Ternary Propagation-Based Local Search for more Bit-Precise Reasoning

Aina Niemetz   Mathias Preiner

FMCAD 2020
Sept. 21-24, 2020

Stanford University

Satisfiability Modulo Theories (SMT)

Given a (quantifier-free) FOL formula and a combination of theories

$$((x \ll 01) \geq 00) \wedge (x < 01) \wedge ((\mathrm{read}(\mathrm{write}(a, x, x), x \cdot 10) = x + 01)$$

$x, 00, 01, 10 \ldots$ Bit-Vectors of size 2 $\qquad$ $a \ldots$ Array

is there an assignment to x such that this formula evaluates to **true**?

Satisfiability Modulo Theories (SMT)

Given a (quantifier-free) FOL formula and a combination of theories

$$((x \ll 01) \geq 00) \wedge (x < 01) \wedge ((\mathrm{read}(\mathrm{write}(a, x, x), x \cdot 10) = x + 01)$$

$x, 00, 01, 10 \ldots$ Bit-Vectors of size 2 $\qquad$ $a \ldots$ Array

is there an assignment to x such that this formula evaluates to **true**? No

Here: Theory of Fixed-Size Bit-Vectors (without Quantifiers)

Theory of Fixed-Size Bit-Vectors

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

Bit-Blasting

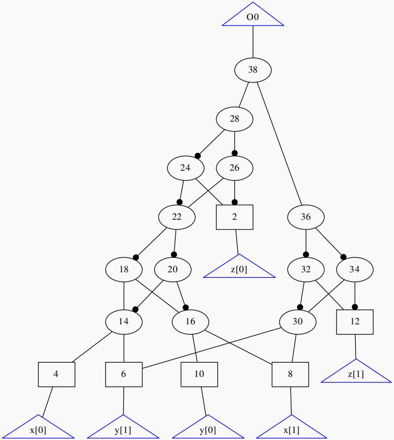
- current state-of-the-art for **quantifier-free** bit-vector formulas
- rewriting + simplifications + eager reduction to SAT

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

- ▶ may not scale well for increasing bit-widths

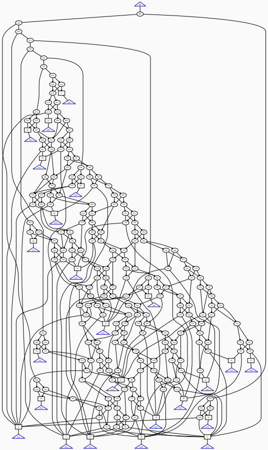
Bit-Blasting

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



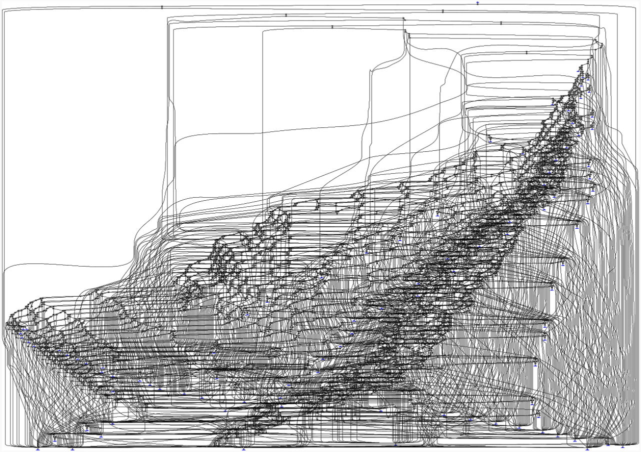
Bit-Blasting

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

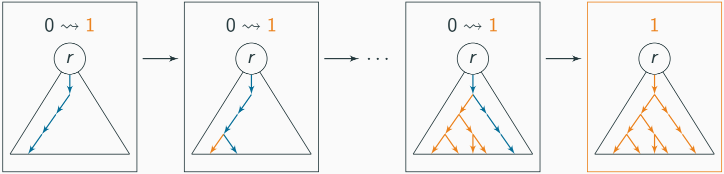


Bit-Blasting

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



Propagation-Based Local Search [CAV'16]



- ◦ assume satisfiability, start with **initial assignment**
- ◦ **propagate** target values towards inputs
- ◦ iteratively improve current state until **solution** is found

- **orthogonal approach**
- lifts concept of **backtracing** from ATPG to the **word-level**
- **without** bit-blasting, **no** SAT solver

