ON THE PROBLEM OF ARITHMETIC CIRCUIT VERIFICATION USING COMPUTER ALGEBRA

Daniela Ritirc joint work with Armin Biere and Manuel Kauers
Johannes Kepler University
Linz, Austria

Theory Reading Group Meeting
April 23, 2018
Stockholm, Sweden

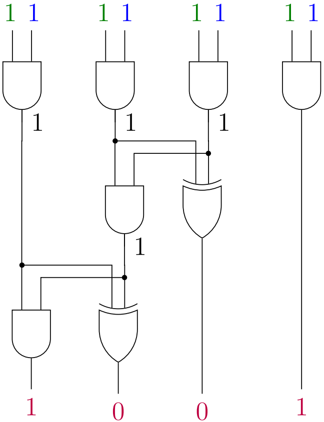
JOHANNES KEPLER UNIVERSITY LINZ

RiSE Rigorous Systems Engineering

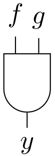
Example: 2 Bit - Binary Multiplication

$$\begin{array}{ccccc}
1 & 1 & \cdot & 1 & 1 \\
\hline
 & & & 1 & 1 \\
 & & 1 & 1 & \\
 & {}_{1} & {}_{1} & {}_{0} & \\
\hline
 & 1 & 0 & 0 & 1
\end{array}$$

$$3 \cdot 3 = 9$$

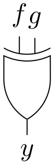


Example: 2 Bit - Binary Multiplication



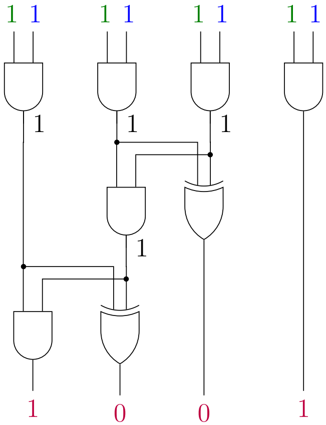
AND-Gate

f	g	y
0	0	0
0	1	0
1	0	0
1	1	1



XOR-Gate

f	g	y
0	0	0
0	1	1
1	0	1
1	1	0

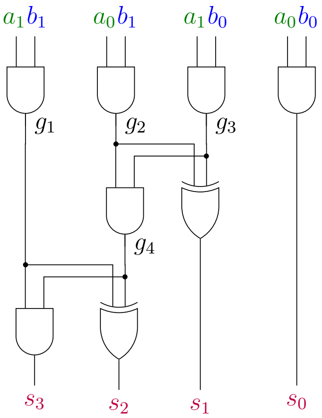


Motivation & Solving Techniques

Given: Gate-level multiplier for fixed bit-width n.

Question: For all possible $a_i, b_i \in \mathbb{B}$:

$$(2a_1 + a_0) * (2b_1 + b_0) = 8s_3 + 4s_2 + 2s_1 + s_0?$$



Solving Techniques

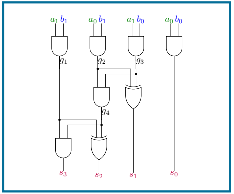
- SAT using CNF encoding
- Binary Moment Diagrams (BMD)
- <u>Algebraic reasoning</u>

Related Work

- **SAT using CNF encoding**
  - A. Biere. **Weakness** of CDCL solvers. SAT Solving Workshop, 2016.
  - P. Beame and V. Liew. **Towards verifying** nonlinear integer arithmetic. In CAV, 2017.
- **Binary moment diagrams**
  - Y.-A. Chen and R.E. Bryant. Verification of arithmetic circuits with **binary moment diagrams**. In DAC, 1995.
- **Algebraic reasoning**
  - O. Wienand, M. Wedler, D. Stoffel, W. Kunz, and G.-M. Greuel. An algebraic approach for proving data correctness in **arithmetic data paths**. In CAV, 2008.
  - J. Lv, P. Kalla, and F. Enescu. Efficient Gröbner basis reductions for formal verification of **Galois field arithmetic circuits**. In IEEE TCAD, 2013.
  - C. Yu, W. Brown, D. Liu, A. Rossi, and M. Ciesielski. Formal verification of arithmetic circuits by **function extraction**. In IEEE TCAD, 2016.
  - A.A.R. Sayed-Ahmed, D. Große, U. Kühne, M. Soeken, and R. Drechsler. Formal verification of integer multipliers by combining **Gröbner basis with logic reduction**. In DATE, 2016.

