Reducing Bit-Vector Polynomials to SAT using Gröbner bases

Tom Seed, Andy King and Neil Evans

27 April 2020

An alliance between two kingdoms

- Symbolic computation aka computer algebra

An alliance between two kingdoms

- Symbolic computation aka computer algebra
- SMT solving

An alliance between two kingdoms

- Symbolic computation aka computer algebra
- SMT solving
- Agenda/challenge papers:

An alliance between two kingdoms

- Symbolic computation aka computer algebra
- SMT solving
- Agenda/challenge papers:
  1. Ábráham: Building Bridges between Symbolic Computation and Satisfiability Checking, ISSAC, ACM Press, 2015

An alliance between two kingdoms

- Symbolic computation aka computer algebra
- SMT solving
- Agenda/challenge papers:
  1. Ábráham: Building Bridges between Symbolic Computation and Satisfiability Checking, ISSAC, ACM Press, 2015
  2. Davenport, England, Griggio, Sturm, and Tinelli: Symbolic Computation and Satisfiability Checking, JSC, 2020

An alliance between two kingdoms

- Symbolic computation aka computer algebra
- SMT solving
- Agenda/challenge papers:
  1. Ábráham: Building Bridges between Symbolic Computation and Satisfiability Checking, ISSAC, ACM Press, 2015
  2. Davenport, England, Griggio, Sturm, and Tinelli: Symbolic Computation and Satisfiability Checking, JSC, 2020
- Headline results:

An alliance between two kingdoms

- Symbolic computation aka computer algebra
- SMT solving
- Agenda/challenge papers:
  1. Ábráham: Building Bridges between Symbolic Computation and Satisfiability Checking, ISSAC, ACM Press, 2015
  2. Davenport, England, Griggio, Sturm, and Tinelli: Symbolic Computation and Satisfiability Checking, JSC, 2020
- Headline results:
  1. Jovanović and de Moura: Solving Non-linear Arithmetic, International Joint Conference on Automated Reasoning, LNCS 7364, 2012

An alliance between two kingdoms

- Symbolic computation aka computer algebra
- SMT solving
- Agenda/challenge papers:
  1. Ábráham: Building Bridges between Symbolic Computation and Satisfiability Checking, ISSAC, ACM Press, 2015
  2. Davenport, England, Griggio, Sturm, and Tinelli: Symbolic Computation and Satisfiability Checking, JSC, 2020
- Headline results:
  1. Jovanović and de Moura: Solving Non-linear Arithmetic, International Joint Conference on Automated Reasoning, LNCS 7364, 2012
  2. Horácek, Burchard, Becker and Kreuzer: Integrating Algebraic and SAT Solvers, Mathematical Aspects of Computer and Information Sciences, LNCS 10693, 2017

Example

- Consider the system:

$$B = \begin{cases} y^2 + 120x^2 + 123x + 48 \equiv_{256} 0 \\ yx + 65x^2 + 50x + 32 \equiv_{256} 0 \\ 2y + 63x^2 + 59x + 128 \equiv_{256} 0 \\ x^3 + 135x^2 + 100x + 64 \equiv_{256} 0 \\ 64x^2 + 192x \equiv_{256} 0 \end{cases}$$

where $x, y \in \mathbb{Z}_{256}$.

Example

- Consider the system:

$$B = \begin{cases} y^2 + 120x^2 + 123x + 48 \equiv_{256} 0 \\ yx + 65x^2 + 50x + 32 \equiv_{256} 0 \\ 2y + 63x^2 + 59x + 128 \equiv_{256} 0 \\ x^3 + 135x^2 + 100x + 64 \equiv_{256} 0 \\ 64x^2 + 192x \equiv_{256} 0 \end{cases}$$

  where $x, y \in \mathbb{Z}_{256}$.

- Is this system satisfiable?

Example

- Consider the system:

$$B = \begin{cases} y^2 + 120x^2 + 123x + 48 \equiv_{256} 0 \\ yx + 65x^2 + 50x + 32 \equiv_{256} 0 \\ 2y + 63x^2 + 59x + 128 \equiv_{256} 0 \\ x^3 + 135x^2 + 100x + 64 \equiv_{256} 0 \\ 64x^2 + 192x \equiv_{256} 0 \end{cases}$$

  where $x, y \in \mathbb{Z}_{256}$.

- Is this system satisfiable?
- If so, how can we find a solution?

SAT versus algebraic approaches

- SAT approach:

SAT versus algebraic approaches

- SAT approach:
  - Compile each product/sum/equation into a propositional formula

SAT versus algebraic approaches

- SAT approach:
  - Compile each product/sum/equation into a propositional formula
  - Solve the resulting system with a SAT solver

SAT versus algebraic approaches

- SAT approach:
  - Compile each product/sum/equation into a propositional formula
  - Solve the resulting system with a SAT solver
  - But it becomes too large (becomes infeasible)

SAT versus algebraic approaches

- SAT approach:
  - Compile each product/sum/equation into a propositional formula
  - Solve the resulting system with a SAT solver
  - But it becomes too large (becomes infeasible)
  - And it becomes too slow (loss of word-level propagation)

SAT versus algebraic approaches

- SAT approach:
  - Compile each product/sum/equation into a propositional formula
  - Solve the resulting system with a SAT solver
  - But it becomes too large (becomes infeasible)
  - And it becomes too slow (loss of word-level propagation)
- Algebraic approach:

SAT versus algebraic approaches

- SAT approach:
  - Compile each product/sum/equation into a propositional formula
  - Solve the resulting system with a SAT solver
  - But it becomes too large (becomes infeasible)
  - And it becomes too slow (loss of word-level propagation)
- Algebraic approach:
  - For polynomials over algebraically closed fields, unsatisfiability can be decided by Hilbert's Nullstellensatz

SAT versus algebraic approaches

- SAT approach:
  - Compile each product/sum/equation into a propositional formula
  - Solve the resulting system with a SAT solver
  - But it becomes too large (becomes infeasible)
  - And it becomes too slow (loss of word-level propagation)
- Algebraic approach:
  - For polynomials over algebraically closed fields, unsatisfiability can be decided by Hilbert's Nullstellensatz
  - This equates unsatisfiability with the existence of a non-zero constant polynomial in a Gröbner basis for the polynomials

SAT versus algebraic approaches

- SAT approach:
  - Compile each product/sum/equation into a propositional formula
  - Solve the resulting system with a SAT solver
  - But it becomes too large (becomes infeasible)
  - And it becomes too slow (loss of word-level propagation)
- Algebraic approach:
  - For polynomials over algebraically closed fields, unsatisfiability can be decided by Hilbert's Nullstellensatz
  - This equates unsatisfiability with the existence of a non-zero constant polynomial in a Gröbner basis for the polynomials
  - Insufficient for bit-vector (machine integers) polynomials:

SAT versus algebraic approaches

- SAT approach:
  - Compile each product/sum/equation into a propositional formula
  - Solve the resulting system with a SAT solver
  - But it becomes too large (becomes infeasible)
  - And it becomes too slow (loss of word-level propagation)
- Algebraic approach:
  - For polynomials over algebraically closed fields, unsatisfiability can be decided by Hilbert's Nullstellensatz
  - This equates unsatisfiability with the existence of a non-zero constant polynomial in a Gröbner basis for the polynomials
  - Insufficient for bit-vector (machine integers) polynomials:
    - ★ Counter-example $x^2 + 2 \equiv_8 0$

SAT versus algebraic approaches

- SAT approach:
  - Compile each product/sum/equation into a propositional formula
  - Solve the resulting system with a SAT solver
  - But it becomes too large (becomes infeasible)
  - And it becomes too slow (loss of word-level propagation)
- Algebraic approach:
  - For polynomials over algebraically closed fields, unsatisfiability can be decided by Hilbert's Nullstellensatz
  - This equates unsatisfiability with the existence of a non-zero constant polynomial in a Gröbner basis for the polynomials
  - Insufficient for bit-vector (machine integers) polynomials:
    - ★ Counter-example $x^2 + 2 \equiv_8 0$
    - ★ Observe that any solution x must be even