- **not able** to determine **unsatisfiability**
- Probabilistically Approximately Complete (PAC) [Hoos, AAAI'99]
  - ▷ guaranteed to find a solution if there is one

Ternary Propagation-Based Local Search

Propagation-based Local Search [CAV'16]:

- **Main Weakness:** oblivious to **constant bits**
  - propagates target values that **can never be assumed**
  - **redundant work**

Ternary Propagation-based Local Search:

- **Generalization** with respect to **constant bits**
  - **symbolic conditions** to determine if value propagation is possible
- **Extension** with **bound-tightening heuristics** for inequalities

Ternary Propagation-Based Local Search

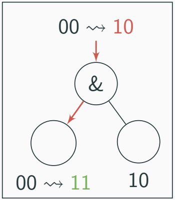
- **Non-deterministic algorithm**
  - **Main components: propagation path** and **value** selection
    - multiple possible paths and values
    - main sources of **non-determinism**
- **Down-propagation** of target values with respect to **constant bits**
  - constant bits are **precomputed** upfront
  - represented as **ternary bit-vectors**
    - **ternary bit-vector** $x = \langle x^{lo}, x^{hi} \rangle$ with $(\sim x^{lo} \mid x^{hi}) \approx \text{ones}$

      x^{lo} ... minimum (unsigned) value of x
      x^{hi} ... maximum (unsigned) value of x

      Example: $x_{[4]} = \bullet\bullet\bullet 0 = \langle 0000, 1110 \rangle$

Propagation Value Selection

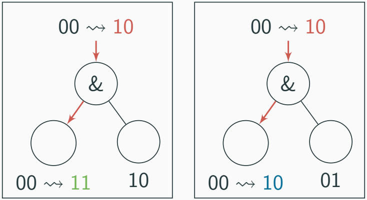
without changing other inputs



inverse value

Propagation Value Selection

without changing other inputs

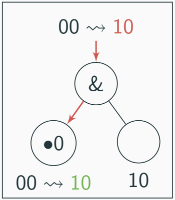


inverse value **consistent value**

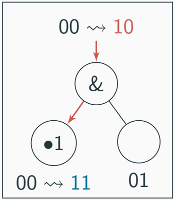
after changing other inputs

Propagation Value Selection

without changing other inputs



inverse value
wrt const bits

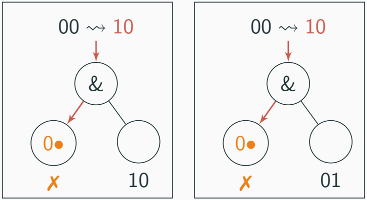


consistent value
wrt const bits

after changing other inputs

Propagation Value Selection

without changing other inputs



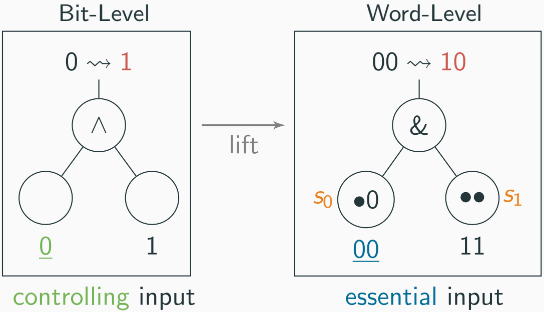
after changing other inputs

inverse value wrt const bits

consistent value wrt const bits

- inverse and consistent value **not always possible**
- **symbolic** invertibility/consistency conditions
- **break and restart** propagation if **consistency** condition **false**

Propagation Path Selection



- s_0 is essential if there **does not** exist an inverse value for s_1
- s_1 is essential if there **does not** exist an inverse value for s_0

Invertibility and Consistency Conditions

Given $x \diamond s \approx t$ with $x = \langle x^{lo}, x^{hi} \rangle$ **ternary**, and s, t **binary** bit-vectors.

$mcb(x, y)$ ... **true** if constant bits in x match corresponding bits in y

Invertibility Condition (IC).

$\forall x, s, t.\,(IC(x, s, t) \Leftrightarrow \exists y.\,(y \diamond s \approx t \wedge mcb(x, y)))$

Consistency Condition (CC).