Basic Idea of Algebraic Approach

Multiplier



Polynomials

$$
\begin{aligned}
B = \{ & \\
& x - a_0 * b_0, \\
& y - a_1 * b_1, \\
& s_0 - x * y, \\
& \dots \\
\} &
\end{aligned}
$$

Specification

$$
\sum_{i=0}^{2n-1} 2^i s_i - \left(\sum_{i=0}^{n-1} 2^i a_i\right)\left(\sum_{i=0}^{n-1} 2^i b_i\right)
$$

Ideal Membership Test

$= 0$ ✓

$\neq 0$ ✗

Polynomials

$$p = c_1\tau_1 + \ldots + c_m\tau_m \quad \in \quad \mathbb{Q}[X] = \mathbb{Q}[x_1, \ldots, x_n]$$

- $\mathbb{Q}[X]$ is the **ring of polynomials** with variables $X = x_1, \ldots, x_n$ and coefficients in $\mathbb{Q}$.
- A **term** τ_i is a product $x_1^{e_1} \cdots x_n^{e_n}$ with $e_j \geq 0$.
- A **monomial** $c_i\tau_i$ is a constant multiple of a term with $c_i \in \mathbb{Q}$.
- A **polynomial** p is a finite sum of monomials.

Polynomials

$$p = c_1\tau_1 + \ldots + c_m\tau_m \quad \in \quad \mathbb{Q}[X] = \mathbb{Q}[x_1, \ldots, x_n]$$

- We fix a **term order** such that for all terms τ, σ_1, σ_2 we have $x_1^0 \cdots x_n^0 = 1 \leq \tau$ and $\sigma_1 \leq \sigma_2 \Rightarrow \tau\sigma_1 \leq \tau\sigma_2$.
- An order is a **lexicographic term order** if for all $\sigma_1 = x_1^{u_1} \cdots x_n^{u_n}$, $\sigma_2 = x_1^{v_1} \cdots x_n^{v_n}$ we have $\sigma_1 < \sigma_2$ iff there exists an index i with $u_j = v_j$ for all $j < i$, and $u_i < v_i$.

- $\mathrm{lm}(p) = c_1\tau_1$ is the **leading monomial** of p.
- $\mathrm{lt}(p) = \tau_1$ is the **leading term** of p.
- $p - \mathrm{lm}(p)$ is the **tail** of p.

Ideals

Ideal. A nonempty subset $I \subseteq \mathbb{Q}[X]$ is called an ideal if

$$\forall\, p, q \in I : p + q \in I \qquad \text{and} \qquad \forall\, p \in \mathbb{Q}[X]\; \forall\, q \in I : pq \in I$$

Basis. A set $P = \{p_1, \ldots, p_m\} \subseteq \mathbb{Q}[X]$ is called a **basis** of an ideal I if

$$I = \{q_1 p_1 + \cdots + q_m p_m \mid q_1, \ldots, q_m \in \mathbb{Q}[X]\} = \langle P \rangle$$

I is the set of polynomials which become zero, when the elements of P become zero.

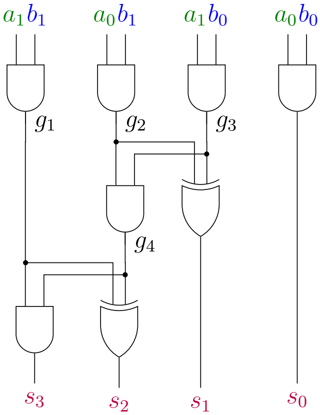
Circuit Polynomials

Gate polynomials.

$$
\begin{array}{ll}
s_3 = g_1 \wedge g_4 & -s_3 + g_1 g_4, \\
s_2 = g_1 \oplus g_4 & -s_2 + g_1 + g_4 - 2g_1 g_4, \\
g_4 = g_2 \wedge g_3 & -g_4 + g_2 g_3, \\
s_1 = g_2 \oplus g_3 & -s_1 + g_2 + g_3 - 2g_2 g_3, \\
g_1 = a_1 \wedge b_1 & -g_1 + a_1 b_1, \\
g_2 = a_0 \wedge b_1 & -g_2 + a_0 b_1, \\
g_3 = a_1 \wedge b_0 & -g_3 + a_1 b_0, \\
s_0 = a_0 \wedge b_0 & -s_0 + a_0 b_0
\end{array}
$$

Input Field polynomials.

$$
\begin{array}{ll}
a_1, a_0 \in \mathbb{B} & a_1(1 - a_1),\ a_0(1 - a_0), \\
b_1, b_0 \in \mathbb{B} & b_1(1 - b_1),\ b_0(1 - b_0)
\end{array}
$$



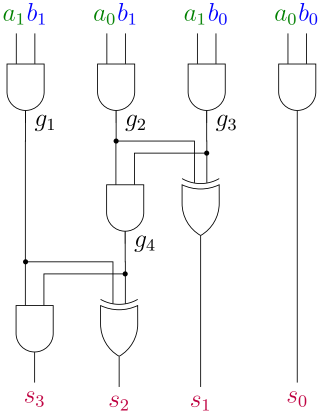
Ideals associated to Circuits

Polynomial Circuit Constraints (PCCs).
A polynomial $p \in \mathbb{Q}[X]$ such that for all

$$(a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1}) \in \{0,1\}^{2n}$$

and resulting values $g_1, \ldots, g_k, s_0, \ldots, s_{2n-1}$ implied by the gates of the circuit C the substitution of these values into p gives zero.

