FORMALE VERIFIKATION VON MULTIPLIZIERERN MIT HILFE VON COMPUTERALGEBRA

Daniela Kaufmann

✉ daniela.kaufmann@jku.at

Johannes Kepler Universität

Linz, Österreich

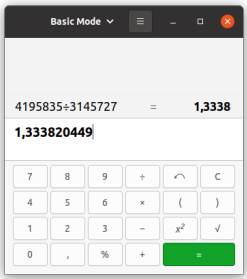
Kolloquium GI Dissertationspreis

16. Juni 2021

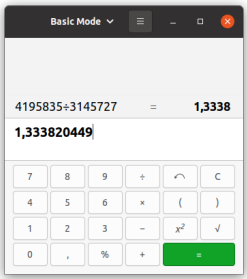
JOHANNES KEPLER
UNIVERSITY LINZ

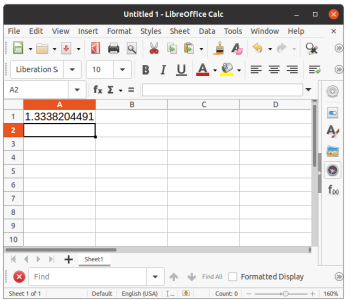
$$4\,195\,835 \div 3\,145\,727 = ?$$

$$4\,195\,835 \div 3\,145\,727 = ?$$

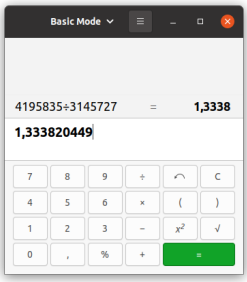


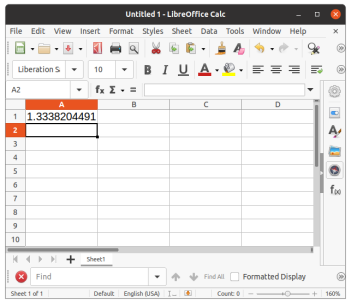
$$4\,195\,835 \div 3\,145\,727 = ?$$

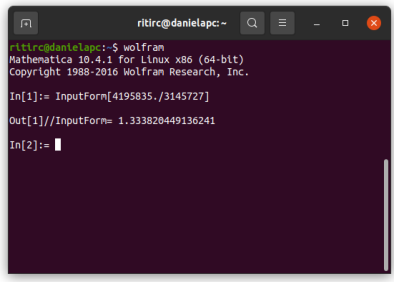


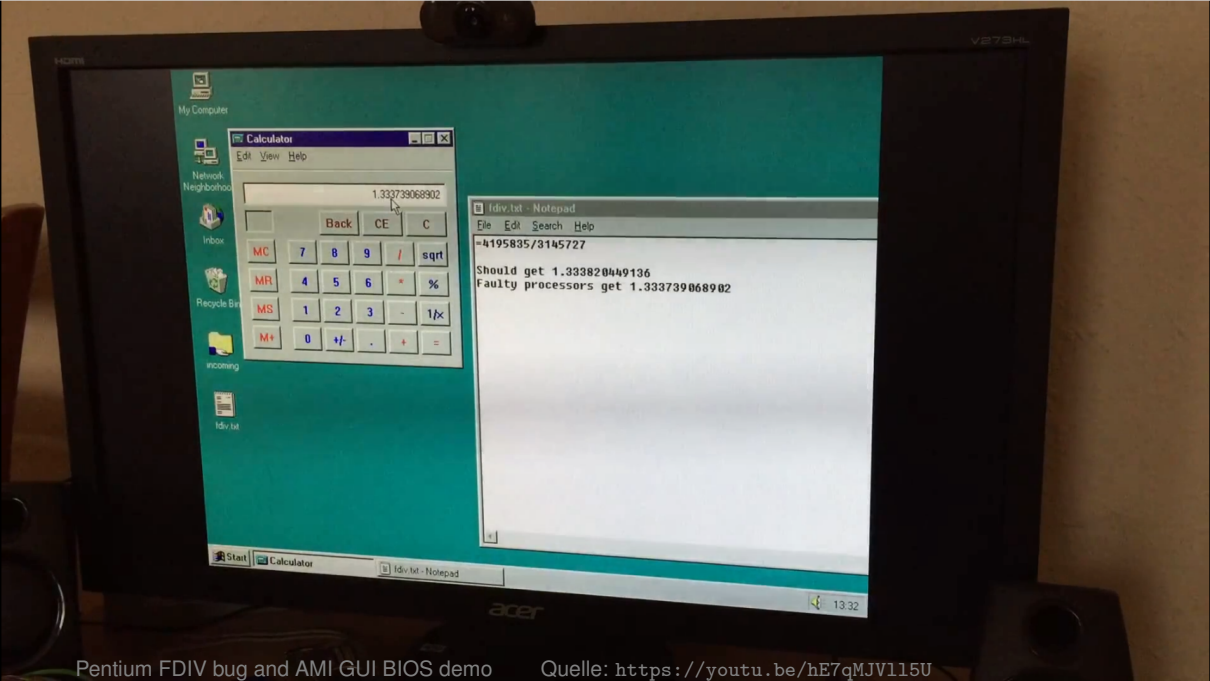


$$4\,195\,835 \div 3\,145\,727 = ?$$









Pentium FDIV bug and AMI GUI BIOS demo Quelle: https://youtu.be/hE7qMJV115U

Intel Pentium FDIV-Bug 1994

Quelle: http://neology.com.au/portfolios/a80502-90-sx923/

- Division von bestimmten Zahlen liefert falsches Ergebnis
- Schaden: 500 Millionen US-Dollar

Korrektheitsbeweise sind noch immer nicht vollautomatisiert möglich

Herausforderung: Multiplizierer