$\forall x, t.\,(CC(x, t) \Leftrightarrow \exists y, s.\,(y \diamond s \approx t \wedge mcb(x, y)))$

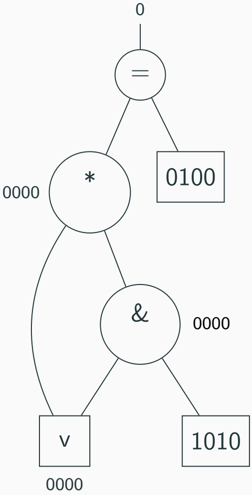
Example. $x \,\&\, s \approx t$

$IC(x \,\&\, s \approx t) = t \,\&\, s \approx t \wedge ((s \,\&\, x^{hi}) \,\&\, {\sim}(x^{lo} \oplus x^{hi})) \approx (t \,\&\, {\sim}(x^{lo} \oplus x^{hi}))$

$CC(x \,\&\, s \approx t) = t \,\&\, x^{hi} \approx t$

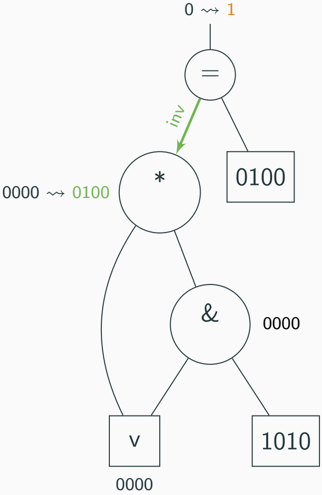
Ternary Propagation-Based Local Search

Example. $v_{[4]} \cdot (v_{[4]} \ \& \ 1010) \approx 0100$



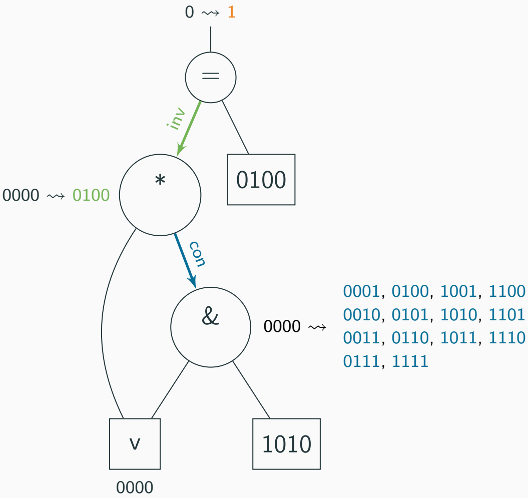
Ternary Propagation-Based Local Search

Example. $v_{[4]} \cdot (v_{[4]} \mathbin{\&} 1010) \approx 0100$



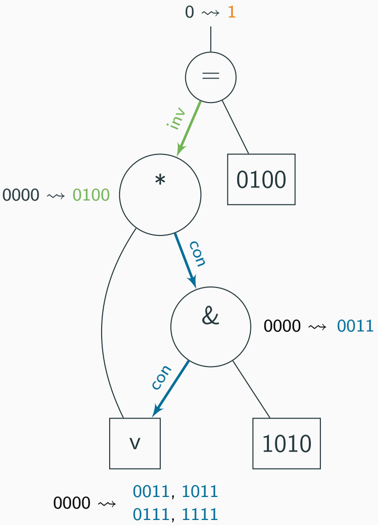
Ternary Propagation-Based Local Search

Example. $v_{[4]} \cdot (v_{[4]} \;\&\; 1010) \approx 0100$



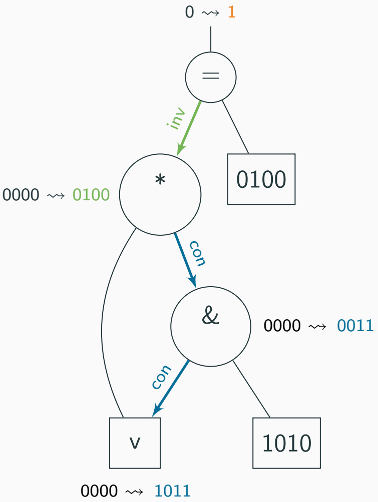
Ternary Propagation-Based Local Search

Example. $v_{[4]} \cdot (v_{[4]} \,\&\, 1010) \approx 0100$



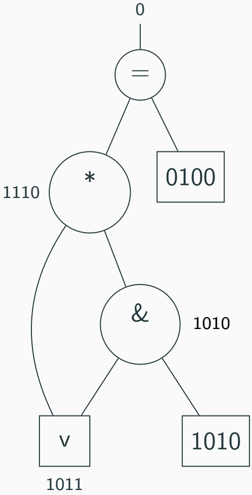
Ternary Propagation-Based Local Search

Example. $v_{[4]} \cdot (v_{[4]} \mathbin{\&} 1010) \approx 0100$



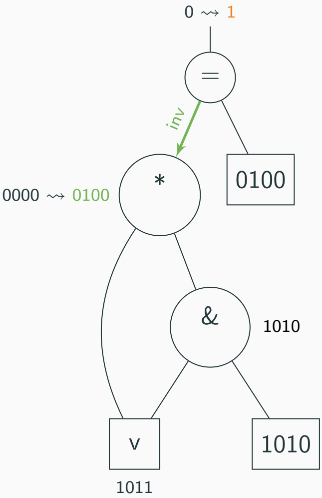
Ternary Propagation-Based Local Search

Example. $v_{[4]} \cdot (v_{[4]} \,\&\, 1010) \approx 0100$



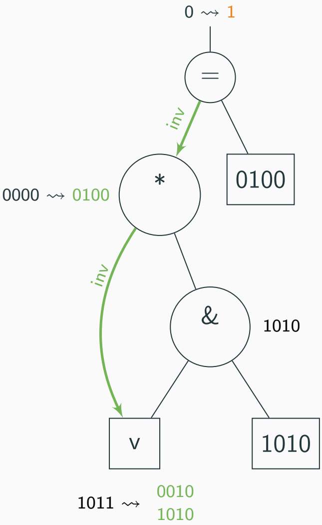
Ternary Propagation-Based Local Search

Example. $v_{[4]} \cdot (v_{[4]} \,\&\, 1010) \approx 0100$

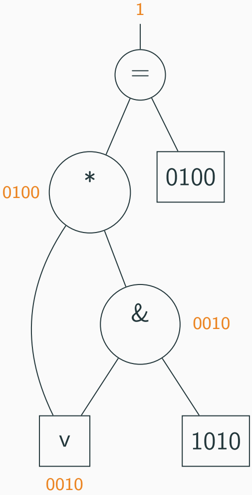


Ternary Propagation-Based Local Search