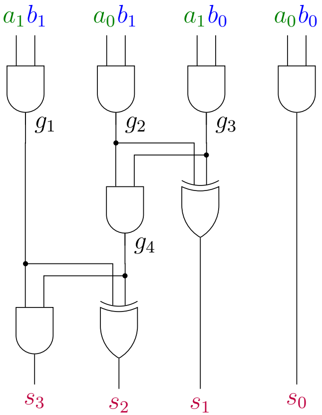
- The set of all PCCs is denoted by $I(C)$.
- $I(C)$ contains all relations of the circuit.
- $I(C)$ is an ideal.



Ideals associated to Circuits

Examples for PCCs:

- $s_0 - a_0 b_0$ and gate
- $a_1^2 - a_1$ a_1 boolean
- $g_2^2 - g_2$ g_2 boolean
- $s_1 g_4$ xor-and constraint
- $\ldots$



Multiplier. A circuit C is called a multiplier if

$$\sum_{i=0}^{2n-1} 2^i s_i - \left(\sum_{i=0}^{n-1} 2^i a_i\right)\left(\sum_{i=0}^{n-1} 2^i b_i\right) \quad \in \quad I(C).$$

Ideal Membership Test

Ideal membership problem. Given a polynomial $f \in \mathbb{Q}[X]$ and an ideal $I = \langle g_1, \ldots, g_m \rangle = \langle G \rangle \subseteq \mathbb{Q}[X]$, determine if $f \in I$.

Given arbitrary basis G of I it is not obvious how to solve ideal membership problem.

Lemma (Ideal membership test)

Let $G = \{g_1, \ldots, g_m\} \subseteq \mathbb{Q}[X]$ be a Gröbner basis, and let $f \in \mathbb{Q}[X]$. Then f is contained in the ideal $I = \langle G \rangle$ iff the unique remainder of f with respect to G is zero.

Gröbner basis

- Every ideal $I \subseteq \mathbb{Q}[X]$ has a **Gröbner basis** w.r.t. a fixed term order.

- Construction algorithm by Buchberger which given an arbitrary basis of an ideal I computes a Gröbner basis of it (doubly exponential)
- Algorithm is based on repeated reduction of so-called S-polynomials (spol).

- A basis G is a Gröbner basis of $I = \langle G \rangle$ if for all $p, q \in G : \mathrm{spol}(p, q)$ reduces to zero.
- **Product criterion.** If $p, q \in \mathbb{Q}[X] \setminus \{0\}$ are such that the leading terms are coprime, i.e., $\mathrm{lcm}(\mathrm{lt}(p), \mathrm{lt}(q)) = \mathrm{lt}(p)\,\mathrm{lt}(q)$, then $\mathrm{spol}(p, q)$ reduces to zero.

Circuit Gröbner basis

We can deduce at least some elements of $I(C)$:

- G = Gate Polynomials + Input Field Polynomials
- Let $J(C) = \langle G \rangle$.
- Lexicographic term order: output variable of a gate is greater than input variables

Theorem

G is a Gröbner basis for $J(C)$.

Proof idea: Application of Buchberger's Product criterion.

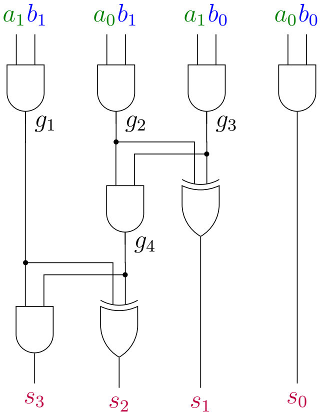
Circuit Polynomials

Gate polynomials.

$$
\begin{aligned}
s_3 &= g_1 \wedge g_4 & -s_3 + g_1 g_4,\\
s_2 &= g_1 \oplus g_4 & -s_2 - 2g_1 g_4 + g_4 + g_1,\\
g_4 &= g_2 \wedge g_3 & -g_4 + g_2 g_3,\\
s_1 &= g_2 \oplus g_3 & -s_1 - 2g_2 g_3 + g_2 + g_3,\\
g_1 &= a_1 \wedge b_1 & -g_1 + a_1 b_1,\\
g_2 &= a_0 \wedge b_1 & -g_2 + a_0 b_1,\\
g_3 &= a_1 \wedge b_0 & -g_3 + a_1 b_0,\\
s_0 &= a_0 \wedge b_0 & -s_0 + a_0 b_0
\end{aligned}
$$

Input Field polynomials.

$$
\begin{aligned}
a_1, a_0 &\in \mathbb{B} & -a_1^2 + a_1,\ -a_0^2 + a_0,\\
b_1, b_0 &\in \mathbb{B} & -b_1^2 + b_1,\ -b_0^2 + b_0
\end{aligned}
$$



Soundness and completeness

Theorem (Soundness and completeness)

For all acyclic circuits C, we have $J(C) = I(C)$.

