From Local Search[1] to Quantifier-Elimination[2] for Bit-Vectors in SMT

Aina Niemetz
Stanford University

joint work with Clark Barrett[2⋆], Armin Biere[1⋄], Mathias Preiner[12⋆], Andrew Reynolds[2†] and Cesare Tinelli[2†]

⋆ Stanford University † The University of Iowa ⋄ Johannes Kepler University Linz

Theory and Practice of Satisfiability Solvers
August 27-31, 2018
Oaxaca, Mexico

Theory of Fixed-Size Bit-Vectors

- constants, variables: 00000010, $2_{[8]}$, $x_{[32]}$, $y_{[2]}$
- bit-vector operators: $=$, $<$, $>$, $\sim$, &, $\ll$, $\gg$, $\circ$, [:], ...
- arithmetic operators modulo 2^n (overflow semantics!)

Bit-Blasting

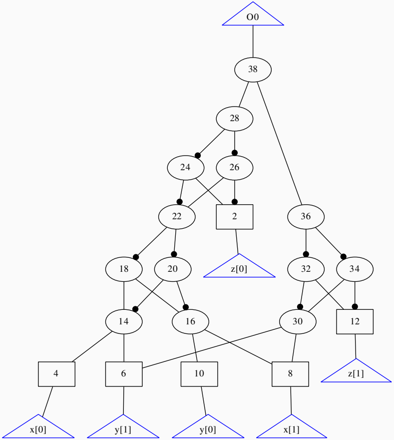
- current state-of-the-art for **quantifier-free** bit-vector formulas
- rewriting + simplifications + eager reduction to SAT

- efficient in practice
- may suffer from an exponential blow-up in the formula size

- may not scale well for increasing bit-widths

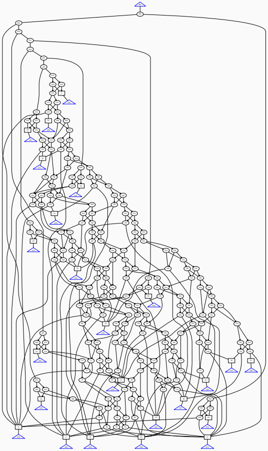
Bit-Blasting

Example $x_{[2]} * y_{[2]} = z_{[2]}$



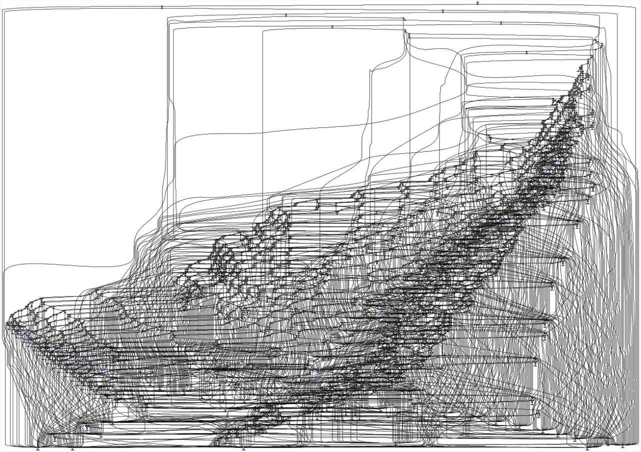
Bit-Blasting

Example $x_{[8]} * y_{[8]} = z_{[8]}$

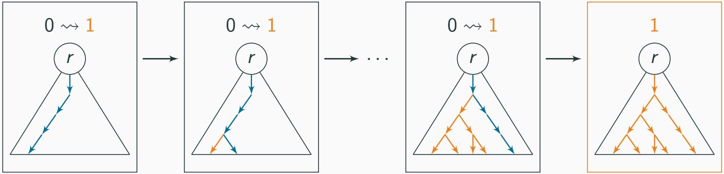


Bit-Blasting

Example $x_{[32]} * y_{[32]} = z_{[32]}$

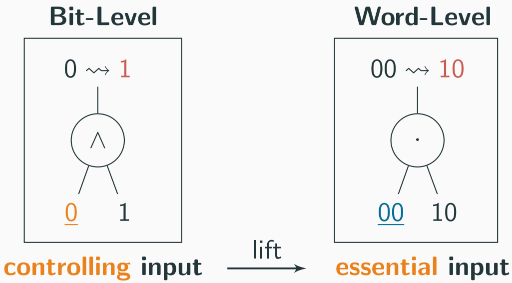


Propagation-Based Local Search for QF_BV [CAV'16]



