Verifying Bit-vector Invertibility Conditions in Coq (Extended Abstract)

Burak Ekici, Arjun Viswanathan, Yoni Zohar, Clark Barrett, Cesare Tinelli

350 universität innsbruck
Wir bauen Brücken. Seit 1669

The University of Iowa

Stanford University

Bit-vectors

Bit-vectors

- Bit-vectors: Fixed-width bit sequences

    10101

    $a \in BV_5$

Bit-vectors

- Bit-vectors: Fixed-width bit sequences

  10101

  $a \in BV_5$

- Bit-vector operations

  $a + b$

  $a <_u b$

  …

Introduction

Bit-vectors have many applications:

- Hardware circuit analysis [Gupta et al., 1993]
- Bounded model checking [Armando et al., 2006]
- Symbolic execution [Cadar et al., 2006]
- ...

Introduction

- Many applications require quantified bit-vector formulas
- Some SMT solvers use quantifier-instantiation to solve quantified formulas
- *Invertibility conditions* are a useful meta-construct for a quantifier-instantiation technique [Niemetz et al., CAV 2018]

Invertibility Conditions

An *invertibility condition* for a variable x in a bit-vector literal

$$\ell\,[\,x\,,\;s\,,\;t\,]$$

is a formula

$$IC\,[\,s\,,\;t\,]$$

s.t. the following *invertibility equivalence* is valid in the theory of bit-vectors:

$$\forall s.\ \forall t.\ IC[s,t] \iff \exists x.\ \ell[x,s,t]$$

where s, t, x : BV_n

Invertibility Conditions: Example

Invertibility Conditions: Example

- Inversion of bit-vector addition is unconditional

$$\exists x.\ x + s = t \iff \top$$

The *inverse* is $x = t - s$

Invertibility Conditions: Example

- Inversion of bit-vector addition is unconditional

$$\exists x.\ x + s = t \iff \top$$

The *inverse* is $x = t - s$

- Inversion of bit-wise conjunction is conditional

$$\exists x.\ x \;\&\; s = t \iff t \;\&\; s = t$$

Motivation

This technique [Niemetz et al., CAV 2018] requires the equivalences to be true independent of bit-width

Proofs of these equivalences are required for the soundness of the technique

…and the solvers that use it

Previous Work

Previous Work

[Niemetz et al., CAV 2018]

- generated 162 invertibility equivalences
- proved them using SMT-solvers for bit-widths up to 65

Previous Work

[Niemetz et al., CAV 2018]

- generated 162 invertibility equivalences
- proved them using SMT-solvers for bit-widths up to 65

[Niemetz et al., CADE 2019]

- encoded the equivalences in theories supported by SMT-solvers
- verified equivalences for parametric bit-widths
- approach succeeded on under 75% of the equivalences

Contributions

Contributions

1. Formalized a representative subset of the 162 invertibility equivalences in Coq

Contributions

1. Formalized a representative subset of the 162 invertibility equivalences in Coq

2. Extended a Coq bit-vector library to support these equivalences

Contributions

1. Formalized a representative subset of the 162 invertibility equivalences in Coq

2. Extended a Coq bit-vector library to support these equivalences

3. Proved 18 of them for arbitrary bit-width

Result Summary

$\ell[x]$	$=$	$\neq$	$<_u$	$>_u$	$\leq_u$	$\geq_u$
$-x \bowtie t$	✓✓	✓	✓	✓	✓	✓
$\sim x \bowtie t$	✓✓	✓	✓	✓	✓	✓
$x \,\&\, s \bowtie t$	✓	✓	✓	✓	✓	✓
$x \mid s \bowtie t$	✓	✓	✓	✓	✓	✓
$x \ll s \bowtie t$	✓	✓	✓	✓	✓	✓
$s \ll x \bowtie t$	✓✓	✓	✓	✓	✓	✓
$x \gg s \bowtie t$	✓✓	✓	✓	✗	✓	✓
$s \gg x \bowtie t$	✓✓	✓	✓	✓	✓	✓
$x \gg_a s \bowtie t$	✓	✓	✓	✓	✓	✓
$s \gg_a x \bowtie t$	✓✓	✓	✓	✓	✓	✓
$x + s \bowtie t$	✓✓	✓	✓	✓	✓	✓

Result Summary (SMT)