SAT versus algebraic approaches

- SAT approach:
  - Compile each product/sum/equation into a propositional formula
  - Solve the resulting system with a SAT solver
  - But it becomes too large (becomes infeasible)
  - And it becomes too slow (loss of word-level propagation)
- Algebraic approach:
  - For polynomials over algebraically closed fields, unsatisfiability can be decided by Hilbert's Nullstellensatz
  - This equates unsatisfiability with the existence of a non-zero constant polynomial in a Gröbner basis for the polynomials
  - Insufficient for bit-vector (machine integers) polynomials:
    - ★ Counter-example $x^2 + 2 \equiv_8 0$
    - ★ Observe that any solution x must be even
    - ★ But $0^2 + 2 = 2$, $2^2 + 2 = 6$, $4^2 + 2 = 18 \equiv_8 2$ and $6^2 + 2 = 38 \equiv_8 6$

SAT versus algebraic approaches

- SAT approach:
  - Compile each product/sum/equation into a propositional formula
  - Solve the resulting system with a SAT solver
  - But it becomes too large (becomes infeasible)
  - And it becomes too slow (loss of word-level propagation)
- Algebraic approach:
  - For polynomials over algebraically closed fields, unsatisfiability can be decided by Hilbert's Nullstellensatz
  - This equates unsatisfiability with the existence of a non-zero constant polynomial in a Gröbner basis for the polynomials
  - Insufficient for bit-vector (machine integers) polynomials:
    - ★ Counter-example $x^2 + 2 \equiv_8 0$
    - ★ Observe that any solution x must be even
    - ★ But $0^2 + 2 = 2$, $2^2 + 2 = 6$, $4^2 + 2 = 18 \equiv_8 2$ and $6^2 + 2 = 38 \equiv_8 6$
    - ★ But Gröbner basis $\{x^2 + 2\}$ does not contain a constant polynomial

Gröbner Bases: An overview

- Normal form for systems of polynomials

Gröbner Bases: An overview

- Normal form for systems of polynomials
- Consider

$$\begin{cases} xy - 1 \equiv_{256} 0 \\ x^2 - x \equiv_{256} 0 \end{cases}$$

Gröbner Bases: An overview

- Normal form for systems of polynomials
- Consider

$$\begin{cases} xy - 1 \equiv_{256} 0 \\ x^2 - x \equiv_{256} 0 \end{cases}$$

- $(x-1)(xy-1) - y(x^2-x) = -x+1$

Gröbner Bases: An overview

- Normal form for systems of polynomials
- Consider

$$\begin{cases} xy - 1 \equiv_{256} 0 \\ x^2 - x \equiv_{256} 0 \end{cases}$$

- $(x-1)(xy-1) - y(x^2-x) = -x+1$
- Hence, $xy - 1 \equiv_{256} 0$ and $x^2 - x \equiv_{256} 0$ implies $-x+1 \equiv_{256} 0$, i.e., $x \equiv_{256} 1$

Gröbner Bases: An overview

- Normal form for systems of polynomials
- Consider

$$\begin{cases} xy - 1 \equiv_{256} 0 \\ x^2 - x \equiv_{256} 0 \end{cases}$$

- $(x-1)(xy-1) - y(x^2-x) = -x+1$
- Hence, $xy - 1 \equiv_{256} 0$ and $x^2 - x \equiv_{256} 0$ implies $-x+1 \equiv_{256} 0$, i.e., $x \equiv_{256} 1$
- More generally, if $p_1 \equiv_{256} 0, \ldots, p_m \equiv_{256} 0$ then for any polynomials $q_1, \ldots, q_m$, $q_1 p_1 + \cdots + q_m p_m \equiv_{256} 0$ is entailed

Gröbner Bases: An overview

- Normal form for systems of polynomials
- Consider

$$\begin{cases} xy - 1 \equiv_{256} 0 \\ x^2 - x \equiv_{256} 0 \end{cases}$$

- $(x-1)(xy-1) - y(x^2-x) = -x+1$
- Hence, $xy - 1 \equiv_{256} 0$ and $x^2 - x \equiv_{256} 0$ implies $-x+1 \equiv_{256} 0$, i.e., $x \equiv_{256} 1$
- More generally, if $p_1 \equiv_{256} 0, \ldots, p_m \equiv_{256} 0$ then for any polynomials $q_1, \ldots, q_m$, $q_1 p_1 + \cdots + q_m p_m \equiv_{256} 0$ is entailed
- $\{p_1, \ldots, p_m\}$ is a Gröbner basis iff all polynomials $p = q_1 p_1 + \cdots + q_m p_m$ are *reducible* by $\{p_1, \ldots, p_m\}$

Gröbner Bases: An overview

- Normal form for systems of polynomials
- Consider

$$\begin{cases} xy - 1 \equiv_{256} 0 \\ x^2 - x \equiv_{256} 0 \end{cases}$$

- $(x-1)(xy-1) - y(x^2-x) = -x+1$
- Hence, $xy - 1 \equiv_{256} 0$ and $x^2 - x \equiv_{256} 0$ implies $-x+1 \equiv_{256} 0$, i.e., $x \equiv_{256} 1$
- More generally, if $p_1 \equiv_{256} 0, \ldots, p_m \equiv_{256} 0$ then for any polynomials $q_1, \ldots, q_m$, $q_1 p_1 + \cdots + q_m p_m \equiv_{256} 0$ is entailed
- $\{p_1, \ldots, p_m\}$ is a Gröbner basis iff all polynomials $p = q_1 p_1 + \cdots + q_m p_m$ are *reducible* by $\{p_1, \ldots, p_m\}$

$$\begin{cases} xy - 1 \equiv_{256} 0 \\ x^2 - x \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y - 1 \equiv_{256} 0 \\ x - 1 \equiv_{256} 0 \end{cases}$$

SAT/algebraic love child

- Linear polynomials can impose/expose bit-level information

SAT/algebraic love child

- Linear polynomials can impose/expose bit-level information
- Example $2^6(x-1) \equiv_{256} 0$ iff last 2 bits of x are 01:

x	$64(x-1)$	$64(x-1) \pmod{256}$
0	-64	196
1	0	0
2	64	64
3	128	128
4	196	196
5	0	0
⋮	⋮	⋮

SAT/algebraic love child

- Linear polynomials can impose/expose bit-level information
- Example $2^6(x-1) \equiv_{256} 0$ iff last 2 bits of x are 01:

x	$64(x-1)$	$64(x-1) \pmod{256}$
0	-64	196
1	0	0
2	64	64
3	128	128
4	196	196
5	0	0
⋮	⋮	⋮

- General idea:

  k least significant bits of x and ℓ agree $\iff 2^{8-k}(x-\ell) \equiv_{256} 0$
  $\iff x - \ell \equiv_{256} 2^k w$

New technique: bit-sequence propagation

- Given $2^{8-k}(x - \ell) \equiv_{256} 0$ (or equivalently $x - \ell \equiv_{256} 2^k w$):

New technique: bit-sequence propagation

- Given $2^{8-k}(x - \ell) \equiv_{256} 0$ (or equivalently $x - \ell \equiv_{256} 2^k w$):
  1. Let

$$\ell' = \begin{cases} \ell & \text{to clear next bit} \\ \ell + 2^k & \text{to set next bit} \end{cases}$$

New technique: bit-sequence propagation

- Given $2^{8-k}(x - \ell) \equiv_{256} 0$ (or equivalently $x - \ell \equiv_{256} 2^k w$):
  1. Let
  $$\ell' = \begin{cases} \ell & \text{to clear next bit} \\ \ell + 2^k & \text{to set next bit} \end{cases}$$
  2. Impose $x - \ell' \equiv_{256} 2^{k+1} w'$

New technique: bit-sequence propagation

- Given $2^{8-k}(x - \ell) \equiv_{256} 0$ (or equivalently $x - \ell \equiv_{256} 2^k w$):
  1. Let
  $$\ell' = \begin{cases} \ell & \text{to clear next bit} \\ \ell + 2^k & \text{to set next bit} \end{cases}$$
  2. Impose $x - \ell' \equiv_{256} 2^{k+1} w'$
  3. Compute a Gröbner basis

New technique: bit-sequence propagation

- Given $2^{8-k}(x - \ell) \equiv_{256} 0$ (or equivalently $x - \ell \equiv_{256} 2^k w$):
  1. Let
  $$\ell' = \begin{cases} \ell & \text{to clear next bit} \\ \ell + 2^k & \text{to set next bit} \end{cases}$$
  2. Impose $x - \ell' \equiv_{256} 2^{k+1} w'$
  3. Compute a Gröbner basis
  4. Eliminate w'