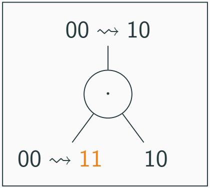
- **without** bit-blasting, **no** SAT solver (orthogonal approach)
  - assume satisfiability, start with **initial assignment**
  - **propagate** target values towards inputs
  - iteratively improve current state until **solution** is found
- **not able** to determine **unsatisfiability**
- Probabilistically Approximately Complete (PAC) [Hoos, AAAI'99]
  if there exists a **non-deterministic** choice of moves that lead to a **solution**

Propagation Path Selection



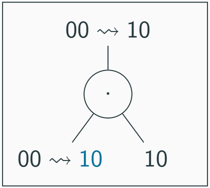
- choose controlling / essential input, else choose random input

Value Selection



inverse value

- produces target value **without** changing the value of other inputs
- **unconditional** inverse not always possible



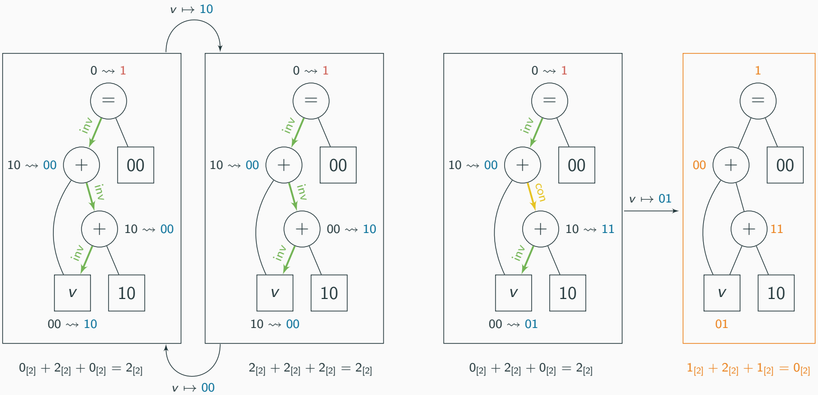
consistent value

- **less strict** notion
  - produces target value **after** changing the value of other inputs

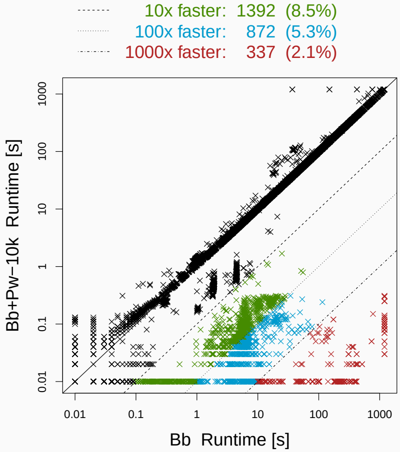
- using **only** inverse values without further randomization is incomplete

Why is using only inverse values incomplete?

Example $v_{[2]} + v_{[2]} + 2_{[2]} = 0_{[2]}$



Results



- Implemented in Boolector
- **Bb** bit-blasting engine
- **Bb+Pw-10k** sequential portfolio combination

- 16436 benchmarks from QF_BV (SMT-LIB) (sat + unknown)
- 1200s time limit

From Concrete Inverse Values to Symbolic Inverses

- **unconditional** inverse value computation not possible in general
  - if the current assignment does not satisfy the **invertibility condition** for a bit-vector operator we choose a **consistent** value

Can we utilize the concept of invertibility conditions for something else?

