On the Complexity of Fixed-Size Bit-Vector Logics with Binary Encoded Bit-Width

Gergely Kovásznai, Andreas Fröhlich, Armin Biere

Institute for Formal Models and Verification
Johannes Kepler University, Linz, Austria
http://fmv.jku.at

SMT 2012
June 30 - July 1, 2012
Manchester, UK

JOHANNES KEPLER UNIVERSITY LINZ | JKU

Motivation

- How the *encoding* of the bit-widths affects the *complexity* of satisfiability checking for BV logics?
- In practice *logarithmic* (e.g. binary, decimal, hexadecimal) encoding is used (in contrast with unary encoding)

Example in SMT2

```
(set-logic QF_BV)
(declare-fun x () (_ BitVec 1000000))
(declare-fun y () (_ BitVec 1000000))
(assert (distinct (bvadd x y) (bvadd y x)))
```

Using Boolector:

- 103 MB in AIGER format; 1 GB in DIMACS format
- Bit-width of 10 million → cannot be bit-blasted (due to integer overflow)

Motivation

- How the *encoding* of the bit-widths affects the *complexity* of satisfiability checking for BV logics?
- In practice *logarithmic* (e.g. binary, decimal, hexadecimal) encoding is used (in contrast with unary encoding)

Example in SMT2

```
(set-logic QF_BV)
(declare-fun x () (_ BitVec 1000000))
(declare-fun y () (_ BitVec 1000000))
(assert (distinct (bvadd x y) (bvadd y x)))
```

Using Boolector:

- 103 MB in AIGER format; 1 GB in DIMACS format
- Bit-width of 10 million $\rightarrow$ cannot be bit-blasted (due to integer overflow)

Preliminaries

quantifiers			
no		*yes*	
uninterpreted functions		uninterpreted functions	
no	*yes*	*no*	*yes*
QF_BV	QF_UFBV	BV	UFBV

Preliminaries

		quantifiers			
		no		*yes*	
		uninterpreted functions		uninterpreted functions	
		no	*yes*	*no*	*yes*
encoding	*unary*	QF_BV1	QF_UFBV1	BV1	UFBV1
	binary	QF_BV2	QF_UFBV2	BV2	UFBV2

Assume common (SMT-LIB) operators → bit-blasting is *polynomial* in bit-width

Complexity: completeness results

		QF_BV	QF_UFBV	BV	UFBV
encoding	unary	?	?	?	?
	binary	?	?	?	?

- QF_BV1 is NP-complete ← bit-blasting to SAT
- QF_UFBV1 is NP-complete ← using Ackermann constraints
- BV1 is PSPACE-complete ← bit-blasting to QBF
- UFBV1 is NEXPTIME-complete ← proved in:

  C. M. Wintersteiger, *Termination Analysis for Bit-Vector Programs*. PhD Thesis, ETH Zürich, 2011.

  C. M. Wintersteiger, Y. Hamadi, L. Mendonça de Moura, *Efficiently Solving Quantified Bit-Vector Formulas*. Proc. FMCAD, 2010.

Complexity: completeness results

		QF_BV	QF_UFBV	BV	UFBV
encoding	*unary*	**NP**	?	?	?
	binary	?	?	?	?

- QF_BV1 is NP-complete ← bit-blasting to SAT
- QF_UFBV1 is NP-complete ← using Ackermann constraints
- BV1 is PSPACE-complete ← bit-blasting to QBF
- UFBV1 is NEXPTIME-complete ← proved in:

  C. M. Wintersteiger, *Termination Analysis for Bit-Vector Programs.* PhD Thesis, ETH Zürich, 2011.

  C. M. Wintersteiger, Y. Hamadi, L. Mendonça de Moura, *Efficiently Solving Quantified Bit-Vector Formulas.* Proc. FMCAD, 2010.

Complexity: completeness results

		QF_BV	QF_UFBV	BV	UFBV
encoding	*unary*	NP	**NP**	?	?
	binary	?	?	?	?

- QF_BV1 is NP-complete ← bit-blasting to SAT
- QF_UFBV1 is NP-complete ← using Ackermann constraints
- BV1 is PSPACE-complete ← bit-blasting to QBF
- UFBV1 is NEXPTIME-complete ← proved in:

  C. M. Wintersteiger, *Termination Analysis for Bit-Vector Programs.* PhD Thesis, ETH Zürich, 2011.

  C. M. Wintersteiger, Y. Hamadi, L. Mendonça de Moura, *Efficiently Solving Quantified Bit-Vector Formulas.* Proc. FMCAD, 2010.