New technique: bit-sequence propagation

- Given $2^{8-k}(x - \ell) \equiv_{256} 0$ (or equivalently $x - \ell \equiv_{256} 2^k w$):
  1. Let
  $$\ell' = \begin{cases} \ell & \text{to clear next bit} \\ \ell + 2^k & \text{to set next bit} \end{cases}$$
  2. Impose $x - \ell' \equiv_{256} 2^{k+1} w'$
  3. Compute a Gröbner basis
  4. Eliminate w'
- Can lead to other bits being set in the order of least significance

Example continued 1

- Consider again

$$B = \begin{cases} y^2 + 120x^2 + 123x + 48 \equiv_{256} 0 \\ yx + 65x^2 + 50x + 32 \equiv_{256} 0 \\ 2y + 63x^2 + 59x + 128 \equiv_{256} 0 \\ x^3 + 135x^2 + 100x + 64 \equiv_{256} 0 \\ 64x^2 + 192x \equiv_{256} 0 \end{cases}$$

Example continued 1

- Consider again

$$B = \begin{cases} y^2 + 120x^2 + 123x + 48 \equiv_{256} 0 \\ yx + 65x^2 + 50x + 32 \equiv_{256} 0 \\ 2y + 63x^2 + 59x + 128 \equiv_{256} 0 \\ x^3 + 135x^2 + 100x + 64 \equiv_{256} 0 \\ 64x^2 + 192x \equiv_{256} 0 \end{cases}$$

- No constraint $2^k(x - \ell) \equiv_{256} 0$ on x so no bits determined

Example continued 1

- Consider again

$$B = \begin{cases} y^2 + 120x^2 + 123x + 48 \equiv_{256} 0 \\ yx + 65x^2 + 50x + 32 \equiv_{256} 0 \\ 2y + 63x^2 + 59x + 128 \equiv_{256} 0 \\ x^3 + 135x^2 + 100x + 64 \equiv_{256} 0 \\ 64x^2 + 192x \equiv_{256} 0 \end{cases}$$

- No constraint $2^k(x - \ell) \equiv_{256} 0$ on x so no bits determined
- To clear bit 0 of x impose $x \equiv_{256} 2w$

Example continued 2

Add $x \equiv_{256} 2w$ and compute a Gröbner basis:

$$\begin{cases} y^2 + 120x^2 + 123x + 48 \equiv_{256} 0 \\ yx + 65x^2 + 50x + 32 \equiv_{256} 0 \\ 2y + 63x^2 + 59x + 128 \equiv_{256} 0 \\ x^3 + 135x^2 + 100x + 64 \equiv_{256} 0 \\ 64x^2 + 192x \equiv_{256} 0 \\ x - 2w \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} wx + 86x + 96 \equiv_{256} 0 \\ 2w + 255x \equiv_{256} 0 \\ y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 134x + 96 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 + 172x + 192 \equiv_{256} 0 \\ 64x \equiv_{256} 0 \end{cases}$$

Example continued 3

- Eliminate w:

$$\begin{cases} wx + 86x + 96 \equiv_{256} 0 \\ 2w + 255x \equiv_{256} 0 \\ y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 134x + 96 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 + 172x + 192 \equiv_{256} 0 \\ 64x \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 134x + 96 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 + 172x + 192 \equiv_{256} 0 \\ 64x \equiv_{256} 0 \end{cases}$$

Example continued 3

- Eliminate w:

$$\begin{cases} wx + 86x + 96 \equiv_{256} 0 \\ 2w + 255x \equiv_{256} 0 \\ y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 134x + 96 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 + 172x + 192 \equiv_{256} 0 \\ 64x \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 134x + 96 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 + 172x + 192 \equiv_{256} 0 \\ 64x \equiv_{256} 0 \end{cases}$$

- New equation $64x = 2^6(x - 0) \equiv_{256} 0$

Example continued 3

- Eliminate w:

$$\begin{cases} wx + 86x + 96 \equiv_{256} 0 \\ 2w + 255x \equiv_{256} 0 \\ y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 134x + 96 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 + 172x + 192 \equiv_{256} 0 \\ 64x \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 134x + 96 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 + 172x + 192 \equiv_{256} 0 \\ 64x \equiv_{256} 0 \end{cases}$$

- New equation $64x = 2^6(x - 0) \equiv_{256} 0$
- So 2 least significant bits of x are now determined

Example continued 4

- To clear bit 3 impose $x - 0 \equiv_{256} 8w$, calculate a Gröbner basis and eliminate w:

$$\begin{cases} y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 134x + 96 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 + 172x + 192 \equiv_{256} 0 \\ 64x \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 128 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 \equiv_{256} 0 \\ 2x + 160 \equiv_{256} 0 \end{cases}$$

Example continued 4

- To clear bit 3 impose $x - 0 \equiv_{256} 8w$, calculate a Gröbner basis and eliminate w:

$$\begin{cases} y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 134x + 96 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 + 172x + 192 \equiv_{256} 0 \\ 64x \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 128 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 \equiv_{256} 0 \\ 2x + 160 \equiv_{256} 0 \end{cases}$$

- New equation $2x + 160 \equiv_{256} 2x - 96 \equiv_{256} 2^1(x - 48) \equiv_{256} 0$

Example continued 4

- To clear bit 3 impose $x - 0 \equiv_{256} 8w$, calculate a Gröbner basis and eliminate w:

$$\begin{cases} y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 134x + 96 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 + 172x + 192 \equiv_{256} 0 \\ 64x \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 128 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 \equiv_{256} 0 \\ 2x + 160 \equiv_{256} 0 \end{cases}$$

- New equation $2x + 160 \equiv_{256} 2x - 96 \equiv_{256} 2^1(x - 48) \equiv_{256} 0$
- So the 7 (!) least significant bits of x are now determined

Example continued 5

- To clear bit 7 impose $x - 48 \equiv_{256} 256w \equiv_{256} 0$, calculate a Gröbner basis and eliminate w:

$$\begin{cases} y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 128 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 \equiv_{256} 0 \\ 2x + 160 \equiv_{256} 0 \\ x - 48 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y^2 + 64 \equiv_{256} 0 \\ 2y + 144 \equiv_{256} 0 \\ x + 208 \equiv_{256} 0 \end{cases}$$

Example continued 5

- To clear bit 7 impose $x - 48 \equiv_{256} 256w \equiv_{256} 0$, calculate a Gröbner basis and eliminate w:

$$\begin{cases} y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 128 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 \equiv_{256} 0 \\ 2x + 160 \equiv_{256} 0 \\ x - 48 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y^2 + 64 \equiv_{256} 0 \\ 2y + 144 \equiv_{256} 0 \\ x + 208 \equiv_{256} 0 \end{cases}$$

- Now x is fully determined

Example continued 5

- To clear bit 7 impose $x - 48 \equiv_{256} 256w \equiv_{256} 0$, calculate a Gröbner basis and eliminate w:

$$\begin{cases} y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 128 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 \equiv_{256} 0 \\ 2x + 160 \equiv_{256} 0 \\ x - 48 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y^2 + 64 \equiv_{256} 0 \\ 2y + 144 \equiv_{256} 0 \\ x + 208 \equiv_{256} 0 \end{cases}$$

- Now x is fully determined
- New equation $2y + 144 \equiv_{256} 2y - 112 \equiv_{256} 2^1(y - 56) \equiv_{256} 0$

Example continued 5

- To clear bit 7 impose $x - 48 \equiv_{256} 256w \equiv_{256} 0$, calculate a Gröbner basis and eliminate w:

$$\begin{cases} y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 128 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 \equiv_{256} 0 \\ 2x + 160 \equiv_{256} 0 \\ x - 48 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y^2 + 64 \equiv_{256} 0 \\ 2y + 144 \equiv_{256} 0 \\ x + 208 \equiv_{256} 0 \end{cases}$$

- Now x is fully determined
- New equation $2y + 144 \equiv_{256} 2y - 112 \equiv_{256} 2^1(y - 56) \equiv_{256} 0$
- Only the most significant bit of y remains undetermined

Example continued 6

- To clear bit 7 impose $y - 56 \equiv_{256} 256w \equiv_{256} 0$, calculate a Gröbner basis and eliminate w:

$$\begin{cases} y^2 + 64 \equiv_{256} 0 \\ 2y + 144 \equiv_{256} 0 \\ x + 208 \equiv_{256} 0 \\ y - 56 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y + 200 \equiv_{256} 0 \\ x + 208 \equiv_{256} 0 \\ 128 \equiv_{256} 0 \end{cases}$$