$\ell[x]$	$=$	$\neq$	$<_u$	$>_u$	$\leq_u$	$\geq_u$
$-x \bowtie t$	✓	✓	✓	✓	✓	✓
$\sim x \bowtie t$	✓	✓	✓	✓	✓	✓
$x \mathbin{\&} s \bowtie t$		✓	✓	✓	✓	✓
$x \mid s \bowtie t$		✓	✓	✓	✓	✓
$x \ll s \bowtie t$			✓		✓	
$s \ll x \bowtie t$	✓	✓	✓	✓	✓	✓
$x \gg s \bowtie t$	✓	✓	✓		✓	✓
$s \gg x \bowtie t$	✓	✓	✓	✓	✓	✓
$x \gg_a s \bowtie t$		✓	✓	✓	✓	✓
$s \gg_a x \bowtie t$	✓	✓				
$x + s \bowtie t$	✓	✓	✓	✓	✓	✓

Result Summary (Coq)

$\ell[x]$	$=$	$\neq$	$<_u$	$>_u$	$\leq_u$	$\geq_u$
$-x \bowtie t$	✓					
$\sim x \bowtie t$	✓					
$x \;\&\; s \bowtie t$	✓					
$x \mid s \bowtie t$	✓					
$x \ll s \bowtie t$	✓	✓		✓		✓
$s \ll x \bowtie t$	✓					
$x \gg s \bowtie t$	✓			✗		
$s \gg x \bowtie t$	✓					
$x \gg_a s \bowtie t$	✓					
$s \gg_a x \bowtie t$	✓		✓	✓	✓	✓
$x + s \bowtie t$	✓					

Result Summary (Both)

$\ell[x]$	$=$	$\neq$	$<_u$	$>_u$	$\leq_u$	$\geq_u$
$-x \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$\sim x \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$x \mathbin{\&} s \bowtie t$	$\color{blue}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$x \mid s \bowtie t$	$\color{blue}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$x \ll s \bowtie t$	$\color{blue}{\checkmark}$	$\color{blue}{\checkmark}$	$\color{green}{\checkmark}$	$\color{blue}{\checkmark}$	$\color{green}{\checkmark}$	$\color{blue}{\checkmark}$
$s \ll x \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$x \gg s \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{red}{\times}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$s \gg x \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$x \gg_a s \bowtie t$	$\color{blue}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$s \gg_a x \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{blue}{\checkmark}$	$\color{blue}{\checkmark}$	$\color{blue}{\checkmark}$	$\color{blue}{\checkmark}$
$x + s \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$

$\color{blue}{\checkmark}$ Verified in Coq

$\color{green}{\checkmark}$ Verified in SMT

$\color{blue}{\checkmark}\color{green}{\checkmark}$ Verified in Coq and SMT

$\color{red}{\times}$ Verified in neither Coq nor SMT

Bit-vector Library

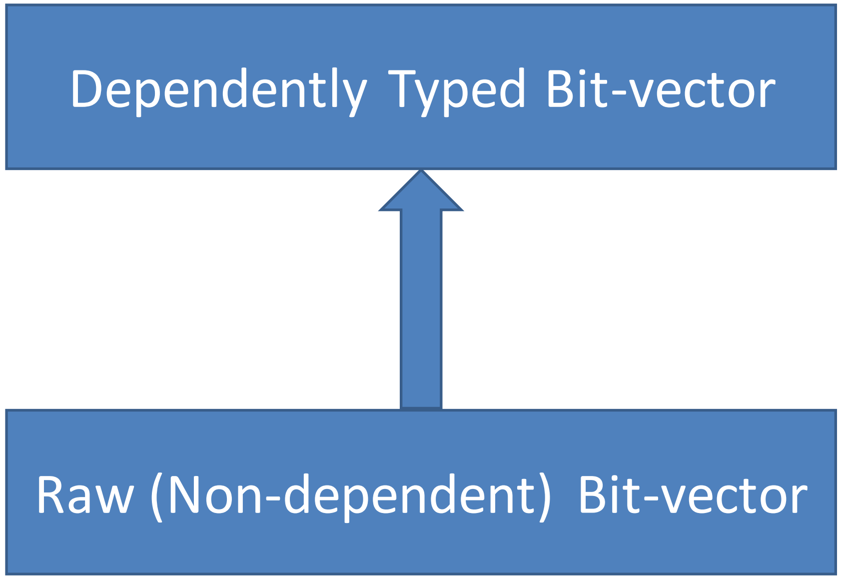
Bit-vector Library

- Used a bit-vector library originally developed for SMTCoq [Ekici et al., 2017]
- SMTCoq is a Coq plugin that uses external SMT solvers to complete proof goals
- Bit-vectors are represented as lists of Booleans