Complexity: completeness results

		QF_BV	QF_UFBV	BV	UFBV
encoding	*unary*	NP	NP	**PSPACE**	?
	binary	?	?	?	?

- QF_BV1 is NP-complete ← bit-blasting to SAT
- QF_UFBV1 is NP-complete ← using Ackermann constraints
- BV1 is PSPACE-complete ← bit-blasting to QBF
- UFBV1 is NEXPTIME-complete ← proved in:

  C. M. Wintersteiger, *Termination Analysis for Bit-Vector Programs.* PhD Thesis, ETH Zürich, 2011.

  C. M. Wintersteiger, Y. Hamadi, L. Mendonça de Moura, *Efficiently Solving Quantified Bit-Vector Formulas.* Proc. FMCAD, 2010.

Complexity: completeness results

		QF_BV	QF_UFBV	BV	UFBV
encoding	*unary*	NP	NP	PSPACE	**NEXPTIME**
	binary	?	?	?	?

- QF_BV1 is NP-complete ← bit-blasting to SAT
- QF_UFBV1 is NP-complete ← using Ackermann constraints
- BV1 is PSPACE-complete ← bit-blasting to QBF
- UFBV1 is NEXPTIME-complete ← proved in:

  C. M. Wintersteiger, *Termination Analysis for Bit-Vector Programs*. PhD Thesis, ETH Zürich, 2011.
  C. M. Wintersteiger, Y. Hamadi, L. Mendonça de Moura, *Efficiently Solving Quantified Bit-Vector Formulas*. Proc. FMCAD, 2010.

Complexity: completeness results

		QF_BV	QF_UFBV	BV	UFBV
encoding	*unary*	NP	NP	PSPACE	NEXPTIME
	binary	**NEXPTIME**	?	?	?

QF_BV2 is NEXPTIME-complete:

- QF_BV2 $\in$ NEXPTIME:

$$\text{QF\_BV2} \xrightarrow{\text{exponentially}} \text{QF\_BV1} \in \text{NP}$$

- QF_BV2 is NEXPTIME-hard:

$$\text{DQBF} \xrightarrow{\text{polynomially}} \text{QF\_BV2}$$

Complexity: completeness results

		QF_BV	QF_UFBV	BV	UFBV
encoding	*unary*	NP	NP	PSPACE	NEXPTIME
	binary	**NEXPTIME**	?	?	?

QF_BV2 is NExpTime-complete:

- QF_BV2 $\in$ NExpTime:

$$\text{QF\_BV2} \xrightarrow{\text{exponentially}} \text{QF\_BV1} \in \text{NP}$$

- QF_BV2 is NExpTime-hard:

$$\text{DQBF} \xrightarrow{\text{polynomially}} \text{QF\_BV2}$$

Complexity: $\mathrm{DQBF} \xrightarrow{\text{polynomially}} \mathrm{QF\_BV2}$

Dependency Quantified Boolean Formulas (DQBF):

- Applying Henkin quantifiers: variable dependencies represent a partial order
- Dependencies are *explicitly* specified

Example DQBF

$$\begin{aligned} \forall u_0, u_1, u_2 \exists x(u_0), y(u_1, u_2) \ . \ & (x \vee y \vee \neg u_0 \vee \neg u_1) \wedge \\ & (x \vee \neg y \vee u_0 \vee \neg u_1 \vee \neg u_2) \wedge \\ & (x \vee \neg y \vee \neg u_0 \vee \neg u_1 \vee u_2) \wedge \\ & (\neg x \vee y \vee \neg u_0 \vee \neg u_2) \wedge \\ & (\neg x \vee \neg y \vee u_0 \vee u_1 \vee \neg u_2) \end{aligned}$$