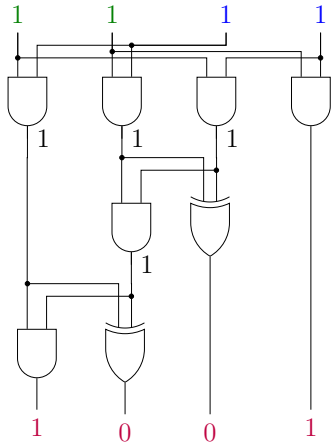
Ganzzahlige Multiplikation

$$
\begin{array}{cccc}
1 & 1 & \cdot & 1 \quad 1 \\
\hline
 & & & 1 \quad 1 \\
 & & 1 & 1 \\
 & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 0} \\
\hline
 & 1 & 0 & 0 \quad 1
\end{array}
$$

$3 \cdot 3 = 9$

Ganzzahlige Multiplikation

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



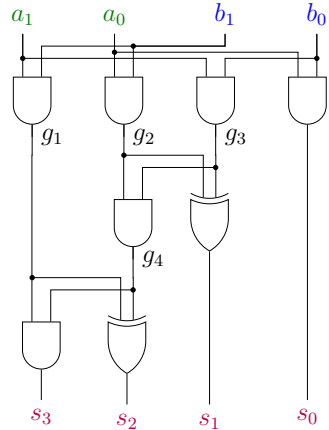
$3 \cdot 3 = 9$

Multiplizierer

Ausgangslage:
Ganzzahlige Multiplizierer mit fixer Bitbreite n

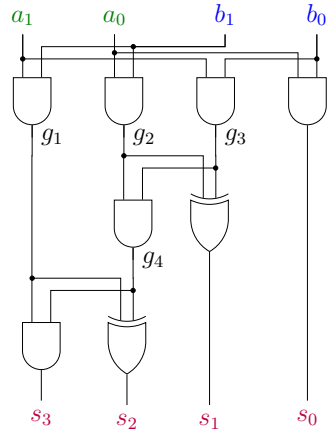
Problem:
Eingänge $a_i, b_i \in \mathbb{B}$, Ausgänge $s_i \in \mathbb{B}$:

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



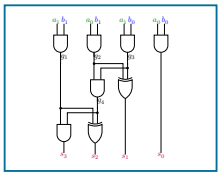
Simulation

Bitbreite	Testfälle
2	16
3	64
4	256
5	1 024
6	4 096
7	16 384
8	65 536
⋮	⋮
32	18 446 744 073 709 551 616



Formale Verifikation

System



Spezifikation

Für alle $a_i, b_i \in \mathbb{B}$:

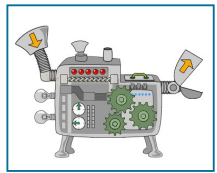
$(2a_1 + a_0) * (2b_1 + b_0) =$

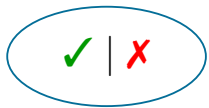
$8s_3 + 4s_2 + 2s_1 + s_0$?

