

Understanding Bit-vector Arithmetic in Z3

1. Introduction

- Z3 — a satisfiability modulo theories (SMT) solver — helps developers by serving as a backend for formal verification tools.

variables $x, y \in \text{BV}[4]$; $x \& y = y$
BV formula
boolean formula
 $x + y < 11 \vee x \cdot y = 2$

$Z3 \rightarrow \begin{array}{l|l} x = 4 & x = 14 \\ y = 14 & y = 15 \end{array}$

1 1 1 0 0 1 1 0 1
925 -99 ACAC TCGG DDD

QF_BV Quantifier-free logic with only fixed-size bit-vector and boolean arithmetic.

2. Research Questions

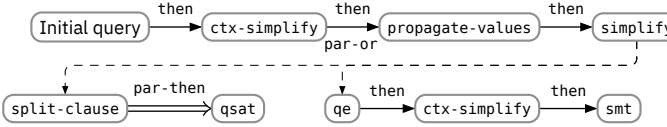
- What techniques can Z3 use to solve QF_BV problems?
 - Exploratory literature study
- Which techniques work well with which sets of problems?
 - Experimental evaluation
- How is Z3 able to use parallelization when solving problems?
 - Literature study + experiments

6. Conclusions

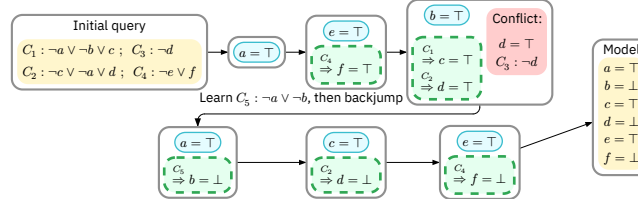
- Sls** is the best approach for most problems, especially easier ones.
- Polysat** and **int-blast** are significantly better on certain datasets.
- Cube-and-conquer** is useful between 8 and 32 CPU cores. 2 and 4 core setups work well only on harder problems.
- Portfolio solving** could be used to complement cube-and-conquer. **Polysat** and **sls** are good candidates for this.
- In the future, automated configuration can be explored further.

3. How Z3 Solves Bit-vector Problems

Strategy and Tactics



The Conflict-Driven Clause Learning SAT Solver



Bit-vector Solving Techniques

Bit-blast Eagerly convert bit-vectors to boolean predicate logic, then solve using the CDCL SAT solver.

Polysat Lazy bit-blasting — only bit-blast when necessary. Perform dynamic simplifications.

Int-blast Convert to integers, simulating the behaviour of bit-vectors, then solve using an integer theory solver.

SLS Keep a candidate solution and randomly mutate it until all clauses are satisfied. Works in tandem with **bit-blast**.

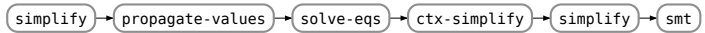
4. Parallelism in Z3

Cube-and-conquer Split problem into sub-problems (cubes) and solve independently.

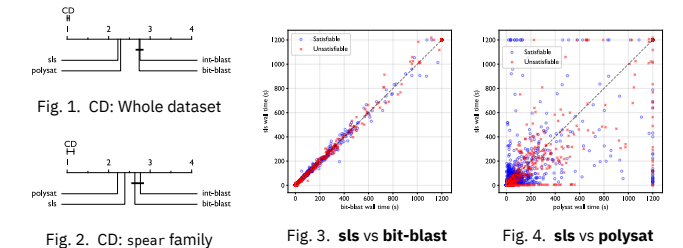
Portfolio solving Run multiple solvers on the same problem simultaneously.

Parallel SLS Run **sls** on another thread alongside **bit-blast**.

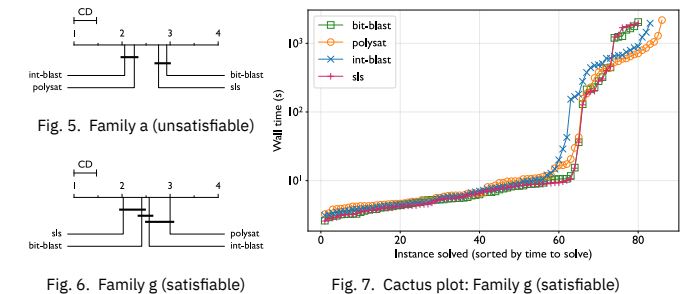
5. Experiments & Results



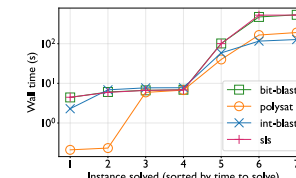
SMT-COMP Sampled from the 2024 SMT-LIB Benchmark Release



VLSAT-3 Very Large Boolean SATisfability benchmark suite



Smart Contract Verification



Parallel Scaling

