-use **APIs**
  - C++, C, Python, OCaml (WIP), Rust (planned)
- **Input Formats**: SMT-LIBv2, BTOR2, SMT-LIBv3 (planned)

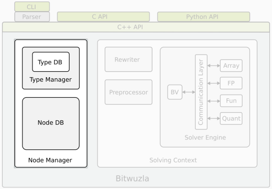
Architecture



Public Interface



Node Manager

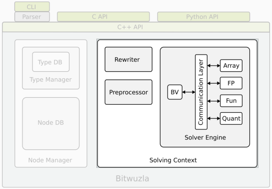


- Formulas and terms represented as reference-counted, immutable nodes in directed acyclic graph

Node Manager, Type Manager

- Used to manage and construct nodes/types
- Employs hash-consing to maximize sharing of subgraphs
- **Global** (thread-local) node and type storage
  - Allows sharing between arbitrarily many solving contexts

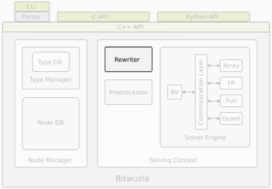
Solving Context



Solving Context

- Internal equivalent of **solver instance**
- Determines satisfiability of a set of asserted formulas
- **Fully configurable** via options
- **Incremental interface** for adding/removing assertions via push/pop
- Provides **models** for satisfiable queries
- Provides **unsat cores** for unsatisfiable queries
- Consists of **three main components**:
  Rewriter, **Preprocessor**, **Solver Engine**

Rewriter

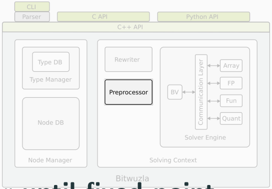


- Transforms terms via predefined set of rewrites rules into semantically equivalent normal forms

Rewriting

- **Local** (independent from current set of assertions)
- Implements more than 230 rules
- **Required** and **optional** rewrite rules grouped into levels 0–2
  - 0 . . . required rewrite rules (e.g.: $-x \rightsquigarrow \sim x + 1$)
  - 1 . . . immediate children only (e.g.: $x + 0 \rightsquigarrow \mathbf{x}$)
  - 2 . . . multiple levels of children (e.g.: $a - b \mathbf{=} \mathbf{c} \rightsquigarrow b + c \mathbf{=} \mathbf{a}$)
- Implemented as preprocessing pass

Preprocessor



Preprocessing

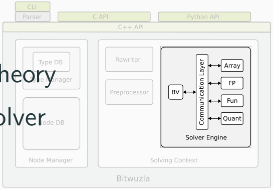
- Invoked prior to solving
- **Global** (based on current set of assertions)
- Applies preprocessing passes in predefined order **until fixed-point**
- Passes implement a set of satisfiability-preserving transformations
- **Fully incremental**
- **Speculative** preprocessing
- Most passes are optional

Passes

Rewriting, Boolean And Flattening, **Term Substitution**, Boolean Skeleton Preprocessing, Embedded Constraints Substitution, Bit-vector Extract Elimination, **Arithmetic Normalization**, Lambda Elimination, . . .

Solver Engine

Solver Engine



- Maintains a theory solver for each supported theory
- **Quantifiers module** implemented as theory solver
- **Distributes relevant terms** to theory solvers
- Processes lemmas generated by theory solvers
- Implements lazy SMT paradigm **lemmas on demand**
- **Bit-vector abstraction** of formula (instead of propositional)
    - Bit-vector solver at its core

Bit-Vector Abstraction

- BV solver reasons about **Boolean and bit-vector terms**
- Non-BV theory atoms abstracted as **Boolean constant**
- BV terms with non-BV operator abstracted as **bit-vector constant**

Theory Solvers

Bit-Vectors

- **Bit-blasting**: BV terms → AIG circuits (+rewriting [BB'06]) → CNF
- **Ternary propagation-based local search** [Niemetz'20]

Floating-Point Arithmetic

- **Word-blasting**: FP terms → BV terms (via SymFPU [BSS'19])

Arrays

- **Lemmas on Demand for Extensional Arrays** [BB'09]
- Supports extensional **nested arrays** and **constant arrays** (ext. WIP)

Uninterpreted Functions

- Dynamic Ackermannization [DdM'06]

Quantifiers

- Model-based Quantifier Instantiation (MBQI) [GdM'09]

Evaluation

Setup

- Comparison against
  - **Boolector**
  - **Z3** (SMT-COMP'22 version)
  - **cvc5** (SMT-COMP'22 version)
  - **SC22** (Bitwuzla SMT-COMP'22 version)
- SMT-LIB 2022 benchmarks
  - $146,235$ **non-incremental** benchmarks in 23 supported logics
  - $25,443$ **incremental** benchmarks in 15 supported logics
- Limits: 1200 seconds, 8GB memory

Evaluation (non-incremental)

Results

	Boolector	Z3	cvc5	SC22	Bitwuzla
Total (146,235)	64,106	141,778	142,995	143,617	**144,287**
Time (solved) [s]	417,643	1,212,584	1,000,466	563,832	580,435