Mathematisches Modell

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

Automatisierter Entscheidungsprozess





Frühere Ansätze

Erfüllbarkeitsproblem der Aussagenlogik (SAT)

- SAT 2016 Competition [Bi16]
- Exponentielle Laufzeit der Solver

Binäre Entscheidungsdiagramme

- Erste Technik mit der der FDIV-Bug gefunden werden konnte [CB95]
- Benötigt Kenntnis des Schaltungsaufbaus

Theorembeweiser

- Industrielle Praxis
- Benötigt Kenntnis der Problemdomäne

Frühere Ansätze

Erfüllbarkeitsproblem der Aussagenlogik (SAT)

- SAT 2016 Competition [Bi16]
- Exponentielle Laufzeit der Solver

Theorembeweiser

- Industrielle Praxis
- Benötigt Kenntnis der Problemdomäne

Binäre Entscheidungsdiagramme

- Erste Technik mit der der FDIV-Bug gefunden werden konnte [CB95]
- Benötigt Kenntnis des Schaltungsaufbaus

Computeralgebra

- Durchbruch: [Yu16] [Sa16]
- Polynomkodierung
- Vollautomatisiert einsetzbar

Beitrag meiner Dissertation

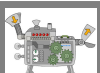
Präzise mathematische Formulierung

- Korrektheit
- Vollständigkeit

Stand der Technik für Verifikation von Multiplizierern:

- Inkrementeller Verifikationsalgorithmus **(Best Paper Award)**
- Elimination von Variablen
- Kombination von logischem und algebraischem Rechnen

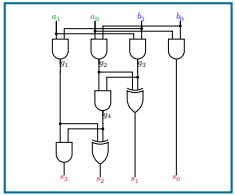
Einzigartig: unabhängige Zertifikate für die Verifikation



Open-Source Tool AMULET

Grundidee Algebraischer Verifikation

Multiplizierer



Polynome

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

Spezifikation

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

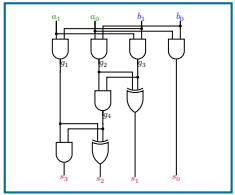
Implikation

$= 0$ ✓

$\neq 0$ ✗

Grundidee Algebraischer Verifikation

Multiplizierer



Polynome

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

Spezifikation

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

Idealzugehörigkeit

$= 0$ ✓

$\neq 0$ ✗

Polynomkodierung

Gatterpolynome $G(C)$

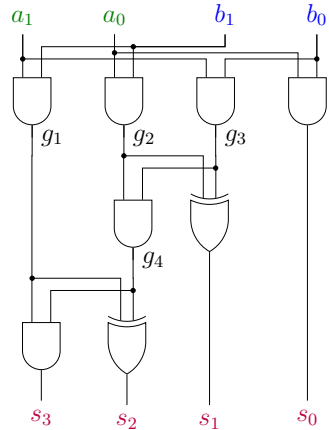
$$
\begin{array}{ll}
s_3 = g_1 \wedge g_4 & -s_3 + g_1 g_4, \\
s_2 = g_1 \oplus g_4 & -s_2 - 2g_1 g_4 + g_4 + g_1, \\
g_4 = g_2 \wedge g_3 & -g_4 + g_2 g_3, \\
s_1 = g_2 \oplus g_3 & -s_1 - 2g_2 g_3 + g_2 + g_3, \\
g_1 = a_1 \wedge b_1 & -g_1 + a_1 b_1, \\
g_2 = a_0 \wedge b_1 & -g_2 + a_0 b_1, \\
g_3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

Boolesche Eingangsbedingungen $B(C)$

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

Spezifikation $\mathcal{S}_n$

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



Präzise Formalisierung

- Idealzugehörigkeit lässt sich eindeutig mit Hilfe von Gröbnerbasen bestimmen.
- Im Allgemeinen ist die Berechnung von Gröbnerbasen aufwändig.

Theorem

Berechnung einer Gröbnerbasis nicht notwendig. $G(C) \cup B(C)$ ist eine Gröbnerbasis.

Reduktionsalgorithmus

Division der Spezifikation $\sum_{i=0}^{2n-1} 2^i s_i - \left(\sum_{i=0}^{n-1} 2^i a_i\right)\left(\sum_{i=0}^{n-1} 2^i b_i\right)$ durch $G(C) \cup B(C)$

bis keine weitere Reduktion mehr durchführbar ist. Die Schaltung ist genau dann ein Multiplizierer, wenn das finale Resultat null ist.