- $J(C) \subset I(C)$: soundness
- $I(C) \subset J(C)$: completeness

Non-Incremental Checking Algorithm

Non-Incremental Checking Algorithm.

Divide polynomial $\sum_{i=0}^{2n-1} 2^i s_i - (\sum_{i=0}^{n-1} 2^i a_i)(\sum_{i=0}^{n-1} 2^i b_i)$ by elements of G until no further reduction is possible, then C is a multiplier iff remainder is zero.

Implications:

- Leading term is one variable, division is actually substitution by tail.
- Leading coefficient ± 1 of all gate polynomials, computation stays in $\mathbb{Z}$.
- Still can use rational coefficients $\mathbb{Q}$ (important for Singular).
- Completeness proof allows to derive input assignment if C is incorrect.

Example: 2 Bit - Binary Multiplication

$$
\begin{aligned}
G = \{ & \\
& -s_3 + g_1 g_4, \\
& -s_2 + g_1 + g_4 - 2g_1 g_4, \\
& -g_4 + g_2 g_3, \\
& -s_1 + g_2 + g_3 - 2g_2 g_3, \\
& -g_1 + a_1 b_1, \\
& -g_2 + a_0 b_1, \\
& -g_3 + a_1 b_0, \\
& -s_0 + a_0 b_0, \\
& -a_1^2 + a_1, \\
& -a_0^2 + a_0, \\
& -b_1^2 + b_1, \\
& -b_0^2 + b_0 \}
\end{aligned}
$$

$$8s_3 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0$$

Example: 2 Bit - Binary Multiplication

$$
\begin{aligned}
G = \{ & \\
& -s_3 + g_1 g_4, \\
& -s_2 + g_1 + g_4 - 2g_1 g_4, \\
& -g_4 + g_2 g_3, \\
& -s_1 + g_2 + g_3 - 2g_2 g_3, \\
& -g_1 + a_1 b_1, \\
& -g_2 + a_0 b_1, \\
& -g_3 + a_1 b_0, \\
& -s_0 + a_0 b_0, \\
& -a_1^2 + a_1, \\
& -a_0^2 + a_0, \\
& -b_1^2 + b_1, \\
& -b_0^2 + b_0 \}
\end{aligned}
$$

$$8s_3 + 4s_2 + 2s_1 + s_0 - 4a_1b_1 - 2a_1b_0 - 2a_0b_1 - a_0b_0$$

$$8g_1g_4 + 4s_2 + 2s_1 + s_0 - 4a_1b_1 - 2a_1b_0 - 2a_0b_1 - a_0b_0$$

Example: 2 Bit - Binary Multiplication

$$
\begin{aligned}
G = \{ & \\
& -s_3 + g_1 g_4, \\
& -s_2 + g_1 + g_4 - 2 g_1 g_4, \\
& -g_4 + g_2 g_3, \\
& -s_1 + g_2 + g_3 - 2 g_2 g_3, \\
& -g_1 + a_1 b_1, \\
& -g_2 + a_0 b_1, \\
& -g_3 + a_1 b_0, \\
& -s_0 + a_0 b_0, \\
& -a_1^2 + a_1, \\
& -a_0^2 + a_0, \\
& -b_1^2 + b_1, \\
& -b_0^2 + b_0 \}
\end{aligned}
$$

$$8s_3 + 4s_2 + 2s_1 + s_0 - 4a_1b_1 - 2a_1b_0 - 2a_0b_1 - a_0b_0$$

$$8g_1g_4 + 4s_2 + 2s_1 + s_0 - 4a_1b_1 - 2a_1b_0 - 2a_0b_1 - a_0b_0$$

$$8g_1g_4 + 4(g_1 + g_4 - 2g_1g_4) + 2s_1 + s_0$$
$$-4a_1b_1 - 2a_1b_0 - 2a_0b_1 - a_0b_0$$