- Solves **largest** number of benchmarks (**+670** compared to SC22)
- Solves **most** benchmarks in **13 out of 23** logics
- On 140, 438 **commonly** solved:
  - slightly faster than SC22 (203, 838*s* vs 208, 310*s*)
  - **2.85× faster** than cvc5 (586, 105*s*)
  - **5.1× faster** than Z3 (1, 049, 534*s*)

SMT-COMP'23 SQ Track: **17 out of 30** gold medals in 6 divisions

Evaluation (incremental)

Results

	Boolector	**Z3**	**cvc5**	**SC22**	**Bitwuzla**
Total (699,612)	60,113	657,512	673,642	685,006	**693,263**
Time (solved) [s]	102,812	3,359,645	1,516,672	157,083	172,534

- Solves **largest** number of queries (**+8257** compared to SC22)
- Solves **most** queries in **11 out of 15** logics

SMT-COMP'23 INC Track: **5 out of 6** gold medals in 6 divisions

Conclusion

CAV Artifact Evaluation ★ Available

CAV Artifact Evaluation ★ ★ ★ Reusable

Bitwuzla

▶ A new state-of-the-art SMT solver for all things bits (and more)

Open Source

- MIT license
- Source code: `https://github.com/bitwuzla/bitwuzla`

Website and Documentation: `https://bitwuzla.github.io`

Appendix: Non-Incremental Results

	Boolector	**Z3**	**cvc5**	**SC22**	**Bitwuzla**
ABV (169)	-	**89**	32	0	1
ABVFP (30)	-	**25**	19	0	16
ABVFPLRA (75)	-	**47**	36	0	31
AUFBV (1,522)	-	403	486	597	**983**
AUFBVFP (57)	-	7	21	24	**39**
BV (6,045)	5,659	5,593	**5,818**	5,624	5,705
BVFP (205)	-	176	171	148	**188**
BVFPLRA (209)	-	189	107	140	**199**
FP (2,669)	-	2,128	2,353	**2,513**	2,481
FPLRA (87)	-	72	51	55	**83**
QF_ABV (15,084)	15,041	14,900	14,923	**15,043**	15,041
QF_ABVFP (18,129)	-	18,017	18,113	**18,125**	**18,125**
QF_ABVFPLRA (74)	-	69	**74**	34	**74**
QF_AUFBV (67)	45	50	42	46	**55**
QF_AUFBVFP (1)	-	1	1	1	1
QF_BV (42,472)	41,958	40,876	41,574	42,039	**42,049**
QF_BVFP (17,244)	-	17,229	17,238	**17,242**	17,241
QF_FP (40,409)	-	40,303	40,357	**40,368**	40,358
QF_FPLRA (57)	-	41	48	**56**	**56**
QF_UFBV (1,434)	1,403	1,404	1,387	**1,413**	1,411
QF_UFFP (2)	-	2	2	2	2
UFBV (192)	-	**156**	141	146	147
UFBVFP (2)	-	1	1	1	1
Total (146,235)	64,106	141,778	142,995	143,617	**144,287**
Time (solved) [s]	417,643	1,212,584	1,000,466	563,832	580,435

Appendix: Incremental Results

Logic	**Boolector**	**Z3**	**cvc5**	**SC22**	**Bitwuzla**
ABVFPLRA (2,269)	-	2,220	818	55	**2,269**
BV (38,856)	-	**37,188**	36,169	35,567	35,246
BVFP (458)	-	**458**	**458**	274	**458**
BVFPLRA (5,597)	-	**5,507**	2,964	3,144	4,797
QF_ABV (3,411)	3,238	2,866	2,746	**3,242**	2,939
QF_ABVFP (550,088)	-	515,714	534,629	550,034	**550,041**
QF_ABVFPLRA (1,876)	-	48	**1,876**	**1,876**	**1,876**
QF_AUFBV (967)	23	860	320	23	**956**
QF_BV (53,684)	52,218	51,826	51,683	51,581	**52,305**
QF_BVFP (3,465)	-	3,403	3,437	**3,444**	3,438
QF_BVFPLRA (32,736)	-	31,287	32,681	**32,736**	**32,736**
QF_FP (663)	-	663	663	663	663
QF_FPLRA (48)	-	48	48	48	48
QF_UFBV (5,492)	4,634	5,422	5,148	2,317	**5,489**
QF_UFFP (2)	-	2	2	2	2
Total (699,612)	60,113	657,512	673,642	685,006	**693,263**
Time (solved) [s]	102,812	3,359,645	1,516,672	157,083	172,534

References

- R. Brummayer, A. Biere. *Lemmas on Demand for the Extensional Theory of Arrays.* JSAT, 2009.
- R. Brummayer, A. Biere. *Local Two-Level And-Inverter Graph Minimization without Blowup.* In Proc. of MEMICS'06, 2006.
- M. Brain, F. Schanda, Y. Sun. *Building Better Bit-Blasting for Floating-Point Problems.* In Proc. of TACAS'19, 2019.
- B. Dutertre, L. de Moura. *The Yices SMT Solver.*, 2006.
- Y. Ge, L. de Moura. *Complete Instantiation for Quantified Formulas in Satisfiabiliby Modulo Theories*, In Proc. of CAV'09, 2009.

References

A. Niemetz, M. Preiner. *Ternary Propagation-Based Local Search for more Bit-Precise Reasoning.* In Proc. of FMCAD'20, pages 214–224, IEEE, 2020.