Korrektheit und Vollständigkeit

[RBK17]

Obwohl algebraische Verifikation bereits genutzt wurde, wurden die Korrektheit und Vollständigkeit dieser Methode bisher nicht bewiesen.

Theorem

Verifikationsmethode ist erfolgreich $\Leftrightarrow$ Schaltung ist ein korrekter Multiplizierer

Exzerpt:

Sei $I(C) = \{p \in \mathbb{Q}[X] \mid p(X) = 0$ für alle $a_0, \ldots, a_{n-1}, b_0, \ldots, b_{n-1} \in \mathbb{B}\}$.
Sei $J(C) = \langle -s_3 + g_1 g_4, -s_2 - 2g_1 g_4 + g_4 + g_1, \ldots \rangle$.

Eine Schaltung C ist ein Multiplizierer wenn $\sum_{i=0}^{2n-1} 2^i s_i - \left(\sum_{i=0}^{n-1} 2^i a_i\right)\left(\sum_{i=0}^{n-1} 2^i b_i\right) \in I(C)$.

Korrektheit und Vollständigkeit: Für alle azyklischen Schaltungen C gilt $J(C) = I(C)$.

Verifikation

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

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

Verifikation

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

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

$$8g_1 g_4 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0$$

Verifikation

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

$$
\begin{aligned}
8s_3 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
8g_1 g_4 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
4g_4 + 4g_1 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0
\end{aligned}
$$

Verifikation

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

$$
\begin{aligned}
8s_3 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
8g_1 g_4 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
4g_4 + 4g_1 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
4g_2 g_3 + 4g_1 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0
\end{aligned}
$$

Verifikation

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

$$
\begin{aligned}
8s_3 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
8g_1 g_4 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
4g_4 + 4g_1 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
4g_2 g_3 + 4g_1 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
4g_1 + 2g_3 + 2g_2 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0
\end{aligned}
$$

Verifikation

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

$$
\begin{aligned}
8s_3 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
8g_1 g_4 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
4g_4 + 4g_1 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
4g_2 g_3 + 4g_1 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
4g_1 + 2g_3 + 2g_2 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
2g_3 + 2g_2 + s_0 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0
\end{aligned}
$$

Verifikation

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

$$
\begin{aligned}
8s_3 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
8g_1 g_4 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
4g_4 + 4g_1 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
4g_2 g_3 + 4g_1 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
4g_1 + 2g_3 + 2g_2 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
2g_3 + 2g_2 + s_0 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
2g_3 + s_0 - 2a_1 b_0 - a_0 b_0
\end{aligned}
$$

Verifikation

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

$$
\begin{aligned}
8s_3 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
8g_1 g_4 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
4g_4 + 4g_1 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
4g_2 g_3 + 4g_1 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
4g_1 + 2g_3 + 2g_2 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
2g_3 + 2g_2 + s_0 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
2g_3 + s_0 - 2a_1 b_0 - a_0 b_0 \\
s_0 - a_0 b_0
\end{aligned}
$$

Verifikation

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

$$
\begin{aligned}
8s_3 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
8g_1 g_4 + 4s_2 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
4g_4 + 4g_1 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
4g_2 g_3 + 4g_1 + 2s_1 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
4g_1 + 2g_3 + 2g_2 + s_0 - 4a_1 b_1 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
2g_3 + 2g_2 + s_0 - 2a_1 b_0 - 2a_0 b_1 - a_0 b_0 \\
2g_3 + s_0 - 2a_1 b_0 - a_0 b_0 \\
s_0 - a_0 b_0 \\
0
\end{aligned}
$$

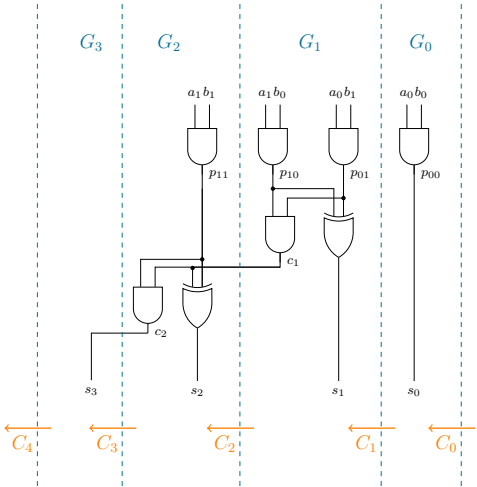
Praktische Probleme

- Algorithmus terminiert für Muliplizierer mit Bitbreite 8 und 708 Gattern nicht mehr:

Schritte	Polynomgröße	Zeit (s)
0	80	0
10	95	0
20	151	0
30	1 015	1
40	114 823	21
41	229 495	360
42	458 887	368
43	–	$> 1\,200$

Inkrementelle Verifikation

[RBK17, KBK20b]



Sei $P_k = \sum_{k=i+j} a_i b_j$.