Example: 2 Bit - Binary Multiplication

$$
\begin{aligned}
G = \{ & \\
& -s_3 + g_1 g_4, \\
& -s_2 + g_1 + g_4 - 2g_1 g_4, \\
& -g_4 + g_2 g_3, \\
& -s_1 + g_2 + g_3 - 2g_2 g_3, \\
& -g_1 + a_1 b_1, \\
& -g_2 + a_0 b_1, \\
& -g_3 + a_1 b_0, \\
& -s_0 + a_0 b_0, \\
& -a_1^2 + a_1, \\
& -a_0^2 + a_0, \\
& -b_1^2 + b_1, \\
& -b_0^2 + b_0 \}
\end{aligned}
$$

$$
\begin{aligned}
8s_3 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
8g_1 g_4 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
8g_1 g_4 + 4(g_1 + g_4 - 2g_1 g_4) + 2s_1 + s_0 \\
-4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
\vdots \\
\color{red}{0}
\end{aligned}
$$

Computation Issues

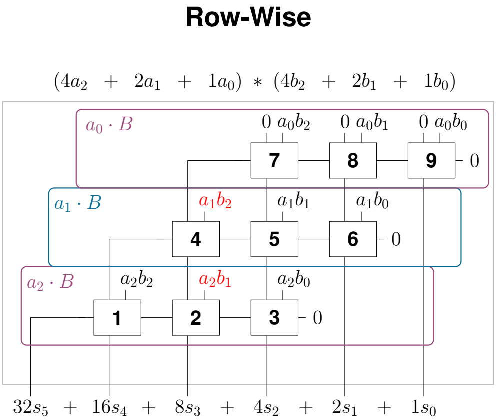
Generally the number of monomials in the intermediate results increases drastically:

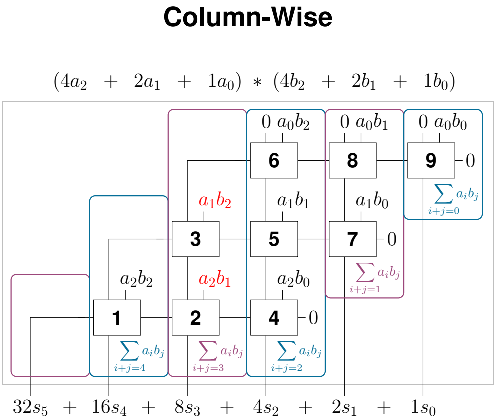
- 8-bit multiplier can not be verified within 20 minutes.

Tailored heuristics become very important:

- Choose appropriate term order.
- Divide verification problem into smaller sub-problems.
- Rewrite and thus simplify Gröbner basis G.

Order


Row-Wise
$(4a_2 + 2a_1 + 1a_0) * (4b_2 + 2b_1 + 1b_0)$
$a_0 \cdot B$
$a_1 \cdot B$
$a_2 \cdot B$
0 a_0b_2
0 a_0b_1
0 a_0b_0
7
8
9
0
a_1b_2
a_1b_1
a_1b_0
4
5
6
0
a_2b_2
a_2b_1
a_2b_0
1
2
3
0
$32s_5 + 16s_4 + 8s_3 + 4s_2 + 2s_1 + 1s_0$


Column-Wise
$(4a_2 + 2a_1 + 1a_0) * (4b_2 + 2b_1 + 1b_0)$
0 a_0b_2
0 a_0b_1
0 a_0b_0
6
8
9
0
$\sum_{i+j=0} a_ib_j$
a_1b_2
a_1b_1
a_1b_0
3
5
7
0
$\sum_{i+j=1} a_ib_j$
a_2b_2
a_2b_1
a_2b_0
1
2
4
0
$\sum_{i+j=4} a_ib_j$
$\sum_{i+j=3} a_ib_j$
$\sum_{i+j=2} a_ib_j$
$32s_5 + 16s_4 + 8s_3 + 4s_2 + 2s_1 + 1s_0$

Slicing

Partial Products. Let $P_k = \sum_{k=i+j} a_i b_j$.

Input Cone. For each output bit s_i we determine its input cone

$$I_i = \{\text{gate } g \mid g \text{ is in input cone of output } s_i\}$$

Slice. Slices S_i are defined as the difference of consecutive cones I_i:

$$S_0 = I_0 \qquad S_{i+1} = I_{i+1} \setminus \bigcup_{j=0}^{i} S_j$$

Sliced Gröbner Bases. Let G_i be the set of gate and input field polynomials in S_i.

Carry Recurrence Relation

Carry Recurrence Relation.
A sequence of $2n+1$ polynomials $C_0, \dots, C_{2n}$ is called a **carry sequence** if

$$-C_i + 2C_{i+1} + s_i - P_i \in I(C) \quad \text{for all } 0 \leq i < 2n+1.$$

Then $R_i = -C_i + 2C_{i+1} + s_i - P_i$ are the **carry recurrence relations** for $C_0, \dots, C_{2n}$.