- DQBF is $\mathrm{N}\textsc{ExpTime}$-complete

  G. L. Peterson, J. H. Reif, *Multiple-Person Alternation*. Foundations of Computer Science, 1979.

Complexity: $\mathrm{DQBF} \xrightarrow{\text{polynomially}} \mathrm{QF\_BV2}$

$$\Big((X^{[8]} \mid Y^{[8]} \mid\!\sim U_0^{[8]} \mid\!\sim U_1^{[8]}) \;\&\; (X \mid\!\sim Y \mid U_0 \mid\!\sim U_1 \mid\!\sim U_2^{[8]}) \;\&$$
$$(X \mid\!\sim Y \mid\!\sim U_0 \mid\!\sim U_1 \mid U_2) \;\&\; (\sim X \mid Y \mid\!\sim U_0 \mid\!\sim U_2) \;\&$$
$$(\sim X \mid\!\sim Y \mid U_0 \mid U_1 \mid\!\sim U_2)\Big) \;=\; \sim 0^{[8]}$$

1. Eliminate the quantifier prefix
2. Replace logical connectives with bit-wise operators
3. Replace Boolean variables with bit-vector variables of ***bit-width*** 2^k

    k: number of universal variables in the DQBF

$$\Big((X^{[8]} \mid Y^{[8]} \mid\!\sim U_0^{[8]} \mid\!\sim U_1^{[8]}) \;\&\; (X \mid\!\sim Y \mid U_0 \mid\!\sim U_1 \mid\!\sim U_2^{[8]}) \;\&$$
$$(X \mid\!\sim Y \mid\!\sim U_0 \mid\!\sim U_1 \mid U_2) \;\&\; (\sim X \mid Y \mid\!\sim U_0 \mid\!\sim U_2) \;\&$$
$$(\sim X \mid\!\sim Y \mid U_0 \mid U_1 \mid\!\sim U_2)\Big) \;=\; \sim 0^{[8]}$$

4. *Universal* vars $\leftarrow$ Assign *binary magic numbers* to U_is!

$$U_0 := \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}, U_1 := \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}, U_2 := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Complexity: $\mathrm{DQBF} \xrightarrow{\text{polynomially}} \mathrm{QF\_BV2}$

$$\Big((X^{[8]} \mid Y^{[8]} \mid\sim U_0^{[8]} \mid\sim U_1^{[8]}) \;\&\; (X \mid\sim Y \mid U_0 \mid\sim U_1 \mid\sim U_2^{[8]}) \;\&$$
$$(X \mid\sim Y \mid\sim U_0 \mid\sim U_1 \mid U_2) \;\&\; (\sim X \mid Y \mid\sim U_0 \mid\sim U_2) \;\&$$
$$(\sim X \mid\sim Y \mid U_0 \mid U_1 \mid\sim U_2)\Big) \;=\; \sim 0^{[8]}$$

4. *Universal* vars ← Assign *binary magic numbers* to U_is!

$$U_0 := \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix},\; U_1 := \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix},\; U_2 := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Complexity: $\mathrm{DQBF} \xrightarrow{\text{polynomially}} \mathrm{QF\_BV2}$

$$\Big((X^{[8]} \mid Y^{[8]} \mid\sim U_0^{[8]} \mid\sim U_1^{[8]}) \;\&\; (X \mid\sim Y \mid U_0 \mid\sim U_1 \mid\sim U_2^{[8]}) \;\&$$
$$(X \mid\sim Y \mid\sim U_0 \mid\sim U_1 \mid U_2) \;\&\; (\sim X \mid Y \mid\sim U_0 \mid\sim U_2) \;\&$$
$$(\sim X \mid\sim Y \mid U_0 \mid U_1 \mid\sim U_2) \Big) \;=\; \sim 0^{[8]}$$

4. *Universal* vars $\leftarrow$ Assign *binary magic numbers* to U_is!

$$U_i := \frac{2^{(2^k)} - 1}{2^{(2^i)} + 1}$$

Complexity: $\mathrm{DQBF} \xrightarrow{\text{polynomially}} \mathrm{QF\_BV2}$

$$\Big((X^{[8]} \mid Y^{[8]} \mid\sim U_0^{[8]} \mid\sim U_1^{[8]}) \;\&\; (X \mid\sim Y \mid U_0 \mid\sim U_1 \mid\sim U_2^{[8]}) \;\&$$
$$(X \mid\sim Y \mid\sim U_0 \mid\sim U_1 \mid U_2) \;\&\; (\sim X \mid Y \mid\sim U_0 \mid\sim U_2) \;\&$$
$$(\sim X \mid\sim Y \mid U_0 \mid U_1 \mid\sim U_2)\Big) \;=\; \sim 0^{[8]}$$

4. *Universal* vars $\leftarrow$ Assign *binary magic numbers* to U_is!