Spalten-basierte Verifikation

Eingabe: Schaltung C mit geteilten Gröbnerbasen G_i
Ausgabe: Entscheidung ob C ein Multiplizierer ist

$C_{2n} \leftarrow 0$

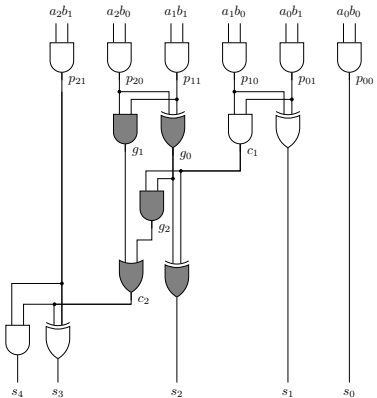
for $i \leftarrow 2n-1$ **to** 0

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

return $C_0 = 0$

Elimination von Variablen

[KBK19, KBK20b]



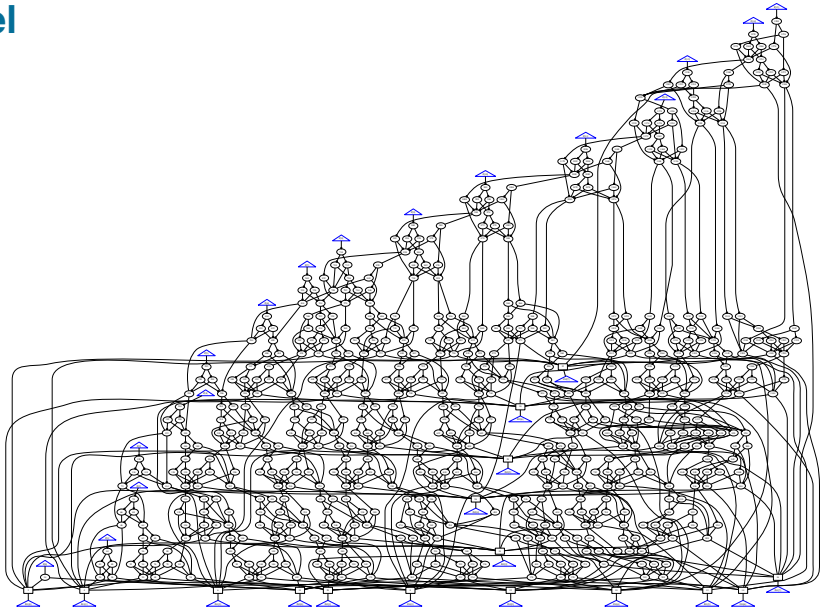
Volladdierer

$$\left.\begin{aligned} &-c_2 + g_2 + g_1 - g_2 g_1, \\ &-s_2 + g_0 + c_1 - 2g_0 c_1, \\ &-g_2 + g_0 c_1, \\ &-g_1 + p_{20} p_{11}, \\ &-g_0 + p_{20} + p_{11} - 2p_{20} p_{11} \end{aligned}\right\} -2c_2 - s_2 + p_{20} + p_{11} + c_1$$

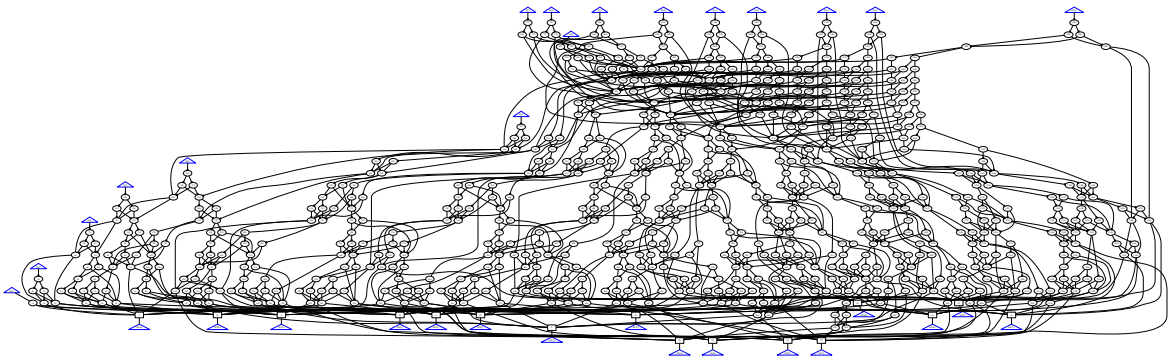
Theorem

Lokale Elimination erhält Gröbnerbaseneigenschaft.