Example continued 6

- To clear bit 7 impose $y - 56 \equiv_{256} 256w \equiv_{256} 0$, calculate a Gröbner basis and eliminate w:

$$\begin{cases} y^2 + 64 \equiv_{256} 0 \\ 2y + 144 \equiv_{256} 0 \\ x + 208 \equiv_{256} 0 \\ y - 56 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y + 200 \equiv_{256} 0 \\ x + 208 \equiv_{256} 0 \\ 128 \equiv_{256} 0 \end{cases}$$

- Alternatively, to set bit 7 impose $y - (56 + 128) \equiv_{256} 256w \equiv_{256} 0$, calculate a Gröbner basis and eliminate w:

$$\begin{cases} y^2 + 64 \equiv_{256} 0 \\ 2y + 144 \equiv_{256} 0 \\ x + 208 \equiv_{256} 0 \\ y - 184 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y + 72 \equiv_{256} 0 \\ x + 208 \equiv_{256} 0 \\ 128 \equiv_{256} 0 \end{cases}$$

Example continued 6

- To clear bit 7 impose $y - 56 \equiv_{256} 256w \equiv_{256} 0$, calculate a Gröbner basis and eliminate w:

$$\begin{cases} y^2 + 64 \equiv_{256} 0 \\ 2y + 144 \equiv_{256} 0 \\ x + 208 \equiv_{256} 0 \\ y - 56 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y + 200 \equiv_{256} 0 \\ x + 208 \equiv_{256} 0 \\ 128 \equiv_{256} 0 \end{cases}$$

- Alternatively, to set bit 7 impose $y - (56 + 128) \equiv_{256} 256w \equiv_{256} 0$, calculate a Gröbner basis and eliminate w:

$$\begin{cases} y^2 + 64 \equiv_{256} 0 \\ 2y + 144 \equiv_{256} 0 \\ x + 208 \equiv_{256} 0 \\ y - 184 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y + 72 \equiv_{256} 0 \\ x + 208 \equiv_{256} 0 \\ 128 \equiv_{256} 0 \end{cases}$$

- Both systems contain an inconsistent constraint $128 \equiv_{256} 0$

Example continued 7

- Backtrack to the last point where inconsistency is not known

Example continued 7

- Backtrack to the last point where inconsistency is not known
- This time set bit 7 of x by adding $x - (48 + 128) \equiv_{256} 256w \equiv_{256} 0$, computing a Gröbner basis and eliminating w:

$$\begin{cases} y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 128 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 \equiv_{256} 0 \\ 2x + 160 \equiv_{256} 0 \\ x - 176 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y^2 + 192 \equiv_{256} 0 \\ 2y + 16 \equiv_{256} 0 \\ x + 80 \equiv_{256} 0 \end{cases}$$

Example continued 7

- Backtrack to the last point where inconsistency is not known
- This time set bit 7 of x by adding $x - (48 + 128) \equiv_{256} 256w \equiv_{256} 0$, computing a Gröbner basis and eliminating w:

$$\begin{cases} y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 128 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 \equiv_{256} 0 \\ 2x + 160 \equiv_{256} 0 \\ x - 176 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y^2 + 192 \equiv_{256} 0 \\ 2y + 16 \equiv_{256} 0 \\ x + 80 \equiv_{256} 0 \end{cases}$$

- Now x is fully determined

Example continued 7

- Backtrack to the last point where inconsistency is not known
- This time set bit 7 of x by adding $x - (48 + 128) \equiv_{256} 256w \equiv_{256} 0$, computing a Gröbner basis and eliminating w:

$$\begin{cases} y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 128 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 \equiv_{256} 0 \\ 2x + 160 \equiv_{256} 0 \\ x - 176 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y^2 + 192 \equiv_{256} 0 \\ 2y + 16 \equiv_{256} 0 \\ x + 80 \equiv_{256} 0 \end{cases}$$

- Now x is fully determined
- New equation $2y + 16 \equiv_{256} 2y - 240 \equiv_{256} 2^1(y - 120) \equiv_{256} 0$

Example continued 7

- Backtrack to the last point where inconsistency is not known
- This time set bit 7 of x by adding $x - (48 + 128) \equiv_{256} 256w \equiv_{256} 0$, computing a Gröbner basis and eliminating w:

$$\begin{cases} y^2 + 219x + 48 \equiv_{256} 0 \\ yx + 128 \equiv_{256} 0 \\ 2y + 231x + 64 \equiv_{256} 0 \\ x^2 \equiv_{256} 0 \\ 2x + 160 \equiv_{256} 0 \\ x - 176 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y^2 + 192 \equiv_{256} 0 \\ 2y + 16 \equiv_{256} 0 \\ x + 80 \equiv_{256} 0 \end{cases}$$

- Now x is fully determined
- New equation $2y + 16 \equiv_{256} 2y - 240 \equiv_{256} 2^1(y - 120) \equiv_{256} 0$
- So only the most significant bit of y is undetermined

Example continued 8

- To clear bit 7 impose $y - 120 \equiv_{256} 256w \equiv_{256} 0$, calculate a Gröbner basis and eliminate w:

$$\begin{cases} y^2 + 64 \equiv_{256} 0 \\ 2y + 144 \equiv_{256} 0 \\ x + 208 \equiv_{256} 0 \\ y - 120 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y + 136 \equiv_{256} 0 \\ x + 80 \equiv_{256} 0 \end{cases}$$

Example continued 8

- To clear bit 7 impose $y - 120 \equiv_{256} 256w \equiv_{256} 0$, calculate a Gröbner basis and eliminate w:

$$\begin{cases} y^2 + 64 \equiv_{256} 0 \\ 2y + 144 \equiv_{256} 0 \\ x + 208 \equiv_{256} 0 \\ y - 120 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y + 136 \equiv_{256} 0 \\ x + 80 \equiv_{256} 0 \end{cases}$$

- To set bit 7 impose $y - (120 + 128) \equiv_{256} 256w \equiv_{256} 0$, calculate a Gröbner basis and eliminate w:

$$\begin{cases} y^2 + 64 \equiv_{256} 0 \\ 2y + 144 \equiv_{256} 0 \\ x + 208 \equiv_{256} 0 \\ y - 248 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y + 8 \equiv_{256} 0 \\ x + 80 \equiv_{256} 0 \end{cases}$$

Example continued 8

- To clear bit 7 impose $y - 120 \equiv_{256} 256w \equiv_{256} 0$, calculate a Gröbner basis and eliminate w:

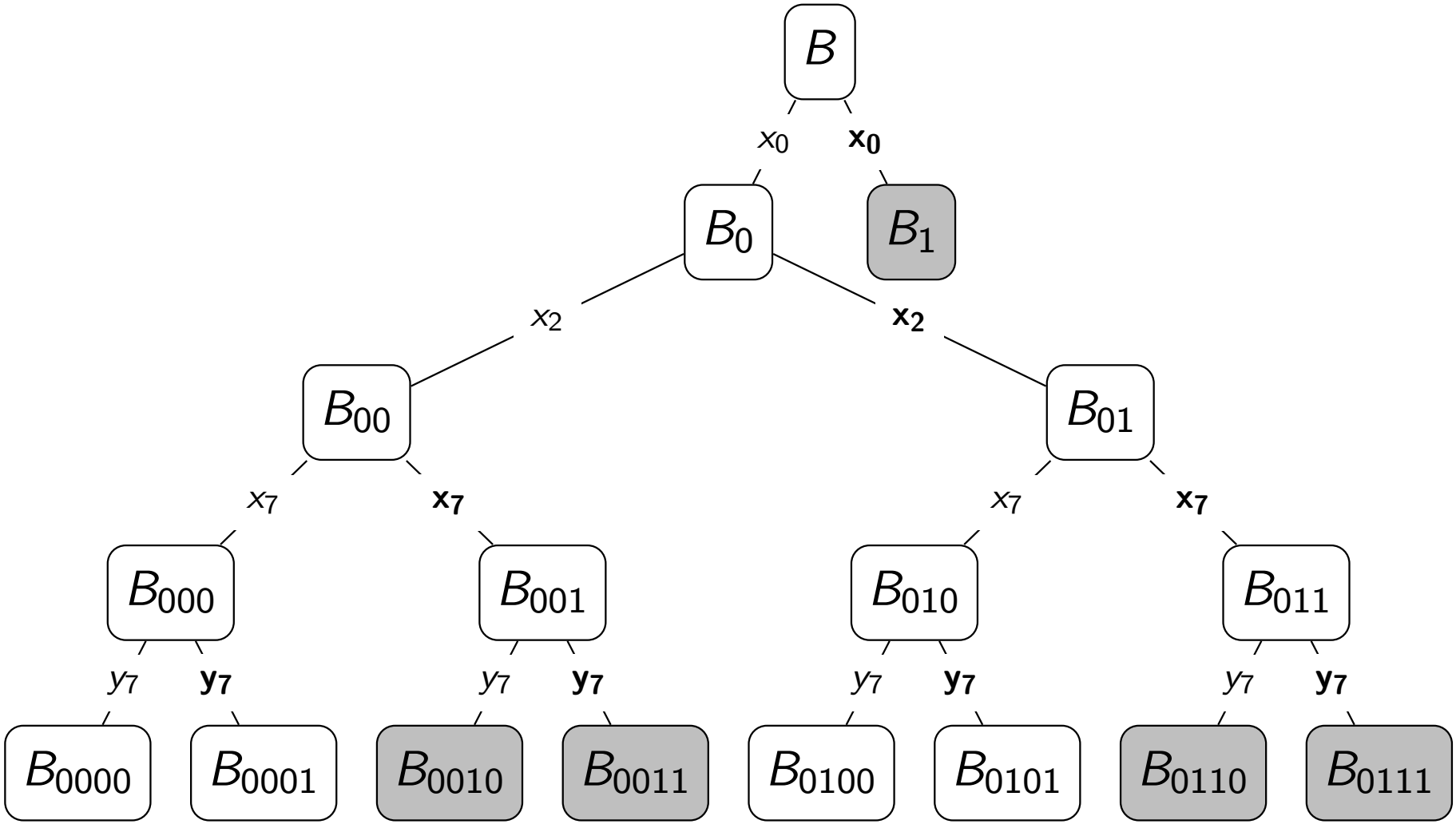
$$\begin{cases} y^2 + 64 \equiv_{256} 0 \\ 2y + 144 \equiv_{256} 0 \\ x + 208 \equiv_{256} 0 \\ y - 120 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y + 136 \equiv_{256} 0 \\ x + 80 \equiv_{256} 0 \end{cases}$$