Theorem

Let C be a circuit where all carry recurrence relations are contained in $I(C)$.
Then C is a multiplier, iff $C_0 - 2^{2n}C_{2n} \in I(C)$.

Incremental Algorithm

Incremental Checking Algorithm.

input: Circuit C with sliced Gröbner bases G_i
output: Determine whether C is a multiplier

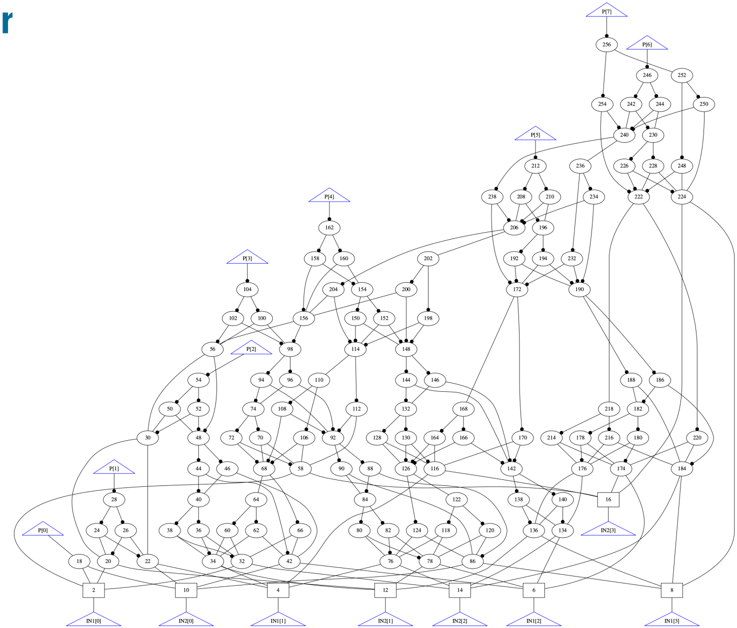
$C_{2n} \leftarrow 0$

for $i \leftarrow 2n - 1$ **to** 0

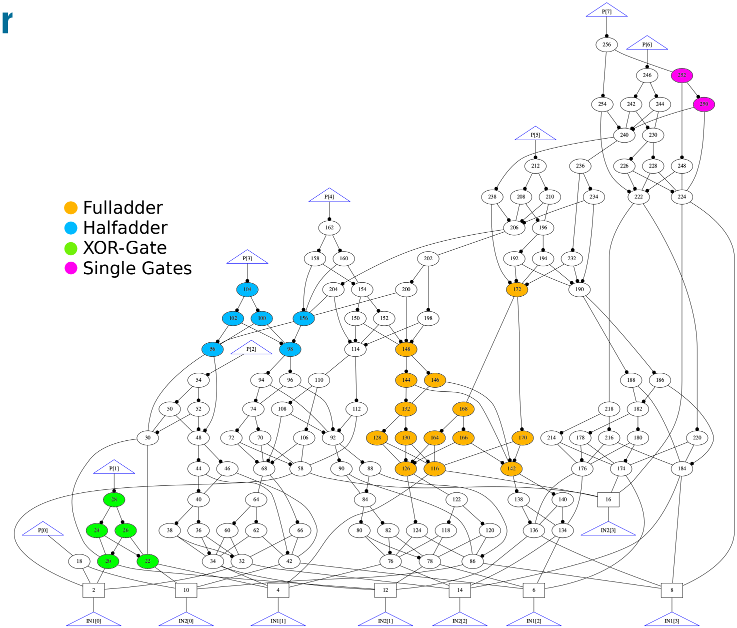
$C_i \leftarrow$ Remainder $(\, 2C_{i+1} + s_i - P_i, \;\; G_i \,)$

return $C_0 = 0$

Multiplier


P[7]
256 P[6]
246 252
254 242 244 250
P[5] 240 230
212 236 226 228 248
P[4] 238 208 210 234 222 224
162 206 196
P[3] 158 160 202 192 194 232
104 204 154 200 172 190
102 100 156 150 152 198
56 P[2] 98 114 148
54 94 96 110 144 146 188 186
50 52 74 108 112 132 168 218 182
30 48 72 70 106 92 128 130 164 166 170 214 178 216 180 220
P[1] 44 46 68 58 90 88 126 116 142 176 174 184
28 40 64 84 122 138 140 16
P[0] 24 26 38 36 60 62 66 80 82 124 118 120 136 134 IN2[3]
18 20 22 34 32 42 76 78 86
2 10 4 12 14 6 8
IN1[0] IN2[0] IN1[1] IN2[1] IN2[2] IN1[2] IN1[3]

Multiplier


Fulladder
Halfadder
XOR-Gate
Single Gates
P[7]
P[6]
P[5]
P[4]
P[3]
P[2]
P[1]
P[0]
IN1[0]
IN2[0]
IN1[1]
IN2[1]
IN2[2]
IN1[2]
IN2[3]
IN1[3]

Variable Elimination

Identify sub-circuits C_S in the AIG and eliminate internal variables:

- Full-adder rewriting
- Half-adder rewriting
- XOR- Rewriting
- Common-Rewriting

Variable elimination is based on **elimination theory of Gröbner bases.**

Elimination theory of Gröbner bases

Elimination order. Let $X = Y \dot{\cup} Z$ and we want to eliminate Z. Order the terms such that for all terms σ, τ where a variable from Z is contained in σ but not in τ, we obtain $\tau < \sigma$.