Simpel

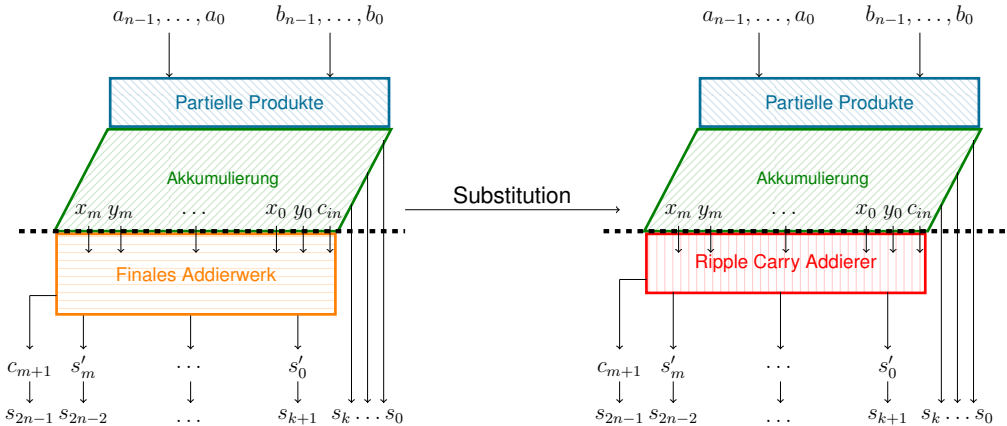


Hart



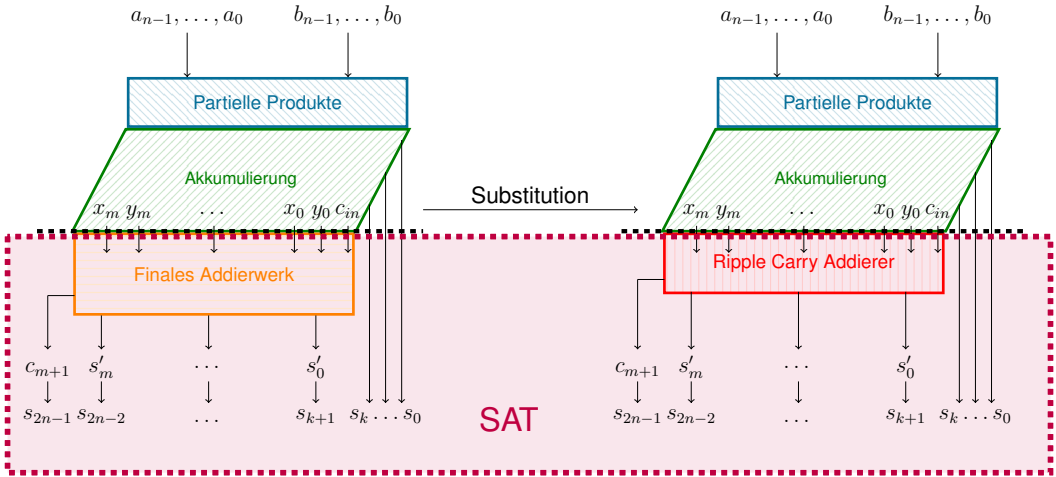
SAT & Computer Algebra

[KBK19]