- To set bit 7 impose $y - (120 + 128) \equiv_{256} 256w \equiv_{256} 0$, calculate a Gröbner basis and eliminate w:

$$\begin{cases} y^2 + 64 \equiv_{256} 0 \\ 2y + 144 \equiv_{256} 0 \\ x + 208 \equiv_{256} 0 \\ y - 248 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y + 8 \equiv_{256} 0 \\ x + 80 \equiv_{256} 0 \end{cases}$$

- Both assignments correspond to solutions

Recap: Solution tree



Symbolic approach

- New idea: set the bits symbolically

Symbolic approach

- New idea: set the bits symbolically
- Consider again

$$B = \begin{cases} y^2 + 120x^2 + 123x + 48 \equiv_{256} 0 \\ yx + 65x^2 + 50x + 32 \equiv_{256} 0 \\ 2y + 63x^2 + 59x + 128 \equiv_{256} 0 \\ x^3 + 135x^2 + 100x + 64 \equiv_{256} 0 \\ 64x^2 + 192x \equiv_{256} 0 \end{cases}$$

Symbolic approach

- New idea: set the bits symbolically
- Consider again

$$B = \begin{cases} y^2 + 120x^2 + 123x + 48 \equiv_{256} 0 \\ yx + 65x^2 + 50x + 32 \equiv_{256} 0 \\ 2y + 63x^2 + 59x + 128 \equiv_{256} 0 \\ x^3 + 135x^2 + 100x + 64 \equiv_{256} 0 \\ 64x^2 + 192x \equiv_{256} 0 \end{cases}$$

- To assign next bit impose $x - b_1 \equiv_{256} 2w$ with $b_1^2 \equiv_{256} b_1$

Symbolic example 1

- Add $x - b_1 \equiv_{256} 2w$ and $b_1^2 \equiv_{256} b_1$, compute a Gröbner basis and eliminate w:

$$\begin{cases} y^2 + 120x^2 + 123x + 48 \equiv_{256} 0 \\ yx + 65x^2 + 50x + 32 \equiv_{256} 0 \\ 2y + 63x^2 + 59x + 128 \equiv_{256} 0 \\ x^3 + 135x^2 + 100x + 64 \equiv_{256} 0 \\ 64x^2 + 192x \equiv_{256} 0 \\ x - b_1 - 2w \equiv_{256} 0 \\ b_1^2 - b_1 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y^2 + 219x + 216b_1 + 48 \equiv_{256} 0 \\ yx + 6x + 181b_1 + 96 \equiv_{256} 0 \\ yb_1 + 183b_1 \equiv_{256} 0 \\ 2y + 103x + 203b_1 + 64 \equiv_{256} 0 \\ x^2 + 44x + 139b_1 + 192 \equiv_{256} 0 \\ xb_1 + 91b_1 \equiv_{256} 0 \\ 64x + 192b_1 \equiv_{256} 0 \\ b_1^2 + 255b_1 \equiv_{256} 0 \end{cases}$$

Symbolic example 1

- Add $x - b_1 \equiv_{256} 2w$ and $b_1^2 \equiv_{256} b_1$, compute a Gröbner basis and eliminate w:

$$\begin{cases} y^2 + 120x^2 + 123x + 48 \equiv_{256} 0 \\ yx + 65x^2 + 50x + 32 \equiv_{256} 0 \\ 2y + 63x^2 + 59x + 128 \equiv_{256} 0 \\ x^3 + 135x^2 + 100x + 64 \equiv_{256} 0 \\ 64x^2 + 192x \equiv_{256} 0 \\ x - b_1 - 2w \equiv_{256} 0 \\ b_1^2 - b_1 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y^2 + 219x + 216b_1 + 48 \equiv_{256} 0 \\ yx + 6x + 181b_1 + 96 \equiv_{256} 0 \\ yb_1 + 183b_1 \equiv_{256} 0 \\ 2y + 103x + 203b_1 + 64 \equiv_{256} 0 \\ x^2 + 44x + 139b_1 + 192 \equiv_{256} 0 \\ xb_1 + 91b_1 \equiv_{256} 0 \\ 64x + 192b_1 \equiv_{256} 0 \\ b_1^2 + 255b_1 \equiv_{256} 0 \end{cases}$$

- New constraint $64x + 192b_1 \equiv_{256} 64x - 64b_1 \equiv_{256} 64(x - b_1) \equiv_{256} 0$

Symbolic example 1

- Add $x - b_1 \equiv_{256} 2w$ and $b_1^2 \equiv_{256} b_1$, compute a Gröbner basis and eliminate w:

$$\begin{cases} y^2 + 120x^2 + 123x + 48 \equiv_{256} 0 \\ yx + 65x^2 + 50x + 32 \equiv_{256} 0 \\ 2y + 63x^2 + 59x + 128 \equiv_{256} 0 \\ x^3 + 135x^2 + 100x + 64 \equiv_{256} 0 \\ 64x^2 + 192x \equiv_{256} 0 \\ x - b_1 - 2w \equiv_{256} 0 \\ b_1^2 - b_1 \equiv_{256} 0 \end{cases} \Rightarrow \begin{cases} y^2 + 219x + 216b_1 + 48 \equiv_{256} 0 \\ yx + 6x + 181b_1 + 96 \equiv_{256} 0 \\ yb_1 + 183b_1 \equiv_{256} 0 \\ 2y + 103x + 203b_1 + 64 \equiv_{256} 0 \\ x^2 + 44x + 139b_1 + 192 \equiv_{256} 0 \\ xb_1 + 91b_1 \equiv_{256} 0 \\ 64x + 192b_1 \equiv_{256} 0 \\ b_1^2 + 255b_1 \equiv_{256} 0 \end{cases}$$

- New constraint $64x + 192b_1 \equiv_{256} 64x - 64b_1 \equiv_{256} 64(x - b_1) \equiv_{256} 0$
- So 2 least significant bits of x determined, albeit parametric on b_1

Symbolic example 2

- To set bit 3 impose $x - 4b_2 \equiv_{256} 8w$ and $b_2^2 \equiv_{256} b_2$, calculate a Gröbner basis and eliminate w:

$$\ldots \Rightarrow \begin{cases} y^2 + 219x + 216b_1 + 48 \equiv_{256} 0 \\ yx + 184b_2 + 187b_1 + 128 \equiv_{256} 0 \\ yb_1 + 183b_1 \equiv_{256} 0 \\ 2y + 103x + 203b_1 + 64 \equiv_{256} 0 \\ x^2 + 240b_2 + 183b_1 \equiv_{256} 0 \\ xb_1 + 91b_1 \equiv_{256} 0 \\ 2x + 24b_2 + 254b_1 + 160 \equiv_{256} 0 \\ b_2^2 + 255b_2 \equiv_{256} 0 \\ 4b_2b_1 + 252b_1 \equiv_{256} 0 \\ b_1^2 + 255b_1 \equiv_{256} 0 \end{cases}$$