Elimination ideal. The elimination ideal J where the Z-variables are eliminated of $I \subseteq \mathbb{Q}[X] = \mathbb{Q}[Y, Z]$ is defined by

$$J = I \cap \mathbb{Q}[Y].$$

Elimination theorem. Given an ideal $I \subseteq \mathbb{Q}[X] = \mathbb{Q}[Y, Z]$. Further let G be a Gröbner basis of I with respect to an elimination order $Y < Z$. Then the set

$$H = G \cap \mathbb{Q}[Y]$$

is a Gröbner basis of the elimination ideal $J = I \cap \mathbb{Q}[Y]$, in particular $\langle H \rangle = J$.

Elimination procedure

Problem: Computing a Gröbner basis H for $I(C)$ w.r.t an elimination order is costly.

Solution: Split G into two parts.

$$\begin{tikzcd}
 & G \arrow[dl] \arrow[dr] & & \\
G_A \arrow[dddr] & & G_B \arrow[d] & \\
 & & H_B \arrow[dl] \arrow[dr] & \\
 & H_Y \arrow[d] & & H_Z \\
 & H & &
\end{tikzcd}$$

Step 1: original Gröbner basis G

Step 2: split G into two subbases

Step 3: change order of $<_G$ to $<_H$

Step 4: eliminate the variables of Z

Step 5: rejoin bases $H = G_A \cup H_Y$

Elimination procedure

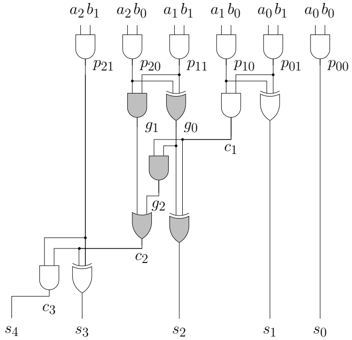
Theorem

Let $G \subseteq \mathbb{Q}[X] = \mathbb{Q}[Y, Z]$ be a Gröbner basis with respect to some term order $<_G$. Let $G_A = G \cap \mathbb{Q}[Y]$ and $G_B = G \setminus G_A$. Let $<_H$ be an elimination order for Z which agrees with $<_G$ for all terms that are free of Z, i.e., terms free of Z are equally ordered in $<_G$ and $<_H$. Suppose that $\langle G_B \rangle$ has a Gröbner basis H_B with respect to $<_H$ which is such that every leading term in H_B is free of Z or free of Y. Then $\langle G \rangle \cap \mathbb{Q}[Y] = (\langle G_A \rangle + \langle G_B \rangle) \cap \mathbb{Q}[Y] = \langle G_A \rangle + (\langle G_B \rangle \cap \mathbb{Q}[Y])$.

Theorem

Let $G, G_A, G_B, H_B, H_Y, H_Z, <_H, <_G$ be as before. Then $H = G_A \cup H_Y$ is a Gröbner basis w.r.t. the ordering $<_H$.

Example: Full-Adder Rewriting



$$
\begin{aligned}
G_A &= G \backslash G_B \\
G_B &= \{ -g_0 + p_{20} + p_{11} - 2p_{20}p_{11}, \quad -g_1 + p_{20}p_{11}, \quad -g_2 + c_1 g_0, \\
&\qquad -s_2 + c_1 + g_0 - 2c_1 g_0, \quad -c_2 + g_1 + g_2 - g_1 g_2 \}
\end{aligned}
$$

Original lexicographic term ordering $<_G$:

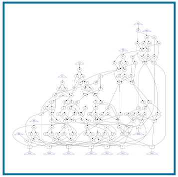
$$
\begin{aligned}
& b_0 < b_1 < a_0 < a_1 < a_2 < p_{00} < s_0 < p_{01} < p_{10} < s_1 < c_1 < \\
& p_{11} < p_{20} < \color{blue}{g_0} < \color{blue}{g_1} < \color{blue}{g_2} < s_2 < c_2 < p_{21} < s_3 < c_3 < s_4
\end{aligned}
$$