Bit-vector Library

- Used a bit-vector library originally developed for SMTCoq [Ekici et al., 2017]
- SMTCoq is a Coq plugin that uses external SMT solvers to complete proof goals
- Bit-vectors are represented as lists of Booleans

Modules



Types

bitvector : N -> Type

Functor: Raw2Bitvector

bitvector : Type
size : bitvector -> N

Bit-vector Representations

	SMTLib [Niemetz et al. 18]	Encoding [Niemetz et al. 19]	Coq Library
Bit-vector Representation:	Bit-vector of width n One sort for each n	Bit-vector of width n Translated to NIA and UF	Bit-vector of width n List of Booleans over 2 layers
Expressivity:	n cannot be symbolic	Allows quantification over n	Bit-vectors dependent over n
Verification:	Automatic proofs using SMT solvers	Automatic proofs using SMT solvers	Manual proofs in Coq
Results	All equivalences for n = 1 to 65	Verified ≈75% of equivalences	18 equivalences

Bit-vector Library

Basic signature (previous work):

$+$	addition	$\circ$	concatenation
$-$	negation	$=$	equality
$\bullet$	multiplication	$\neq$	disequality
$\&$	bit-wise conjunction	$<_u$	unsigned less than
$\mid$	bit-wise disjunction	$>_u$	unsigned greater than
$\sim$	bit-wise negation	$<_s$	signed less than
$<<$	logical left shift	$>_s$	signed greater than
$>>$	logical right shift		

Extended signature (this work):

$\leq_u$	unsigned weak less than	$\underline{<<}$	logical left shift (alt. def.)
$\geq_u$	unsigned weak greater than	$\underline{>>}$	logical right shift (alt. def.)
$>>_a$	arithmetic right shift	$\underline{>>}_a$	arithmetic right shift (alt. def.)

Original Shift Definition

```
Definition shl_one_bit (a: list bool) : list bool :=
  match a with
    | [] => []
    | _ => false :: removelast a
  end.

Fixpoint shl_n_bits (a: list bool) (n: nat): list bool :=
  match n with
    | O => a
    | S n' => shl_n_bits (shl_one_bit a) n'
  end.

Definition shl_aux (a b: list bool): list bool :=
 shl_n_bits a (list2nat_be_a b).
```

Shift Redefined

```
Definition shl_n_bits_a (a: list bool) (n: nat): list bool :=
  if (n <? length a)%nat then
    mk_list_false n ++ firstn (length a - n) a
  else
    mk_list_false (length a).

Definition bv_shl_a (a b: bitvector) : bitvector :=
  if ((@size a) =? (@size b)) then
    shl_n_bits_a a (list2nat_be_a b)
  else
    nil.
```

Invertibility Conditions Proofs

Result Summary

$\ell[x]$	$=$	$\neq$	$<_u$	$>_u$	$\leq_u$	$\geq_u$
$-x \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$\sim x \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$x \ \& \ s \bowtie t$	$\color{blue}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$x \mid s \bowtie t$	$\color{blue}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$x \ll s \bowtie t$	$\color{blue}{\checkmark}$	$\color{blue}{\checkmark}$	$\color{green}{\checkmark}$	$\color{blue}{\checkmark}$	$\color{green}{\checkmark}$	$\color{blue}{\checkmark}$
$s \ll x \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$x \gg s \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{red}{\times}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$s \gg x \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$x \gg_a s \bowtie t$	$\color{blue}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$s \gg_a x \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{blue}{\checkmark}$	$\color{blue}{\checkmark}$	$\color{blue}{\checkmark}$	$\color{blue}{\checkmark}$
$x + s \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$