Symbolic example 2

- To set bit 3 impose $x - 4b_2 \equiv_{256} 8w$ and $b_2^2 \equiv_{256} b_2$, calculate a Gröbner basis and eliminate w:

$$\ldots \Rightarrow \begin{cases} y^2 + 219x + 216b_1 + 48 \equiv_{256} 0 \\ yx + 184b_2 + 187b_1 + 128 \equiv_{256} 0 \\ yb_1 + 183b_1 \equiv_{256} 0 \\ 2y + 103x + 203b_1 + 64 \equiv_{256} 0 \\ x^2 + 240b_2 + 183b_1 \equiv_{256} 0 \\ xb_1 + 91b_1 \equiv_{256} 0 \\ 2x + 24b_2 + 254b_1 + 160 \equiv_{256} 0 \\ b_2^2 + 255b_2 \equiv_{256} 0 \\ 4b_2b_1 + 252b_1 \equiv_{256} 0 \\ b_1^2 + 255b_1 \equiv_{256} 0 \end{cases}$$

- New $2x + 24b_2 + 254b_1 + 160 \equiv_{256} 2(x - (116b_2 + b_1 + 48)) \equiv_{256} 0$

Symbolic example 2

- To set bit 3 impose $x - 4b_2 \equiv_{256} 8w$ and $b_2^2 \equiv_{256} b_2$, calculate a Gröbner basis and eliminate w:

$$\ldots \Rightarrow \begin{cases} y^2 + 219x + 216b_1 + 48 \equiv_{256} 0 \\ yx + 184b_2 + 187b_1 + 128 \equiv_{256} 0 \\ yb_1 + 183b_1 \equiv_{256} 0 \\ 2y + 103x + 203b_1 + 64 \equiv_{256} 0 \\ x^2 + 240b_2 + 183b_1 \equiv_{256} 0 \\ xb_1 + 91b_1 \equiv_{256} 0 \\ 2x + 24b_2 + 254b_1 + 160 \equiv_{256} 0 \\ b_2^2 + 255b_2 \equiv_{256} 0 \\ 4b_2b_1 + 252b_1 \equiv_{256} 0 \\ b_1^2 + 255b_1 \equiv_{256} 0 \end{cases}$$

- New $2x + 24b_2 + 254b_1 + 160 \equiv_{256} 2(x - (116b_2 + b_1 + 48)) \equiv_{256} 0$
- So only most significant bit of x is yet to be set

Symbolic example 3

- To set bit 7 impose $x - (128b_3 + 116b_2 + b_1 + 48) \equiv_{256} 256w \equiv_{256} 0$ and $b_3^2 \equiv_{256} b_3$, calculate a Gröbner basis and eliminate w:

$$\ldots \Rightarrow \begin{cases} y^2 + 128b_3 + 60b_2 + 179b_1 + 64 \equiv_{256} 0 \\ yb_1 + 183b_1 \equiv_{256} 0 \\ 2y + 128b_3 + 172b_2 + 50b_1 + 144 \equiv_{256} 0 \\ x + 128b_3 + 140b_2 + 255b_1 + 208 \equiv_{256} 0 \\ b_3^3 + 255b_3 \equiv_{256} 0 \\ 128b_3b_1 \equiv_{256} 0 \\ b_2^2 + 255b_2 \equiv_{256} 0 \\ 4b_2b_1 + 252b_1 \equiv_{256} 0 \\ b_1^2 + 255b_1 \equiv_{256} 0 \end{cases}$$

Symbolic example 3

- To set bit 7 impose $x - (128b_3 + 116b_2 + b_1 + 48) \equiv_{256} 256w \equiv_{256} 0$ and $b_3^2 \equiv_{256} b_3$, calculate a Gröbner basis and eliminate w:

$$\ldots \Rightarrow \begin{cases} y^2 + 128b_3 + 60b_2 + 179b_1 + 64 \equiv_{256} 0 \\ yb_1 + 183b_1 \equiv_{256} 0 \\ 2y + 128b_3 + 172b_2 + 50b_1 + 144 \equiv_{256} 0 \\ x + 128b_3 + 140b_2 + 255b_1 + 208 \equiv_{256} 0 \\ b_3^3 + 255b_3 \equiv_{256} 0 \\ 128b_3b_1 \equiv_{256} 0 \\ b_2^2 + 255b_2 \equiv_{256} 0 \\ 4b_2b_1 + 252b_1 \equiv_{256} 0 \\ b_1^2 + 255b_1 \equiv_{256} 0 \end{cases}$$

- Now x is fully determined

Symbolic example 3

- To set bit 7 impose $x - (128b_3 + 116b_2 + b_1 + 48) \equiv_{256} 256w \equiv_{256} 0$ and $b_3^2 \equiv_{256} b_3$, calculate a Gröbner basis and eliminate w:

$$\ldots \Rightarrow \begin{cases} y^2 + 128b_3 + 60b_2 + 179b_1 + 64 \equiv_{256} 0 \\ yb_1 + 183b_1 \equiv_{256} 0 \\ 2y + 128b_3 + 172b_2 + 50b_1 + 144 \equiv_{256} 0 \\ x + 128b_3 + 140b_2 + 255b_1 + 208 \equiv_{256} 0 \\ b_3^3 + 255b_3 \equiv_{256} 0 \\ 128b_3b_1 \equiv_{256} 0 \\ b_2^2 + 255b_2 \equiv_{256} 0 \\ 4b_2b_1 + 252b_1 \equiv_{256} 0 \\ b_1^2 + 255b_1 \equiv_{256} 0 \end{cases}$$

- Now x is fully determined
- Only most significant bit of y is yet to be set

Symbolic example 4

- To assign bit 7 impose
  $y - (128b_4 + 64b_3 + 42b_2 + 103b_1 + 56) \equiv_{256} 256w$ and $b_4^2 \equiv_{256} b_4$, calculate a Gröbner basis and eliminate w:

$$\ldots \Rightarrow \begin{cases} y + 128b_4 + 192b_3 + 214b_2 + 153b_1 + 200 \equiv_{256} 0 \\ x + 12b_2 + 255b_1 + 80 \equiv_{256} 0 \\ b_4^2 + 255b_4 \equiv_{256} 0 \\ 128b_4b_1 + 128b_1 \equiv_{256} 0 \\ b_3^2 + 255b_3 \equiv_{256} 0 \\ 64b_3b_1 \equiv_{256} 0 \\ 128b_3 + 128b_2 + 128 \equiv_{256} 0 \\ b_2^2 + 255b_2 \equiv_{256} 0 \\ 2b_2b_1 + 254b_1 \equiv_{256} 0 \\ b_1^2 + 255b_1 \equiv_{256} 0 \end{cases}$$

Symbolic example 4

- To assign bit 7 impose
  $y - (128b_4 + 64b_3 + 42b_2 + 103b_1 + 56) \equiv_{256} 256w$ and $b_4^2 \equiv_{256} b_4$, calculate a Gröbner basis and eliminate w:

$$\ldots \Rightarrow \begin{cases} y + 128b_4 + 192b_3 + 214b_2 + 153b_1 + 200 \equiv_{256} 0 \\ x + 12b_2 + 255b_1 + 80 \equiv_{256} 0 \\ b_4^2 + 255b_4 \equiv_{256} 0 \\ 128b_4b_1 + 128b_1 \equiv_{256} 0 \\ b_3^2 + 255b_3 \equiv_{256} 0 \\ 64b_3b_1 \equiv_{256} 0 \\ 128b_3 + 128b_2 + 128 \equiv_{256} 0 \\ b_2^2 + 255b_2 \equiv_{256} 0 \\ 2b_2b_1 + 254b_1 \equiv_{256} 0 \\ b_1^2 + 255b_1 \equiv_{256} 0 \end{cases}$$

