ON INTERVALS AND BOUNDS IN BIT-VECTOR ARITHMETIC

Mikoláš Janota and Christoph M. Wintersteiger

Microsoft Research, Cambridge, UK

What are bit vectors?

- numbers as in computer (roughly)

What are bit vectors?

- numbers as in computer (roughly)
- fixed bit-width

WHAT ARE BIT VECTORS?

- numbers as in computer (roughly)
- fixed bit-width
- $+$, $\times$ wraps around

WHAT ARE BIT VECTORS?

- numbers as in computer (roughly)
- fixed bit-width
- $+$, $\times$ wraps around
- negative numbers via 2's complement

WHAT ARE BIT VECTORS?

- numbers as in computer (roughly)
- fixed bit-width
- $+$, $\times$ wraps around
- negative numbers via 2's complement
- Example: $(x_8 +_8 1_8) \leq^s_8 (y_8 \times_8 3_8)$

WHAT ARE BIT VECTORS?

- numbers as in computer (roughly)
- fixed bit-width
- $+$, $\times$ wraps around
- negative numbers via 2's complement
- Example: $(x_8 +_8 1_8) \leq^s_8 (y_8 \times_8 3_8)$

- $x_8 = 0x7f, y_8 = 0x80$

How do we solve bit-vectors?

- Bit-blasting — convert everything to propositional form (SAT).

How do we solve bit-vectors?

- Bit-blasting — convert everything to propositional form (SAT).
- Exponential in bit-width and losing “domain” knowledge.

How do we solve bit-vectors?

- Bit-blasting — convert everything to propositional form (SAT).
- Exponential in bit-width and losing "domain" knowledge.
- It is important to apply preprocessing before sending to SAT.

How do we solve bit-vectors?

- Bit-blasting — convert everything to propositional form (SAT).
- Exponential in bit-width and losing "domain" knowledge.
- It is important to apply preprocessing before sending to SAT.
- Example: $(x_m + 0_m) = x_m$

How do we solve bit-vectors?

- Bit-blasting — convert everything to propositional form (SAT).
- Exponential in bit-width and losing "domain" knowledge.
- It is important to apply preprocessing before sending to SAT.
- Example: $(x_m + 0_m) = x_m$
- Example: $(x_m \times 4_m) = x[m-3:0] +\!\!+ 0_2$

PROBLEM

- inequalities with multiple variables and addition are NP-complete

PROBLEM

- inequalities with multiple variables and addition are NP-complete
- conjunction $\bigwedge \neg C_i \wedge \bigwedge C_i$, where C_i is one of the following with x a bit-vector variable and c_1, c_2 constants.

- inequalities with multiple variables and addition are NP-complete
- conjunction $\bigwedge \neg C_i \wedge \bigwedge C_i$, where C_i is one of the following with x a bit-vector variable and c_1, c_2 constants.
  1. $(c_1 +_w x) \leq^u (c_2 +_w x)$

PROBLEM

- inequalities with multiple variables and addition are NP-complete
- conjunction $\bigwedge \neg C_i \wedge \bigwedge C_i$, where C_i is one of the following with x a bit-vector variable and c_1, c_2 constants.
  1. $(c_1 +_w x) \leq^u (c_2 +_w x)$
  2. $c_1 \leq^u (c_2 +_w x)$

- inequalities with multiple variables and addition are NP-complete
- conjunction $\bigwedge \neg C_i \wedge \bigwedge C_i$, where C_i is one of the following with x a bit-vector variable and c_1, c_2 constants.
  1. $(c_1 +_w x) \leq^u (c_2 +_w x)$
  2. $c_1 \leq^u (c_2 +_w x)$
  3. $(c_1 +_w x) \leq^u c_2$

PROBLEM

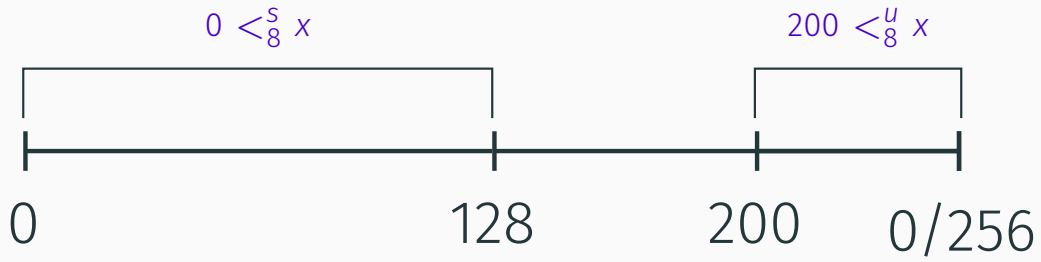
- inequalities with multiple variables and addition are NP-complete
- conjunction $\bigwedge \neg C_i \wedge \bigwedge C_i$, where C_i is one of the following with x a bit-vector variable and c_1, c_2 constants.
  1. $(c_1 +_w x) \leq^u (c_2 +_w x)$
  2. $c_1 \leq^u (c_2 +_w x)$
  3. $(c_1 +_w x) \leq^u c_2$
  4. $x \leq^s c_1$