$\color{blue}{\checkmark}$ Verified in Coq

$\color{green}{\checkmark}$ Verified in SMT

$\color{blue}{\checkmark}\color{green}{\checkmark}$ Verified in Coq and SMT

$\color{red}{\times}$ Verified in neither Coq nor SMT

Result Summary

$\ell[x]$	$=$	$\neq$	$<_u$	$>_u$	$\leq_u$	$\geq_u$
$-x \bowtie t$	$\textcolor{blue}{\checkmark}\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$
$\sim x \bowtie t$	$\textcolor{blue}{\checkmark}\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$
$x \ \& \ s \bowtie t$	$\textcolor{blue}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$
$x \mid s \bowtie t$	$\textcolor{blue}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$
$x \ll s \bowtie t$	$\textcolor{blue}{\checkmark}$	$\textcolor{blue}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{blue}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{blue}{\checkmark}$
$s \ll x \bowtie t$	$\textcolor{blue}{\checkmark}\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$
$x \gg s \bowtie t$	$\textcolor{blue}{\checkmark}\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{red}{\otimes}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$
$s \gg x \bowtie t$	$\textcolor{blue}{\checkmark}\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$
$x \gg_a s \bowtie t$	$\textcolor{blue}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$
$s \gg_a x \bowtie t$	$\textcolor{blue}{\checkmark}\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{blue}{\checkmark}$	$\textcolor{blue}{\checkmark}$	$\textcolor{blue}{\checkmark}$	$\textcolor{blue}{\checkmark}$
$x + s \bowtie t$	$\textcolor{blue}{\checkmark}\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$	$\textcolor{green}{\checkmark}$

$\textcolor{blue}{\checkmark}$ Verified in Coq

$\textcolor{green}{\checkmark}$ Verified in SMT

$\textcolor{blue}{\checkmark}\textcolor{green}{\checkmark}$ Verified in Coq and SMT

$\textcolor{red}{\times}$ Verified in neither Coq nor SMT

Result Summary

$\ell[x]$	$=$	$\neq$	$<_u$	$>_u$	$\leq_u$	$\geq_u$
$-x \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$\sim x \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$x \ \& \ s \bowtie t$	$\color{blue}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$x \mid s \bowtie t$	$\color{blue}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$x \ll s \bowtie t$	$\color{blue}{\checkmark}$	$\color{blue}{\checkmark}$	$\color{green}{\checkmark}$	$\color{blue}{\checkmark}$	$\color{green}{\checkmark}$	$\color{blue}{\checkmark}$
$s \ll x \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$x \gg s \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{red}{\otimes}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$s \gg x \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$x \gg_a s \bowtie t$	$\color{blue}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$
$s \gg_a x \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{blue}{\checkmark}$	$\color{blue}{\checkmark}$	$\color{blue}{\checkmark}$	$\color{blue}{\checkmark}$
$x + s \bowtie t$	$\color{blue}{\checkmark}\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$	$\color{green}{\checkmark}$

$$\texttt{bvshr\_ugt\_rtl}: \forall n.\ \forall x, s, t : BV_n.\ (x \gg s) <_u t \ \rightarrow\ t <_u (\sim s \gg s)$$

- $\color{blue}{\checkmark}$ Verified in Coq
- $\color{green}{\checkmark}$ Verified in SMT
- $\color{blue}{\checkmark}\color{green}{\checkmark}$ Verified in Coq and SMT
- $\color{red}{\times}$ Verified in neither Coq nor SMT

Challenge

Challenge

Consider bit-vectors s where $k < l$.
For $l = 4$,

s	k	l – k
0000	0	4
0001	1	3
0010	2	2
0011	3	1

$k = toNat(s)$
$l = length(s)$

Challenge

Consider bit-vectors s where $k < l$.
For $l = 4$,

s	k	l – k
0000	0	4
0001	1	3
0010	2	2
0011	3	1

$$k = toNat(s)$$
$$l = length(s)$$

Challenge

Consider bit-vectors s where $k < l$.
For $l = 4$,

s	k	l – k
0000	0	4
0001	1	3
0010	2	2
0011	3	1

$k = toNat(s)$
$l = length(s)$

More generally,

$\texttt{msb\_zero:} \quad \forall n.\ \forall s : BV_n.\ k < l \rightarrow s[(l-1)...k] = [0...0]$

Challenge

Consider bit-vectors s where $k < l$.
For $l = 4$,

s	k	l – k
0000	0	4
0001	1	3
0010	2	2
0011	3	1

$k = toNat(s)$
$l = length(s)$

More generally,

$\texttt{msb\_zero:} \quad \forall n.\ \forall s : BV_n.\ k < l \rightarrow s[(l-1)...k] = [0...0]$