From Concrete Inverse Values to Symbolic Inverses

- **unconditional** inverse value computation not possible in general
  - if the current assignment does not satisfy the **invertibility condition** for a bit-vector operator we choose a **consistent** value

Can we utilize the concept of invertibility conditions for something else?

- Yes! For **quantified** bit-vector formulas!

Motivation

Example $\psi = \forall x \,.\, (x + s \not\approx t)$ x, s, t ... bit-vectors of size N

State of the Art in SMT: Quantifier instantiation-based techniques

- Find conflicting ground instances of the formula
- Crucial to find good instantiation candidates

- **Naive:** Enumerate values for x (2^N possible instantiations)
- **Better:** Instantiate with symbolic term $t - s$

$$\underbrace{(t - s) + s \not\approx t}_{\text{UNSAT}}$$

- **Idea:** Compute symbolic inverses of bit-vector operators [CAV'18]

Symbolic Inverses

- **unconditional inverses** do not always exist

Example $x \cdot s \approx t$ x, s, t ... bit-vectors of size N

- solve for x
- no inverse for, e.g., $x \cdot 2 \approx 3$
- **invertibility condition:** $((-s \mid s) \,\&\, t) \approx t$
- identifies condition under which $x \cdot s \approx t$ is invertible:

  $$((-s \mid s) \,\&\, t) \approx t \;\Leftrightarrow\; x \cdot s \approx t$$

- independent from the bit-width

Invertibility Conditions

- 162 invertibility conditions for:
  Operators: $\diamond \in \{\&, |, \ll, \gg, \gg_a, \cdot, \bmod, \div, \circ\}$
  Relations: $\bowtie \in \{\approx, \not\approx, <_u, \leq_u, >_u, \geq_u, <_s, \leq_s, >_s, \geq_s\}$
- 83 *manually*, 79 *synthesized* with SyGuS (syntax-guided synthesis)

- **SyGuS problem:** $\exists C \forall s \forall t. ((\exists x. x \diamond s \bowtie t) \Leftrightarrow C(s,t))$
- **Expand innermost** $\exists$ (4-bit): $\exists C \forall s \forall t. (\bigvee_{i=0}^{15} i \diamond s \bowtie t) \Leftrightarrow C(s,t)$

- Synthesized 118 conditions (out of 140) with CVC4
- Verified correctness of 94.6% the 162 ICs for bit-width 1 to 65 (with Boolector, CVC4, Q3B, Z3)

From Invertibility Conditions to Symbolic Instantiations

Hilbert choice function $\varepsilon x.\, \varphi[x]$

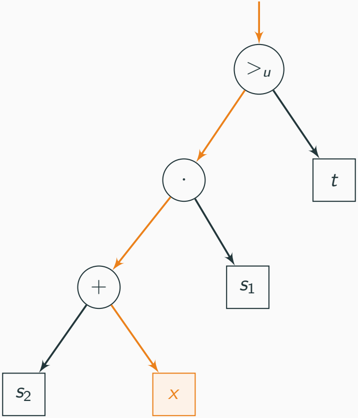
- represents a solution for $\varphi[x]$ if there is one
- and an arbitrary value otherwise
- $\exists x.\, \varphi[x] \Leftrightarrow \varphi[\varepsilon x.\, \varphi[x]]$

Embed invertibility conditions into Hilbert choice functions

- bit-vector literal: $e[x] := x \diamond s \bowtie t$
- invertibility condition: $C(s, t) \;\Leftrightarrow\; e[x]$
- symbolic term: $\varepsilon y.\, (C(s, t) \;\Rightarrow\; e[y])$

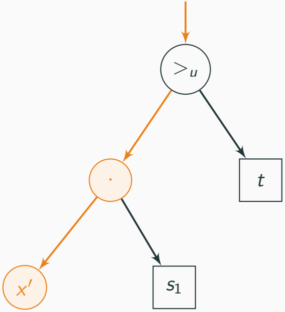
- choice functions express **all** conditional solutions in one symbolic term