- x and y are determined as linear combinations of b_4, b_3, b_2, b_1

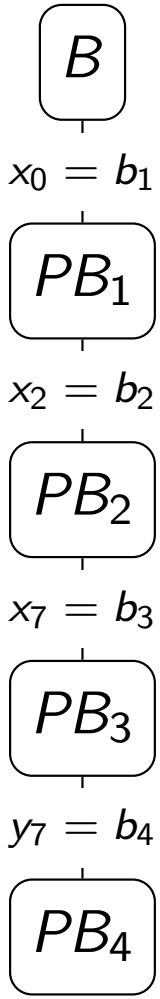
Symbolic example 4

- To assign bit 7 impose
  $y - (128b_4 + 64b_3 + 42b_2 + 103b_1 + 56) \equiv_{256} 256w$ and $b_4^2 \equiv_{256} b_4$, calculate a Gröbner basis and eliminate w:

$$\ldots \Rightarrow \begin{cases} y + 128b_4 + 192b_3 + 214b_2 + 153b_1 + 200 \equiv_{256} 0 \\ x + 12b_2 + 255b_1 + 80 \equiv_{256} 0 \\ b_4^2 + 255b_4 \equiv_{256} 0 \\ 128b_4b_1 + 128b_1 \equiv_{256} 0 \\ b_3^2 + 255b_3 \equiv_{256} 0 \\ 64b_3b_1 \equiv_{256} 0 \\ 128b_3 + 128b_2 + 128 \equiv_{256} 0 \\ b_2^2 + 255b_2 \equiv_{256} 0 \\ 2b_2b_1 + 254b_1 \equiv_{256} 0 \\ b_1^2 + 255b_1 \equiv_{256} 0 \end{cases}$$

- x and y are determined as linear combinations of b_4, b_3, b_2, b_1
- Observe that the remaining polynomials are non-linear modulo pseudo-boolean constraints

Recap: Solution tree



Symbolic example 5

- The pseudo-boolean constraints form a residue system:

$$\begin{aligned} 128b_4b_1 + 128b_1 &\equiv_{256} 0 \\ 64b_3b_1 &\equiv_{256} 0 \\ 128b_3 + 128b_2 + 128 &\equiv_{256} 0 \\ 2b_2b_1 + 254b_1 &\equiv_{256} 0 \end{aligned}$$

Symbolic example 5

- The pseudo-boolean constraints form a residue system:

$$\begin{aligned} 128b_4b_1 + 128b_1 &\equiv_{256} 0 \\ 64b_3b_1 &\equiv_{256} 0 \\ 128b_3 + 128b_2 + 128 &\equiv_{256} 0 \\ 2b_2b_1 + 254b_1 &\equiv_{256} 0 \end{aligned}$$

- Residue system is lowered:

$$\begin{aligned} 128b_4b_1 + 128b_1 \equiv_{256} 0 &\iff b_4b_1 + b_1 \equiv_2 0 \\ 64b_3b_1 \equiv_{256} 0 &\iff b_3b_1 \equiv_4 0 \\ 128b_3 + 128b_2 + 128 \equiv_{256} 0 &\iff b_3 + b_2 + 1 \equiv_2 0 \\ 2b_2b_1 + 254b_1 \equiv_{256} 0 &\iff b_2b_1 - b_1 \equiv_{128} 0 \end{aligned}$$

Symbolic example 5

- The pseudo-boolean constraints form a residue system:

$$\begin{aligned} 128b_4b_1 + 128b_1 &\equiv_{256} 0 \\ 64b_3b_1 &\equiv_{256} 0 \\ 128b_3 + 128b_2 + 128 &\equiv_{256} 0 \\ 2b_2b_1 + 254b_1 &\equiv_{256} 0 \end{aligned}$$

- Residue system is lowered:

$$\begin{aligned} 128b_4b_1 + 128b_1 \equiv_{256} 0 &\iff b_4b_1 + b_1 \equiv_2 0 \\ 64b_3b_1 \equiv_{256} 0 &\iff b_3b_1 \equiv_4 0 \\ 128b_3 + 128b_2 + 128 \equiv_{256} 0 &\iff b_3 + b_2 + 1 \equiv_2 0 \\ 2b_2b_1 + 254b_1 \equiv_{256} 0 &\iff b_2b_1 - b_1 \equiv_{128} 0 \end{aligned}$$

- Lowered system converted to propositional formulae:

$$\begin{aligned} b_4b_1 + b_1 \equiv_2 0 &\iff ((b_4 \wedge b_1) \iff b_1) \\ b_3b_1 \equiv_4 0 &\iff \neg b_3 \vee \neg b_1 \\ b_3 + b_2 + 1 \equiv_2 0 &\iff b_3 \oplus b_2 \\ b_2b_1 - b_1 \equiv_{128} 0 &\iff (\neg b_2 \wedge \neg b_1) \vee b_2 \end{aligned}$$

New approach to solving polynomials over bit-vectors

- Bit-sequence propagation reduces polynomials into pseudo-booleans

New approach to solving polynomials over bit-vectors

- Bit-sequence propagation reduces polynomials into pseudo-booleans
- Pseudo-booleans lowered to propositional constraints (clauses)

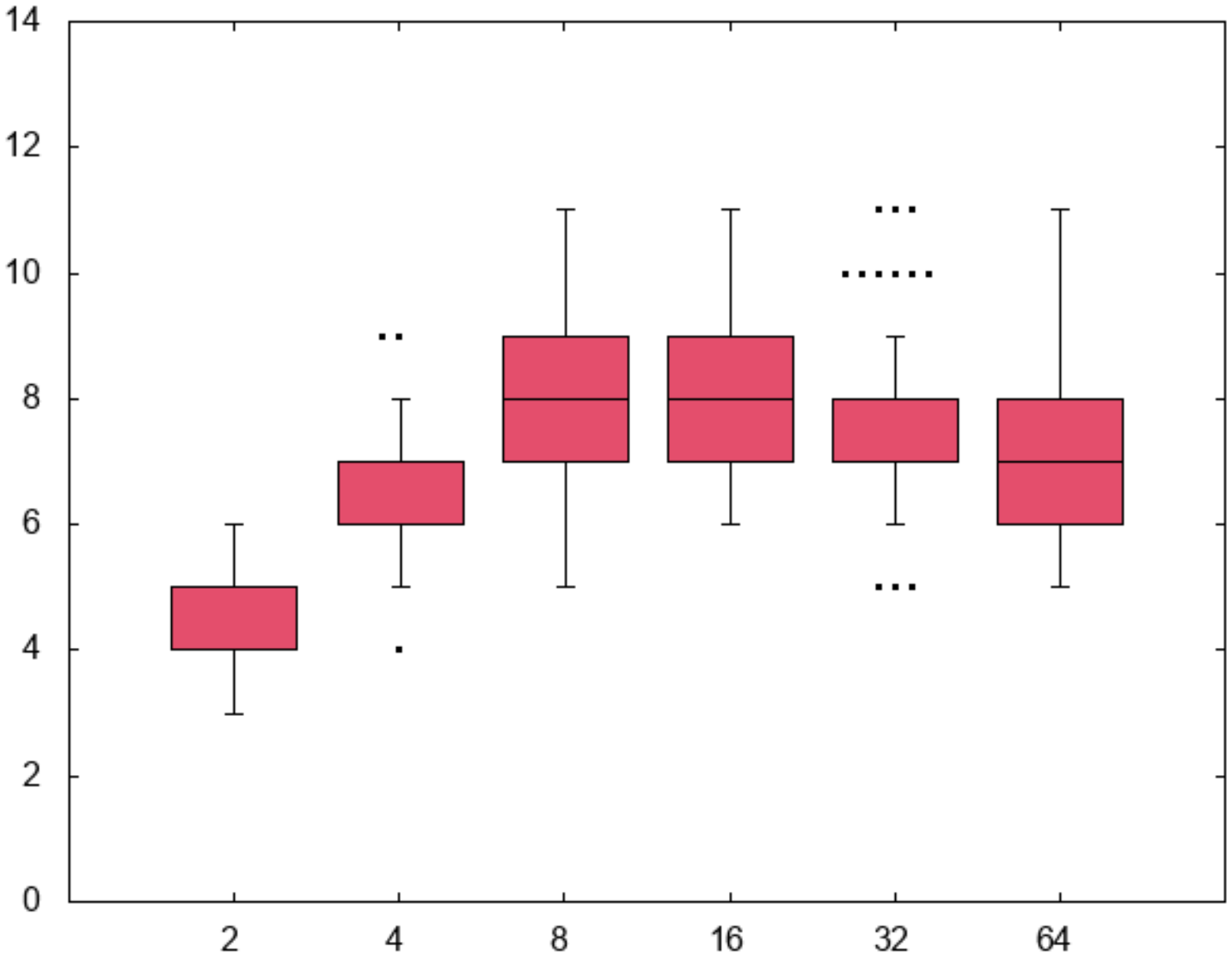
New approach to solving polynomials over bit-vectors