$$U_i := \frac{2^{(2^k)} - 1}{2^{(2^i)} + 1}$$

$$\Big((U_i \ll (1 \ll i)) + U_i\Big) = \sim 0^{[2^k]}$$

Complexity: $\mathrm{DQBF} \xrightarrow{\text{polynomially}} \mathrm{QF\_BV2}$

$$\Big((X^{[8]} \mid Y^{[8]} \mid\sim U_0^{[8]} \mid\sim U_1^{[8]}) \;\&\; (X \mid\sim Y \mid U_0 \mid\sim U_1 \mid\sim U_2^{[8]}) \;\&$$

$$(X \mid\sim Y \mid\sim U_0 \mid\sim U_1 \mid U_2) \;\&\; (\sim X \mid Y \mid\sim U_0 \mid\sim U_2) \;\&$$

$$(\sim X \mid\sim Y \mid U_0 \mid U_1 \mid\sim U_2)\Big) \;=\; \sim 0^{[8]}$$

$$\wedge \; \boxed{\bigwedge_{i \in \{0,1,2\}} \Big(\big((U_i \ll (1 \ll i)) + U_i\big) \;=\; \sim 0^{[8]}\Big)}$$

4. *Universal* vars ← Assign *binary magic numbers* to U_is!

$$\Big((U_i \ll (1 \ll i)) + U_i\Big) = \sim 0^{[2^k]}$$

Complexity: DQBF $\xrightarrow{\text{polynomially}}$ QF_BV2

$$
\Big((X^{[8]} \mid Y^{[8]} \mid\!\sim U_0^{[8]} \mid\!\sim U_1^{[8]}) \;\&\; (X \mid\!\sim Y \mid U_0 \mid\!\sim U_1 \mid\!\sim U_2^{[8]}) \;\&
$$
$$
(X \mid\!\sim Y \mid\!\sim U_0 \mid\!\sim U_1 \mid U_2) \;\&\; (\sim X \mid Y \mid\!\sim U_0 \mid\!\sim U_2) \;\&
$$
$$
(\sim X \mid\!\sim Y \mid U_0 \mid U_1 \mid\!\sim U_2)\Big) \;=\; \sim 0^{[8]}
$$
$$
\wedge \bigwedge_{i \in \{0,1,2\}} \Big(((U_i \ll (1 \ll i)) + U_i) \;=\; \sim 0^{[8]}\Big)
$$

5 *Existential* vars $\leftarrow$ Represent *Skolem-functions* as bit-vectors!

$$
X = \begin{bmatrix} x_7 \\ x_6 \\ x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \\ x_0 \end{bmatrix} \text{ is independent of } U_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}
$$

Complexity: DQBF $\xrightarrow{\text{polynomially}}$ QF_BV2

$$\Big((X^{[8]} \mid Y^{[8]} \mid\!\sim U_0^{[8]} \mid\!\sim U_1^{[8]}) \;\&\; (X \mid\!\sim Y \mid U_0 \mid\!\sim U_1 \mid\!\sim U_2^{[8]}) \;\&$$

Positive and negative cofactors of a Skolem-function w.r.t. U_i

$$\wedge \bigwedge_{i\in\{0,1,2\}} \Big(((U_i \ll (1 \ll i \quad U_i) \;=\; \sim 0^{[8]}\Big)$$

5. *Existential* vars ← Represent *Skolem-functions* as bit-vectors!

$$X = \begin{bmatrix} x_7 \\ x_6 \\ x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \\ x_0 \end{bmatrix} \text{ is independent of } U_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Complexity: $\mathrm{DQBF} \xrightarrow{\text{polynomially}} \mathrm{QF\_BV2}$

$$\Big((X^{[8]} \mid Y^{[8]} \mid\!\sim U_0^{[8]} \mid\!\sim U_1^{[8]}) \;\&\; (X \mid\!\sim Y \mid U_0 \mid\!\sim U_1 \mid\!\sim U_2^{[8]}) \;\&$$

Positive and negative cofactors of a Skolem-function w.r.t. U_i

$$\wedge \bigwedge_{i\in\{0,1,2\}} \Big(((U_i \ll (1 \ll i \quad U_i) \;= \sim 0^{[8]}\Big)$$

5 *Existential* vars $\leftarrow$ Represent *Skolem-functions* as bit-vectors!