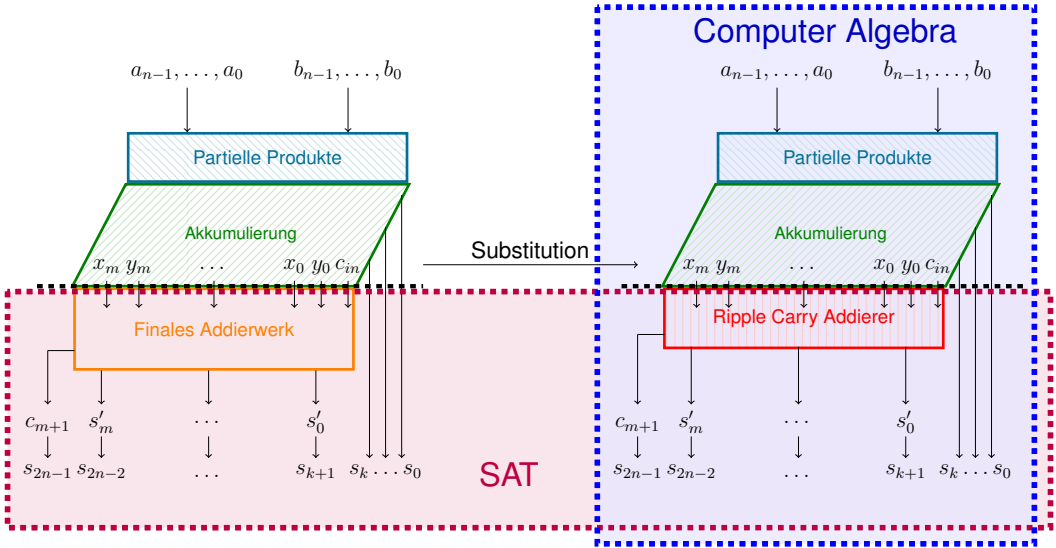
SAT & Computer Algebra

[KBK19]



SAT & Computer Algebra

[KBK19]



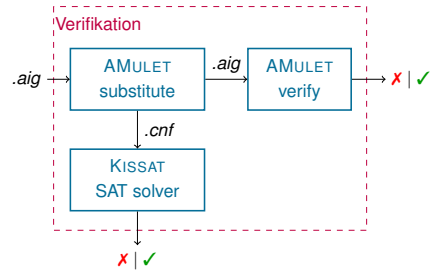
AMulet

Computer Algebra Systeme [KBK20b]

- Mathematica, Singular
- Umfassende Werkzeuge
  (Mathematica hat $> 5\,000$ Funktionen)
- Allgemeiner Anwendungsbereich

AMULET [KBK19]

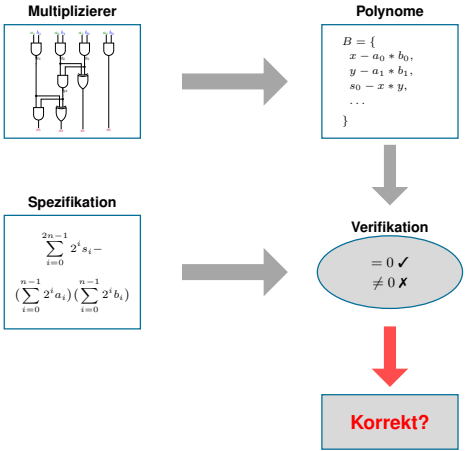
- Open-Source (MIT)
- Substitution und Verifikation sind vollautomatisiert
- Polynomiale Algorithmen sind für Verifikation von Multiplizierern optimiert



`https://github.com/d-kfmnn/amulet`

Zertifikate

[RBK18, KFB20]

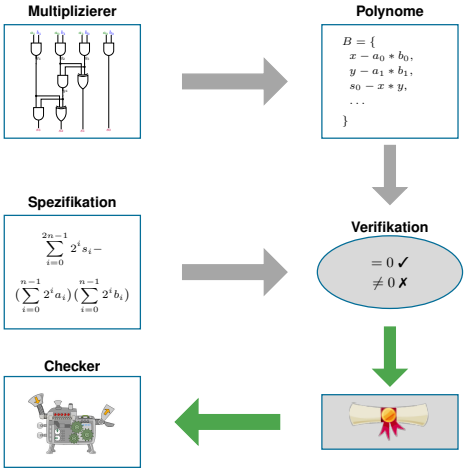


Problem:

- Kann man dem Verifikationstool vertrauen?
- Ist der Verifikationsprozess korrekt?

Zertifikate

[RBK18, KFB20]



Problem:

- Kann man dem Verifikationstool vertrauen?
- Ist der Verifikationsprozess korrekt?