- Bit-sequence propagation reduces polynomials into pseudo-booleans
- Pseudo-booleans lowered to propositional constraints (clauses)
- Clauses solved with SAT (using learning)

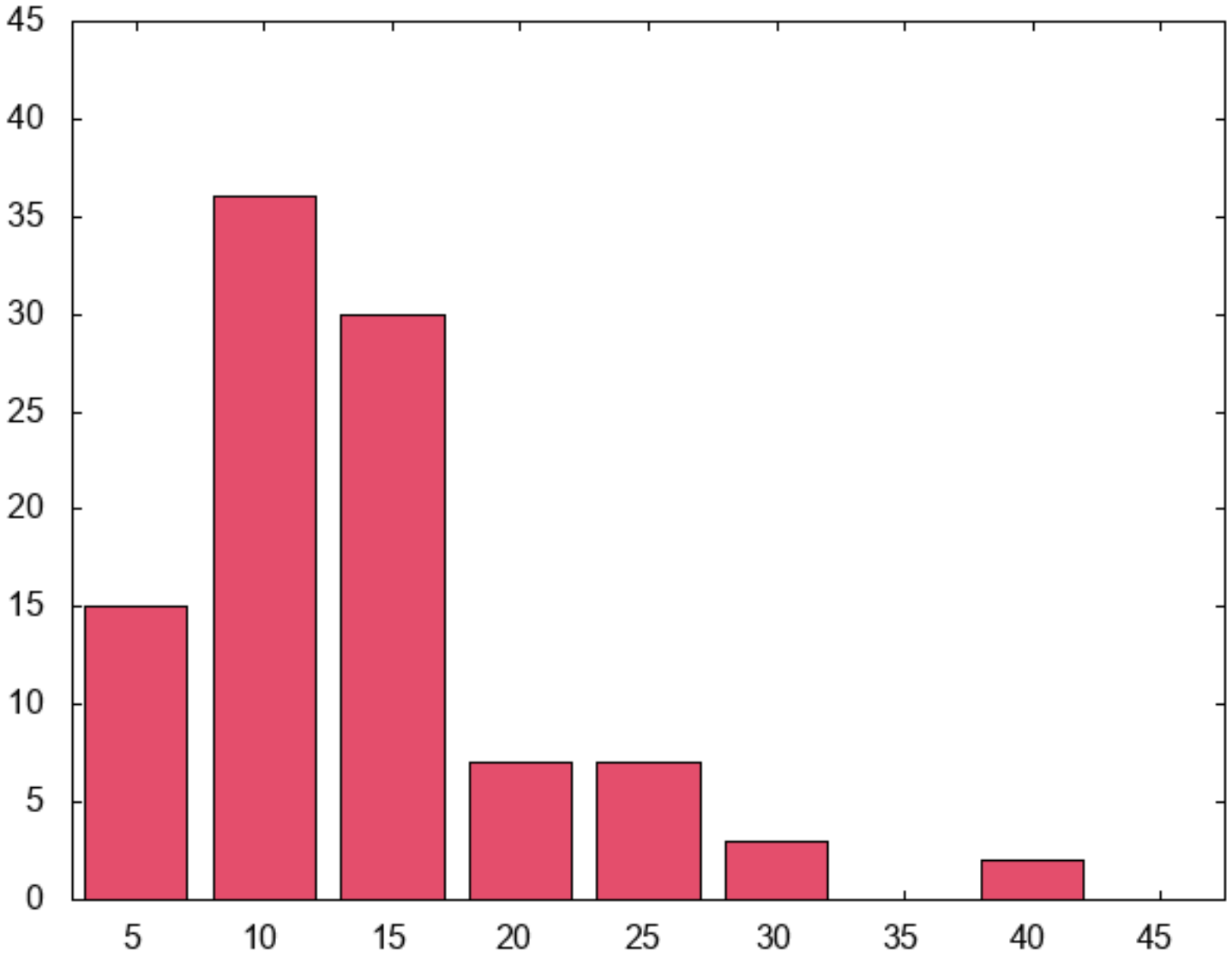
New approach to solving polynomials over bit-vectors

- Bit-sequence propagation reduces polynomials into pseudo-booleans
- Pseudo-booleans lowered to propositional constraints (clauses)
- Clauses solved with SAT (using learning)
- Symbolic variables (b_i) back-substituted into bit-vectors (x and y)

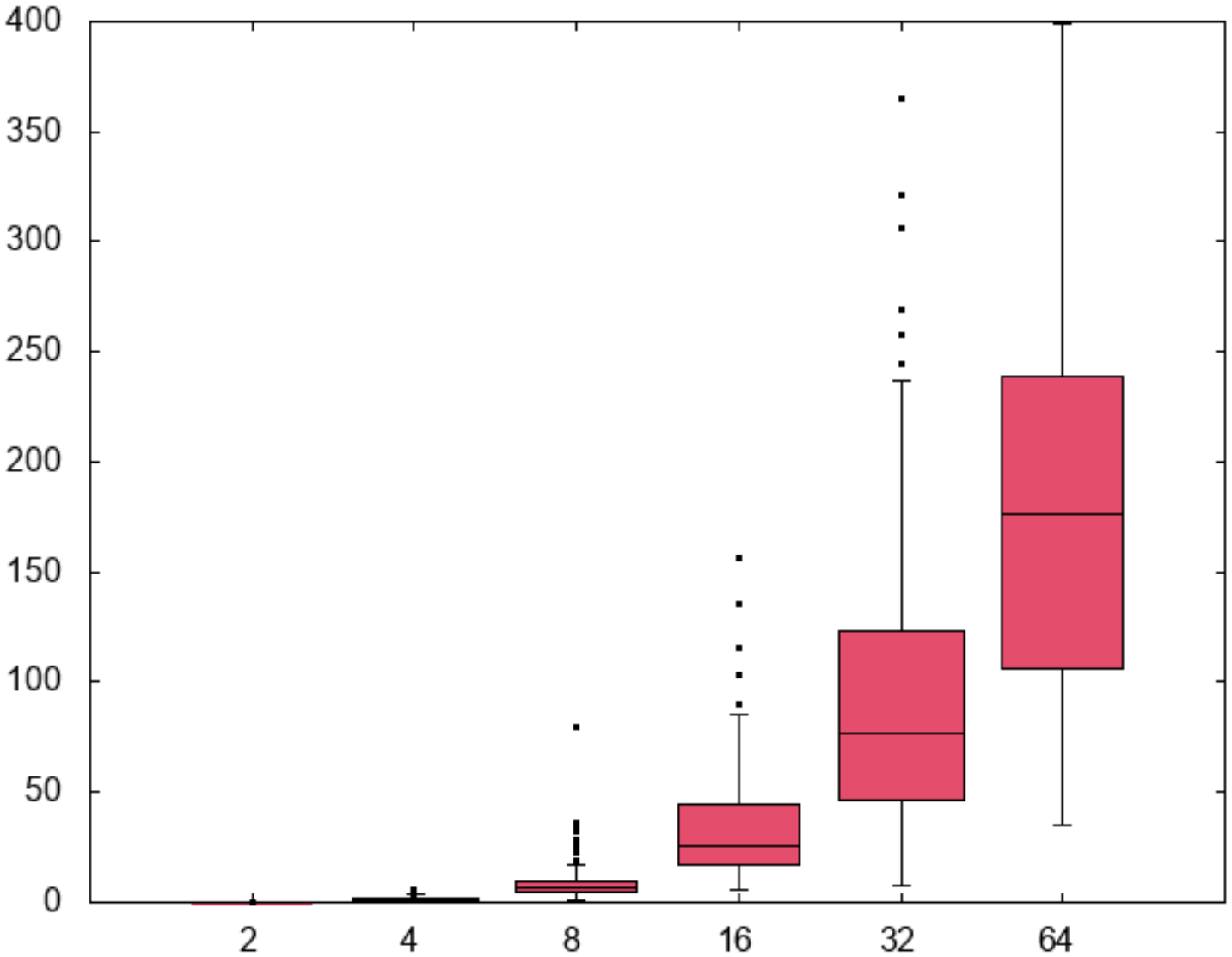
Number of symbolic variables vs bits for 4 var systems



Ratio of gates to products for 4 vars and 32 bits



Timings in seconds versus bits for 4 variable systems



Questions?