X is independent of U_i:

$$(X \;\&\; U_i) = \Big((X \gg (1 \ll i)) \;\&\; U_i\Big)$$

Complexity: $\mathrm{DQBF} \xrightarrow{\text{polynomially}} \mathrm{QF\_BV2}$

$$\Big((X^{[8]} \mid Y^{[8]} \mid\!\sim U_0^{[8]} \mid\!\sim U_1^{[8]}) \;\&\; (X \mid\!\sim Y \mid U_0 \mid\!\sim U_1 \mid\!\sim U_2^{[8]}) \;\&$$
$$(X \mid\!\sim Y \mid\!\sim U_0 \mid\!\sim U_1 \mid U_2) \;\&\; (\sim X \mid Y \mid\!\sim U_0 \mid\!\sim U_2) \;\&$$
$$(\sim X \mid\!\sim Y \mid U_0 \mid U_1 \mid\!\sim U_2)\Big) \;=\; \sim 0^{[8]}$$
$$\wedge \bigwedge_{i \in \{0,1,2\}} \Big(((U_i \ll (1 \ll i)) + U_i) \;=\; \sim 0^{[8]}\Big)$$
$$\wedge \boxed{(X \;\&\; U_1) \;=\; \big((X \gg (1 \ll 1)) \;\&\; U_1\big)}$$

5 *Existential* vars $\leftarrow$ Represent *Skolem-functions* as bit-vectors!

X is independent of U_i:

$$(X \;\&\; U_i) = \Big((X \gg (1 \ll i)) \;\&\; U_i\Big)$$

Complexity: $\mathrm{DQBF} \xrightarrow{\text{polynomially}} \mathrm{QF\_BV2}$

$$\Big((X^{[8]} \mid Y^{[8]} \mid\!\sim U_0^{[8]} \mid\!\sim U_1^{[8]}) \;\&\; (X \mid\!\sim Y \mid U_0 \mid\!\sim U_1 \mid\!\sim U_2^{[8]}) \;\&$$

$$(X \mid\!\sim Y \mid\!\sim U_0 \mid\!\sim U_1 \mid U_2) \;\&\; (\sim X \mid Y \mid\!\sim U_0 \mid\!\sim U_2) \;\&$$

$$(\sim X \mid\!\sim Y \mid U_0 \mid U_1 \mid\!\sim U_2)\Big) \;=\; \sim 0^{[8]}$$

$$\wedge \bigwedge_{i \in \{0,1,2\}} \Big(((U_i \ll (1 \ll i)) + U_i) \;=\; \sim 0^{[8]}\Big)$$

$$\wedge \;\boxed{(X \;\&\; U_1) \;=\; ((X \gg (1 \ll 1)) \;\&\; U_1)}$$

$$\wedge \;\boxed{(X \;\&\; U_2) \;=\; ((X \gg (1 \ll 2)) \;\&\; U_2)}$$

$$\wedge \;\boxed{(Y \;\&\; U_0) \;=\; ((Y \gg (1 \ll 0)) \;\&\; U_0)}$$

Complexity: $\text{DQBF} \xrightarrow{\text{polynomially}} \text{QF\_BV2}$

$$
\Big((X^{[8]} \mid Y^{[8]} \mid\sim U_0^{[8]} \mid\sim U_1^{[8]}) \,\&\, (X \mid\sim Y \mid U_0 \mid\sim U_1 \mid\sim U_2^{[8]}) \,\&\,
(X \mid\sim Y \mid\sim U_0 \mid\sim U_1 \mid U_2) \,\&\, (\sim X \mid Y \mid\sim U_0 \mid\sim U_2) \,\&\,
(\sim X \mid\sim Y \mid U_0 \mid U_1 \mid\sim U_2)\Big) \;=\; \sim 0^{[8]}
$$

$$
\wedge \bigwedge_{i \in \{0,1,2\}} \Big(((U_i \ll (1 \ll i)) + U_i) \;=\; \sim 0^{[8]}\Big)
$$

$$
\wedge \;\boxed{(X \;\&\; U_1) \;=\; ((X \gg (1 \ll 1)) \;\&\; U_1)}
$$

$$
\wedge \;\boxed{(X \;\&\; U_2) \;=\; ((X \gg (1 \ll 2)) \;\&\; U_2)}
$$

$$
\wedge \;\boxed{(Y \;\&\; U_0) \;=\; ((Y \gg (1 \ll 0)) \;\&\; U_0)}
$$

Bit-width 2^k is encoded logarithmically, due to *binary encoding*!