$\texttt{bvshr\_ugt\_rtl:} \quad \forall n.\ \forall x, s, t : BV_n.\ (x \gg s) <_u t \rightarrow t <_u (\sim s \gg s)$

reduces to

$\texttt{msb\_zero}$

Proof Sketch of msb_zero

`msb_zero:` $\forall s : BV_n.\ k < l \rightarrow s[(l-1)...k] = [0...0]$

Proof Sketch:

$$\begin{array}{lcccccccc} & & l-1 & & k & k-1 & & 0 & \\ \text{s:} & [& 0 & \dots & 0 & \_ & \dots & \_ &] \end{array}$$

For $l = 4$,

s	k	l – k
0000	0	4
0001	1	3
0010	2	2
0011	3	1

Proof Sketch of msb_zero

$\texttt{msb\_zero:} \quad \forall s : BV_n.\ k < l \rightarrow s[(l-1)...k] = [0...0]$

Proof Sketch:

$$k = \sum_{i=0}^{l-1} s[i] \cdot 2^i$$

$$\begin{array}{cccccccc} & & l-1 & & k & k-1 & & 0 \\ s: & [& 0 & \ldots & 0 & \_ & \ldots & \_ \;] \end{array}$$

For $l = 4$,

s	k	l – k
0000	0	4
0001	1	3
0010	2	2
0011	3	1

Proof Sketch of msb_zero

$\texttt{msb\_zero:} \quad \forall s : BV_n.\ k < l \rightarrow s[(l-1)...k] = [0...0]$

Proof Sketch:

$$k = \sum_{i=0}^{l-1} s[i] \cdot 2^i$$

$$= s[l-1] \cdot 2^{l-1} + ... + s[1] \cdot 2^1 + s[0] \cdot 2^0$$

$$\begin{array}{cccccccccc} & & l-1 & & & k & k-1 & & & 0 \\ \text{s:} & [& 0 & \ldots & & 0 & \_ & \ldots & & \_\ ] \end{array}$$

For $l = 4$,

s	k	l – k
0000	0	4
0001	1	3
0010	2	2
0011	3	1

Proof Sketch of msb_zero

$\texttt{msb\_zero:}$ $\quad \forall s : BV_n.\ k < l \rightarrow s[(l-1)...k] = [0...0]$

Proof Sketch:

$$k = \sum_{i=0}^{l-1} s[i] \cdot 2^i$$

$$= s[l-1] \cdot 2^{l-1} + ... + s[1] \cdot 2^1 + s[0] \cdot 2^0$$

But $k < l$

$$\begin{array}{lccccccc} & & l-1 & & k & k-1 & & 0 \\ \text{s:} & [& 0 & \ldots & 0 & \_ & \ldots & \_ &] \end{array}$$

For $l = 4$,

s	k	l – k
0000	0	4
0001	1	3
0010	2	2
0011	3	1

Proof Sketch of msb_zero

`msb_zero:` $\forall s : BV_n .\ k < l \rightarrow s[(l-1)...k] = [0...0]$

$$\begin{array}{ccccccc} l-1 & & k & k-1 & & 0 \\ \text{s: } [\ 0 & ... & 0 & \_ & ... & \_\ ] \end{array}$$

Proof Sketch:

$$k = \sum_{i=0}^{l-1} s[i] \cdot 2^i$$

$$= s[l-1] \cdot 2^{l-1} + ... + s[1] \cdot 2^1 + s[0] \cdot 2^0$$

But $k < l$

$$= s[l-1] \cdot 2^{l-1} + \ ... \ + s[k] \cdot 2^k + s[k-1] \cdot 2^{k-1} + \ ... \ + s[1] \cdot 2^1 + s[0] \cdot 2^0$$

For $l = 4$,

s	k	l – k
0000	0	4
0001	1	3
0010	2	2
0011	3	1

Proof Sketch of msb_zero

`msb_zero:` $\forall s : BV_n.\ k < l \rightarrow s[(l-1)...k] = [0...0]$

Proof Sketch:

$$
\begin{array}{cccccccc}
l-1 & & & k & k-1 & & & 0 \\
\end{array}
$$

$$\text{s: } [\ 0\ \ldots\ 0\ \_\ \ldots\ \_\ ]$$

$$k = \sum_{i=0}^{l-1} s[i] \cdot 2^i$$

$$= s[l-1] \cdot 2^{l-1} + ... + s[1] \cdot 2^1 + s[0] \cdot 2^0$$

But $k < l$

$$= s[l-1] \cdot 2^{l-1} + \ \ ... \ \ + s[k] \cdot 2^k + s[k-1] \cdot 2^{k-1} + \ \ ... \ \ + s[1] \cdot 2^1 + s[0] \cdot 2^0$$