Example. $v_{[4]} \cdot (v_{[4]} \mathbin{\&} 1010) \approx 0100$

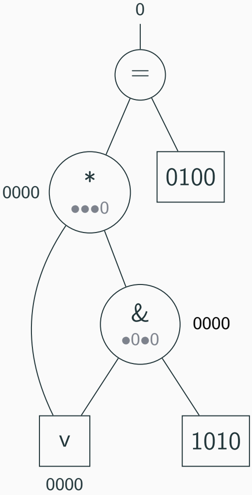


Example. $v_{[4]} \cdot (v_{[4]} \,\&\, 1010) \approx 0100$

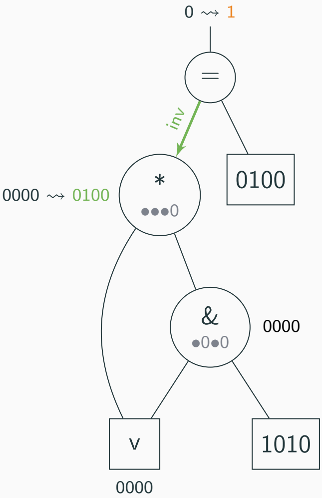


Ternary Propagation-Based Local Search

Example. $v_{[4]} \cdot (v_{[4]}\ \&\ 1010) \approx 0100$

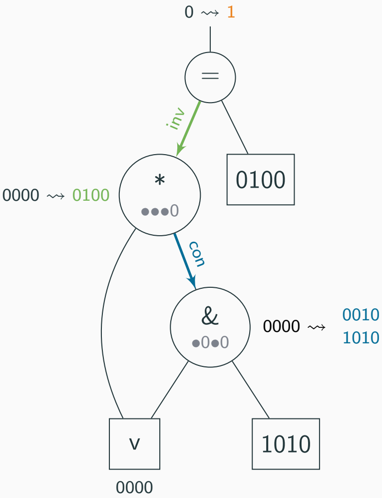


Example. $v_{[4]} \cdot (v_{[4]} \ \& \ 1010) \approx 0100$



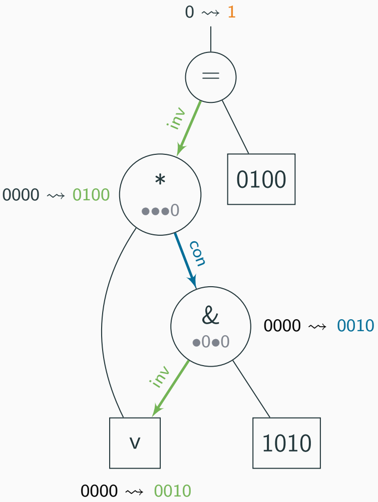
Ternary Propagation-Based Local Search

Example. $v_{[4]} \cdot (v_{[4]} \,\&\, 1010) \approx 0100$



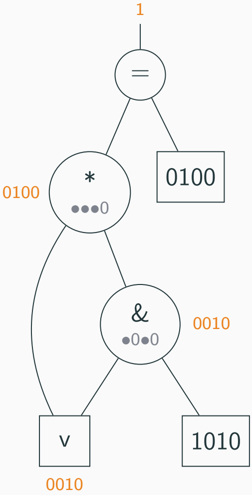
Ternary Propagation-Based Local Search

Example. $v_{[4]} \cdot (v_{[4]} \,\&\, 1010) \approx 0100$



Ternary Propagation-Based Local Search

Example. $v_{[4]} \cdot (v_{[4]} \,\&\, 1010) \approx 0100$



Extension: Bound Tightenting Heuristics

Example. $x \leq_u 0111 \wedge x >_u 0101$

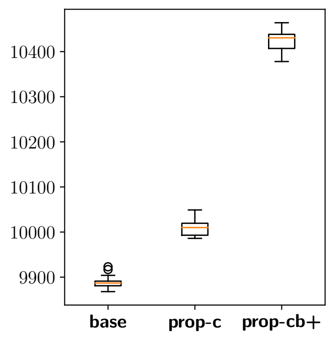
- $x \leq_u 0111$: $[0000, 0111]$
- $x >_u 0101$: $[0110, 1111]$
- actual: $[0110, 0111]$

- Heuristic for **inverse value** computation

- compute **bounds** for x in $x \diamond s$ ($s \diamond x$) over inequalities $\diamond \in \{<_s, <_u\}$
- based on **currently satisfied top-level inequalities** on x
  - **bounds invalid:** no inverse value for x
- **special handling** of $x\langle n \rangle <_u s$ and $x\langle n \rangle <_s s$
  - derive bounds based on prefix ($0_{[n]}$ or $\text{ones}_{[n]}$)

Results

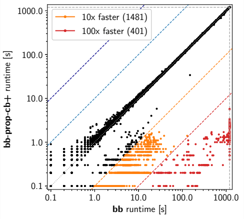