PROBLEM

- inequalities with multiple variables and addition are NP-complete
- conjunction $\bigwedge \neg C_i \wedge \bigwedge C_i$, where C_i is one of the following with x a bit-vector variable and c_1, c_2 constants.
  1. $(c_1 +_w x) \leq^u (c_2 +_w x)$
  2. $c_1 \leq^u (c_2 +_w x)$
  3. $(c_1 +_w x) \leq^u c_2$
  4. $x \leq^s c_1$
  5. $c_1 \leq^s x$

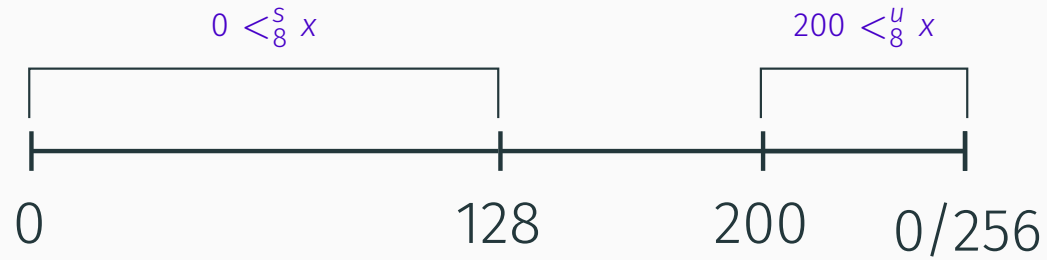
INEQUALITIES

- $(0 <_8^s x) \wedge (200 <_8^u x)$ ... UNSAT

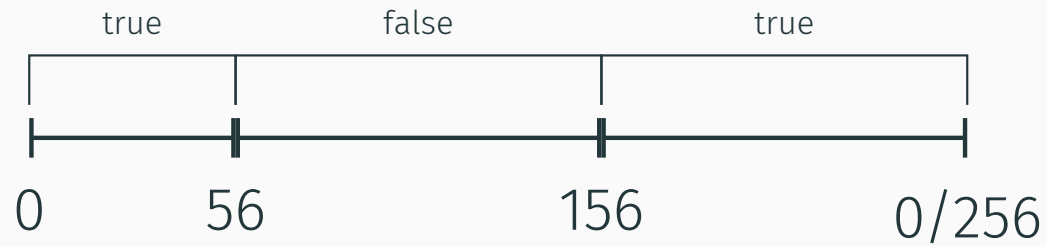


INEQUALITIES

- $(0 <_8^s x) \wedge (200 <_8^u x)$ ... UNSAT

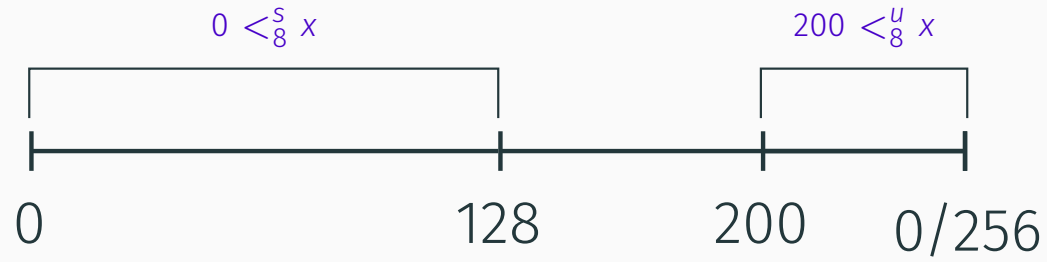


- $(x + 100 <_8^u x + 200)$

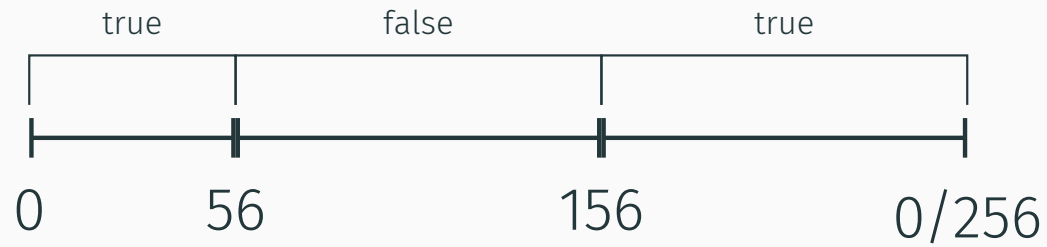


INEQUALITIES

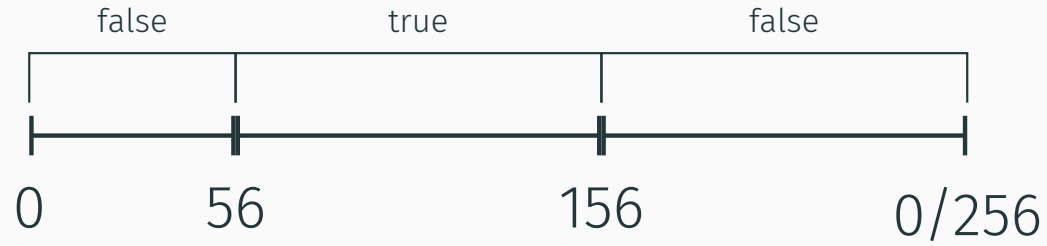
- $(0 <_8^s x) \wedge (200 <_8^u x)$ ... UNSAT



- $(x + 100 <_8^u x + 200)$



- $(x + 200 <_8^u x + 100)$



Expression	Condition	Interval
$c_1 +_w x \leq^u c_2 +_w x$	$c_1 \leq c_2$	$\backsim[-c_2\,;\,-c_1-1]$
$c_1 +_w x \leq^u c_2 +_w x$	$c_1 > c_2$	$[-c_1\,;\,-c_2-1]$
$c_1 \leq^u c_2 +_w x$	$c_1 < c_2$	$\backsim[-c_2\,;c_1-c_2-1]$
$c_1 \leq^u c_2 +_w x$	$c_1 \geq c_2$	$[c_1-c_2\,;\,-c2-1]$
$c_1 +_w x \leq^u c_2$	$c_1 \leq c_2$	$\backsim[c_2-c_1+1\,;\,-c_1-1]$
$c_1 +_w x \leq^u c_2$	$c_1 > c_2$	$[-c_1\,;\,-c_1+c_2]$
$x \leq^s c_1$	$c_1 < 2^{w-1}$	$\backsim[c_1+1\,;2^{w-1}-1]$
$x \leq^s c_1$	$c_1 \geq 2^{w-1}$	$[2^{w-1}\,;c_1]$
$c_1 \leq^s x$	$c_1 < 2^{w-1}$	$[c_1\,;2^{w-1}-1]$
$c_1 \leq^s x$	$c_1 \geq 2^{w-1}$	$\backsim[2^{w-1}\,;c_1-1]$

COMPUTING THE ENVELOPE