Gröbner basis H_B w.r.t. elimination order $<_H$:

$$
\begin{aligned}
H_B = \{ & \color{blue}{g_0 + 2p_{20}p_{11} - p_{20} - p_{11}}, \quad \color{blue}{g_1 - p_{20}p_{11}}, \\
& \color{blue}{g_2 + 2p_{20}p_{11}c_1 - p_{20}c_1 - p_{11}c_1}, \\
& s_2 - 4p_{20}p_{11}c_1 + 2p_{20}p_{11} + 2p_{20}c_1 - p_{20} + 2p_{11}c_1 - p_{11} - c_1, \\
& 2c_2 + s_2 - p_{20} - p_{11} - c_1 \}
\end{aligned}
$$

Experiments

Multiplier



AIG

AIGMULTOPOLY

Polynomials

$$B = \{ \\ x - a_0 * b_0, \\ y - a_1 * b_1, \\ s_0 - x * y, \\ \dots \\ \}$$

CAS-File

MATHEMATICA

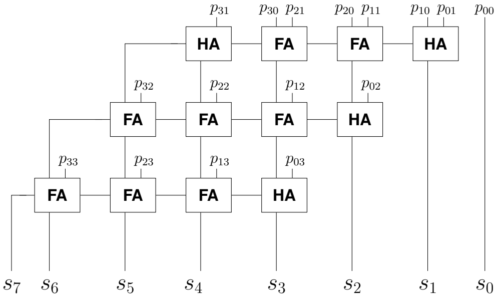
SINGULAR

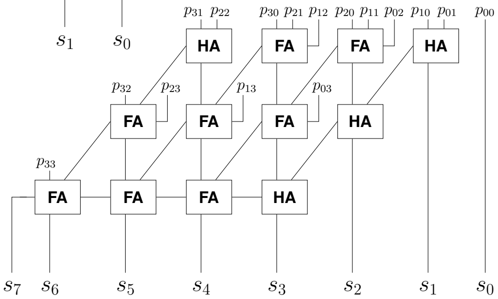
Ideal Membership

$C_0 = 0$ ✓

$C_0 \neq 0$ ✗

Experiments





Experiments

mult	n	Mathematica				Singular			
		non-inc	incremental			non-inc	incremental		
				+xor	+xor +add			+xor	+xor +add
btor	16	3	5	2	1	1	1	1	1
btor	32	56	31	14	2	42	28	10	1
btor	64	MO	292	131	11	MO	MO	MO	14
btor	128	TO	TO	TO	101	EE	EE	EE	EE
sp-ar-rc	16	9	7	4	1	TO	6	1	0
sp-ar-rc	32	326	171	30	2	TO	242	28	2
sp-ar-rc	64	MO	TO	300	11	MO	EE	MO	16
sp-ar-rc	128	TO	TO	TO	102	EE	EE	EE	EE

Table: time in sec; TO = 1200 sec, MO = 14GB, EE=more than 32767 variables

Current Work - Generating Proofs

- Polynomial calculus as frame-work
- Define a more practical calculus
- Generate and certify low-level algebraic proofs

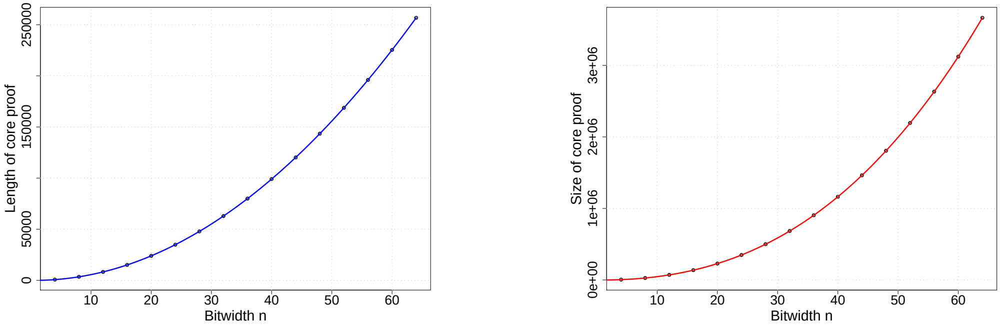


Figure: Length and size of btor-btor commutativity check

Future Work

Circuit Verification

- other word-level operators (shift, division, . . .)
- more complex multipliers
- negative numbers

Proof Generation

- connection to clausal proof systems
- certified proof checker
- boolean proofs

ON THE PROBLEM OF ARITHMETIC CIRCUIT VERIFICATION USING COMPUTER ALGEBRA

Daniela Ritirc joint work with Armin Biere and Manuel Kauers
Johannes Kepler University
Linz, Austria

Theory Reading Group Meeting
April 23, 2018
Stockholm, Sweden

JOHANNES KEPLER UNIVERSITY LINZ

Rigorous Systems Engineering