Complexity: completeness results

		QF_BV	QF_UFBV	BV	UFBV
encoding	*unary*	NP	NP	PSPACE	NEXPTIME
	binary	NEXPTIME	?	?	?

- QF_UFBV2 is NEXPTIME-complete ← using Ackermann constraints
- BV2 $\in$ EXPSPACE and is NEXPTIME-hard
- UFBV2 is 2-NEXPTIME-complete:

$$2^{(2^n)}\text{-square domino tiling problem} \xrightarrow{\text{polynomially}} \text{UFBV2}$$

Complexity: completeness results

		QF_BV	QF_UFBV	BV	UFBV
encoding	*unary*	NP	NP	PSPACE	NEXPTIME
	binary	NEXPTIME	**NEXPTIME**	?	?

- QF_UFBV2 is NEXPTIME-complete $\leftarrow$ using Ackermann constraints
- BV2 $\in$ EXPSPACE and is NEXPTIME-hard
- UFBV2 is 2-NEXPTIME-complete:

$$2^{(2^n)}\text{-square domino tiling problem} \xrightarrow{\text{polynomially}} \text{UFBV2}$$

Complexity: completeness results

		QF_BV	QF_UFBV	BV	UFBV
encoding	unary	NP	NP	PSPACE	NEXPTIME
	binary	NEXPTIME	NEXPTIME	?	?

- QF_UFBV2 is NEXPTIME-complete $\leftarrow$ using Ackermann constraints
- BV2 $\in$ EXPSPACE and is NEXPTIME-hard
- UFBV2 is 2-NEXPTIME-complete:

$$2^{(2^n)}\text{-square domino tiling problem} \xrightarrow{\text{polynomially}} \text{UFBV2}$$

Complexity: completeness results

		QF_BV	QF_UFBV	BV	UFBV
encoding	*unary*	NP	NP	PSPACE	NEXPTIME
	binary	NEXPTIME	NEXPTIME	?	**2-NEXPTIME**

- QF_UFBV2 is NEXPTIME-complete ← using Ackermann constraints
- BV2 ∈ EXPSPACE and is NEXPTIME-hard
- UFBV2 is 2-NEXPTIME-complete:

$$2^{(2^n)}\text{-square domino tiling problem} \xrightarrow{\text{polynomially}} \text{UFBV2}$$

Bit-width bounded problems

		QF_BV	QF_UFBV	BV	UFBV
encoding	*unary*	**NP**	**NP**	**PSPACE**	**NEXPTIME**
	binary	NExpTime	NExpTime	?	2-NExpTime

- Bit-width boundedness: a practical property for BV problems to avoid exponential blow-up
- If a BV problem S is *bit-width bounded*:
  - $S \subset \mathrm{QF\_BV}(2) \quad \Rightarrow \quad \boxed{S \in \mathrm{NP}}$
  - $S \subset \mathrm{QF\_UFBV}(2) \quad \Rightarrow \quad \boxed{S \in \mathrm{NP}}$
  - $S \subset \mathrm{BV}(2) \quad \Rightarrow \quad \boxed{S \in \mathrm{PSPACE}}$
  - $S \subset \mathrm{UFBV}(2) \quad \Rightarrow \quad \boxed{S \in \mathrm{NEXPTIME}}$

Conclusion

		QF_BV	QF_UFBV	BV	UFBV
encoding	*unary*	NP	NP	PSPACE	NEXPTIME
	binary	**NEXPTIME**	**NEXPTIME**	?	2-NEXPTIME

The complexity of deciding the commonly used BV logics *was an open question*.

Future work:

- Is BV2 complete?
- Consider these results in the context of parametrized complexity
- Instead of bit-blasting+SAT, other approaches with EPR/DQBF?

Conclusion

		QF_BV	QF_UFBV	BV	UFBV
encoding	*unary*	NP	NP	PSPACE	NEXPTIME
	binary	**NEXPTIME**	**NEXPTIME**	?	2-NEXPTIME

The complexity of deciding the commonly used BV logics *was an open question*.

Future work:

- Is BV2 complete?
- Consider these results in the context of parametrized complexity
- Instead of bit-blasting+SAT, other approaches with EPR/DQBF?