1 $\mathcal{P} \leftarrow \{[a\,;b] \mid [a\,;b] \in \mathcal{I}\}$; // positive intervals
2 $l \leftarrow \mathcal{P} = \emptyset\,?\,0 : \mathtt{min}\,\{a \mid [a\,;b] \in \mathcal{P}\}$; // lower bound
3 $h \leftarrow \mathcal{P} = \emptyset\,?\,2^w - 1 : \mathtt{max}\,\{b \mid [a\,;b] \in \mathcal{P}\}$; // upper bound
4 $\mathcal{N} \leftarrow \{\backsim[a\,;b] \mid \backsim[a\,;b] \in \mathcal{I}\}$; // negative intervals
5 $\mathcal{N} \leftarrow$ sort $\mathcal{N}$ by first element;
6 $p, l', h' \leftarrow l, l, l - 1$;
7 **for** $\backsim[a\,;b] \in \mathcal{N} \cup \backsim[2^w\,;2^w]$ **do**
8 **if** $p > h$ **then break**; // space exhausted
9 **if** $b < p$ **then continue**; // redundant interval
10 **if** $p < a$ **then** // satisfiable portion
11 **if** $h' > l'$ **then** $l' \leftarrow p$; // first satisfiable point
12 $h' \leftarrow a - 1$; // update upper bound
13 $p \leftarrow b + 1$; // move onto next portion
14 **return** $[l'\,;h']$

Table 1: Conflict count.

Example	Avg.	Med	Min	Max
(1) unsat	26	36	3	100
(2) redundant	31	25	7	89
(2) reduced	29	37	3	92
(3) unique	21	40	0	123

REMARKS

- Single-variable inequalities appear in tests for overflows.

REMARKS

- Single-variable inequalities appear in tests for overflows.
- As simple preprocessing do not seem to be very effective.

REMARKS

- Single-variable inequalities appear in tests for overflows.
- As simple preprocessing do not seem to be very effective.
- Context sensitive? During search?

Thank You for Your Attention!

Questions?