Example: $\forall x. (s_2 + x) \cdot s_1 >_u t$



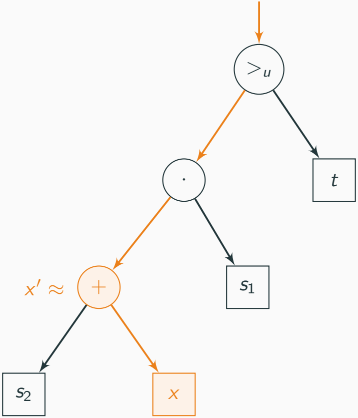
1. Pick variable to solve for (x)
2. Compute inverse/invertibility conditions along path to x

Example: $\forall x. (s_2 + x) \cdot s_1 >_u t$



1. Pick variable to solve for (x)
2. Compute inverse/invertibility conditions along path to x
3. $x' \cdot s_1 >_u t$
   - $IC_{x'} = t <_u -s \mid s$
   - $x' = \varepsilon y. (IC_{x'} \Rightarrow y \cdot s_1 >_u t)$

Example: $\forall x.\,(s_2 + x) \cdot s_1 >_u t$



1. Pick variable to solve for (x)
2. Compute inverse/invertibility conditions along path to x
3. $x' \cdot s_1 >_u t$
   - $IC_{x'} = t <_u -s \mid s$
   - $x' = \varepsilon y.\,(IC_{x'} \Rightarrow y \cdot s_1 >_u t)$
4. $s_2 + x \approx x'$
   - $IC_x = \top$
   - $x = x' - s_2$

Instantiation for x: $\varepsilon y.\,(t <_u -s \mid s \Rightarrow s_1 \cdot y >_u t) - s_2$

Multiple Variable Occurrences

Non-linear constraints (multiple occurrences of a variable)

- Try to linearize with rewriting/normalization
  e.g., $x + x + s \approx t \rightarrow 2 \cdot x + s \approx t$
- Else: Replace all but one occurrence with value in current model $\mathcal{I}$
  e.g., $x \cdot x + s \approx t \rightarrow x \cdot x^{\mathcal{I}} + s \approx t$

▶ Future work: Use SyGuS to synthesize ICs for non-linear cases

Unit linear invertible formulas

- If $\forall x.\, \varphi[x]$ is linear in x (only one occurrence of x)

▶ Quantifier elimination: reduce to quantifier-free bit-vector formula

Experiments

	$\textbf{CVC4}_{\textbf{base}}$	**Q3B**	**Boolector**	**Z3**	$\textbf{CVC4}_{\textbf{ic}}$
keymaera (4035)	3823	3805	4025	**4031**	3993
psyco (194)	**194**	99	193	193	190
scholl (374)	239	214	**289**	271	246
tptp (73)	**73**	**73**	72	**73**	**73**
uauto (284)	112	256	180	190	**274**
wintersteiger (191)	168	**184**	154	162	168
Total (5151)	4609	4631	4913	4920	**4944**

Limits: 300 seconds CPU time limit, 100G memory limit

$\textbf{CVC4}_{\textbf{ic}}$ won division BV at SMT-COMP 2018

Conclusion

- Propagation-based local search approach implemented in Boolector
  https://github.com/boolector/boolector
- Quantifier elimination approach implemented in CVC4
  https://github.com/cvc4/cvc4

References

- A. Niemetz, M. Preiner, A. Reynolds, C. Barrett and C. Tinelli. Solving Quantified Bit-Vectors Using Invertibility Conditions. In Proc. of CAV'18, pages 236–255, Springer, 2018.
- A. Niemetz, M. Preiner and A. Biere. Precise and Complete Propagation Based Local Search for Satisfiability Modulo Theories. In Proc. of CAV'16, pages 199-217, Springer, 2016.
- H. H. Hoos. *On the Run-time Behaviour of Stochastic Local Search Algorithms for SAT.* In Proc. of AAAI/IAAI'99, pages 661–666, AAAI Press / The MIT Press, 1999.