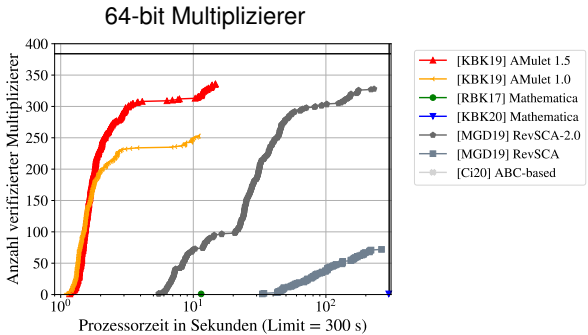
Kalkül:

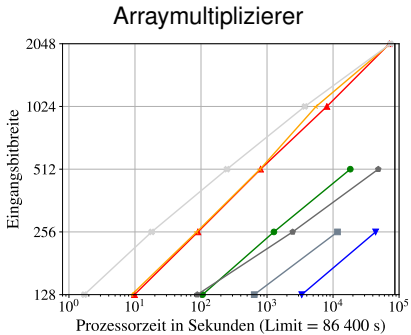
```
* : -b+1-a,        a,          -a*b+a-a^2;
* : -a^2+a,        -1,         a^2-a;
+ : -a*b+a-a^2,    a^2-a,      -a*b;
+ : -a*b,          -c+a*b,     -c;
* : -c,            -1,         c;
```

Checker:

PACHECK `https://github.com/d-kfmnn/pacheck`

Evaluierung





Beitrag meiner Dissertation

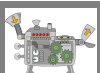
Präzise mathematische Formulierung

- Korrektheit
- Vollständigkeit

Stand der Technik für Verifikation von Multiplizierern:

- Inkrementeller Verifikationsalgorithmus **(Best Paper Award)**
- Elimination von Variablen
- Kombination von logischem und algebraischem Rechnen

Einzigartig: unabhängige Zertifikate für die Verifikation



Open-Source Tool AMULET

FORMALE VERIFIKATION VON MULTIPLIZIERERN MIT HILFE VON COMPUTERALGEBRA

Daniela Kaufmann

✉ daniela.kaufmann@jku.at

Johannes Kepler Universität

Linz, Österreich

Kolloquium GI Dissertationspreis

16. Juni 2021

JOHANNES KEPLER
UNIVERSITY LINZ

Referenzen I

[Bi16] A. Biere. Collection of Combinational Arithmetic Miters Submitted to the SAT Competition 2016. In SAT Competition 2016, pages 65–66, Dep. of Computer Science Report Series B, University of Helsinki, 2016.

[CB95] Y. Chen and R. Bryant. Verification of Arithmetic Circuits with Binary Moment Diagrams. In Proc. of DAC'95, pages 535–541, ACM, 1995.

[Ci20] M. J. Ciesielski, T. Su, A. Yasin and C. Yu. Understanding Algebraic Rewriting for Arithmetic Circuit Verification: a Bit-Flow Model. In IEEE TCAD, vol. 39, no. 6, pages 1346–1357, 2020.

[KBK19] D. Kaufmann, A. Biere and M. Kauers. Verifying Large Multiplizierers by Combining SAT and Computer Algebra. In Proc. of FMCAD'19, pages 28–36, IEEE, 2019.

[KBK20b] D. Kaufmann, A. Biere and M. Kauers. Incremental Column-Wise Verification of Arithmetic Circuits Using Computer Algebra. In FMSD, vol 56, pages 22–54, 2020.

[KFB20] D. Kaufmann, M. Fleury and A. Biere. Pacheck and Pastèque, Checking Practical Algebraic Calculus Proofs. In Proc. of FMCAD'20, pages 264–269, TU Vienna Academic Press, 2020.

Referenzen II

[MGD19] A. Mahzoon, D. Große and R. Drechsler. RevSCA: Using Reverse Engineering to Bring Light into Backwards Rewriting for Big and Dirty Multiplizierers. In Proc. of DAC'19, pages 185:1–185:6, ACM, 2019.

[RBK17] D. Ritirc, A. Biere and M. Kauers. Column-Wise Verification of Multiplizierers Using Computer Algebra. In Proc. of FMCAD'17, pages 23–30, IEEE, 2017.

[RBK18] D. Ritirc, A. Biere and M. Kauers. A Practical Polynomial Calculus for Arithmetic Circuit Verification. In Proc. of SC'2, pages 61–76, CEUR-WS, 2018.

[Sa16] A. Sayed-Ahmed, D. Große, U. Kühne, M. Soeken, and R. Drechsler. Formal verification of integer Multiplizierers by combining Gröbner basis with logic reduction. In Proc. of DATE'16, pages 1048–1053, IEEE, 2016.

[Yu16] C. Yu, W. Brown, D. Liu, A. Rossi and M. Ciesielski. Formal Verification of Arithmetic Circuity by Function Extraction. In IEEE TCAD, vol 35, pages 12:2131–12:2142, 2016.