- implemented in our **new** SMT solver **Bitwuzla**



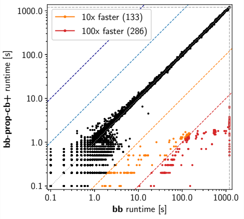
- **base** Propagation-based LS [CAV'16]
- **prop-c** **Ternary propagation-based LS**
  - + 120 (median) instances vs. **base**
  - 70% less moves
  - 63% less propagations
- **prop-cb+** **prop-c** with bound-tightening
  - + 400 (median) instances vs. **prop-c**

- 14,382 QF_BV **sat** instances in SMT-LIB
- 20 runs with different seeds for RNG
- 60s time limit, 8GB memory limit

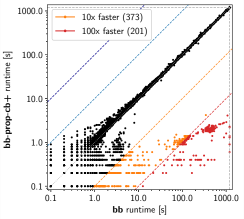
Results



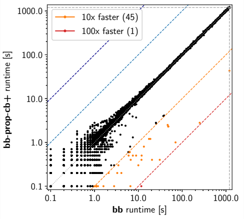
Lingeling SAT back end



CryptoMiniSat SAT back end



Kissat SAT back end
SAT competition winner 2020



CaDiCaL SAT back end

- **sequential portfolio** (first run LS, then fall back to bit-blasting)
- **all** 41,713 benchmarks in SMT-LIB QF_BV
- 1200s time limit, 8GB memory limit
- winner of division QF_BV in the SMT-COMP 2020

Conclusion

- **Generalization** of propagation-based local search **with respect to constant bits**
  - reduces redundant work
  - improves performance
- Determined **symbolic invertibility** and **consistency conditions**
  - for a representative set of bit-vector operators
  - verified up to size 65
  - **Future work:** proofs of correctness
- **Bound-Tightenting heuristic**
  - **Future Work:** apply in general, not only for inequality literals
- **Future work:** Hybrid approach
  - share information between bit-blasting and local search

References

- 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.