But $k < 2^k < 2^{k+1} < ... < 2^{l-1}$

For $l = 4$,

s	k	l – k
0000	0	4
0001	1	3
0010	2	2
0011	3	1

Proof Sketch of msb_zero

`msb_zero:` $\forall s : BV_n.\ k < l \rightarrow s[(l-1)...k] = [0...0]$

Proof Sketch:

$$\begin{array}{cccccccccc} & & l-1 & & k & k-1 & & 0 & \\ \mathrm{s:} & [& 0 & \ldots & 0 & \_ & \ldots & \_ &] \end{array}$$

$$k = \sum_{i=0}^{l-1} s[i] \cdot 2^i$$

$$= s[l-1] \cdot 2^{l-1} + ... + s[1] \cdot 2^1 + s[0] \cdot 2^0$$

But $k < l$

$$= s[l-1] \cdot 2^{l-1} + \ \ldots\ + s[k] \cdot 2^k + s[k-1] \cdot 2^{k-1} + \ \ldots\ + s[1] \cdot 2^1 + s[0] \cdot 2^0$$

But $k < 2^k < 2^{k+1} < ... < 2^{l-1}$

Thus, the coefficients of $2^k, ..., 2^{l-1}$ are 0.

For $l = 4$,

s	k	l – k
0000	0	4
0001	1	3
0010	2	2
0011	3	1

Conclusion

- [Niemetz et al., CAV 2018] presented 162 invertibility conditions
- [Niemetz et al., CADE 2019] verified ≈75% of the equivalences using an encoding
- We complemented [Niemetz et al., CADE 2019] in proving all but one invertibility equivalences from 66 of them
- We did this in the Coq proof assistant
- We extended the Coq bit-vector library for SMTCoq to do this

Future Work

- Integrate the extended bit-vector library into SMTCoq
- Prove the remaining invertibility equivalences
- Extend the library with $/$ $\%$ $\leq_s$ $\geq_s$
- Refactor the library for improved readability and modularity
- Consider using MathComp library for future proofs

References

- [Gupta et al., 1993] Aarti Gupta, Allan L. Fisher. Representation and Symbolic Manipulation of Linearly Inductive Boolean Functions. In proceedings of ICCAD '93 of the 1993 IEEE/ACM international conference on Computer-aided design, pages 192-199.
- [Armando et al., 2006] Alessandro Armando, Jacopo Mantovani, Lorenzo Platania. Bounded Model Checking of Software Using SMT Solvers Instead of SAT Solvers. In proceedings of International SPIN Workshop on Model Checking of Software. SPIN 2006: Model Checking Software, pages 146-162.
- [Cadar et al., 2006] Cristian Cadar, Vijay Ganesh, Peter M. Pawlowski, David L. Dill, Dawson R. Engler. EXE: Automatically Generating Inputs of Death. In proceedings of CCS '06 Proceedings of the 13th ACM conference on Computer and communications security, pages 322-335.
- [Niemetz et al., CAV 2018] Aina Niemetz, Mathias Preiner, Andrew Reynolds, Clark Barrett, Cesare Tinelli. Solving Quantified Bit-Vectors Using Invertibility Conditions. In proceedings of International Conference on Computer Aided Verification 2018, pages 236-255.
- [Niemetz et al., CADE 2019] Aina Niemetz, Mathias Preiner, Andrew Reynolds, Yoni Zohar, Clark Barrett and Cesare Tinelli. *Towards Bit Width Independent Proofs in SMT Solvers.* To appear in proceedings of International Conference on Automated Deduction 2019.
- [Ekici et al., 2017] Burak Ekici, Alain Mebsout, Cesare Tinelli, Chantal Keller, Guy Katz, Andrew Reynolds, Clark Barrett. SMTCoq: A Plug-in for Integrating SMT Solvers into Coq. In proceedings of 29th International Conference of Computer Aided Verification 2017.

Useful Lemmas

```
Lemma firstn_all: ∀ (l : list A), firstn (length l) l = l.

Lemma firstn_length_le: ∀ (l : list A) (n : nat), n <= length l
-> length (firstn n l) = n.

Lemma firstn_length: ∀ (n : nat) (l : list A), length (firstn n
l) = min n (length l).

Theorem app_nil_r: ∀ (l : list A), l ++ [] = l.

Lemma app_length: ∀ (l l' : list A), length (l++l') = length l +
length l'.
```