MAKING MANY MORE MATRIX MULTIPLICATION METHODS

Manuel Kauers · Institute for Algebra · JKU

Joint work with Marijn Heule (Texas) and Martina Seidl (Linz)

$$\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \begin{pmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{pmatrix} = \begin{pmatrix} c_{1,1} & c_{1,2} \\ c_{2,1} & c_{2,2} \end{pmatrix}$$

$$c_{1,1} = a_{1,1} \cdot b_{1,1} + a_{1,2} \cdot b_{2,1}$$
$$c_{1,2} = a_{1,1} \cdot b_{1,2} + a_{1,2} \cdot b_{2,2}$$
$$c_{2,1} = a_{2,1} \cdot b_{1,1} + a_{2,2} \cdot b_{2,1}$$
$$c_{2,2} = a_{2,1} \cdot b_{1,2} + a_{2,2} \cdot b_{2,2}$$

$$\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \begin{pmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{pmatrix} = \begin{pmatrix} c_{1,1} & c_{1,2} \\ c_{2,1} & c_{2,2} \end{pmatrix}$$

$$\begin{aligned} c_{1,1} &= M_1 + M_4 - M_5 + M_7 \\ c_{1,2} &= M_3 + M_5 \\ c_{2,1} &= M_2 + M_4 \\ c_{2,2} &= M_1 - M_2 + M_3 + M_6 \end{aligned}$$

$$\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \begin{pmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{pmatrix} = \begin{pmatrix} c_{1,1} & c_{1,2} \\ c_{2,1} & c_{2,2} \end{pmatrix}$$

... where

$$M_1 = (a_{1,1} + a_{2,2}) \cdot (b_{1,1} + b_{2,2})$$
$$M_2 = (a_{2,1} + a_{2,2}) \cdot b_{1,1}$$
$$M_3 = a_{1,1} \cdot (b_{1,2} - b_{2,2})$$
$$M_4 = a_{2,2} \cdot (b_{2,1} - b_{1,1})$$
$$M_5 = (a_{1,1} + a_{1,2}) \cdot b_{2,2}$$
$$M_6 = (a_{2,1} - a_{1,1}) \cdot (b_{1,1} + b_{1,2})$$
$$M_7 = (a_{1,2} - a_{2,2}) \cdot (b_{2,1} + b_{2,2})$$

$$\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \begin{pmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{pmatrix} = \begin{pmatrix} c_{1,1} & c_{1,2} \\ c_{2,1} & c_{2,2} \end{pmatrix}$$

- This scheme needs 7 multiplications instead of 8.

$$\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \begin{pmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{pmatrix} = \begin{pmatrix} c_{1,1} & c_{1,2} \\ c_{2,1} & c_{2,2} \end{pmatrix}$$

- This scheme needs 7 multiplications instead of 8.
- Recursive application allows to multiply $n \times n$ matrices with $O(n^{\log_2 7})$ operations in the ground ring.

$$\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \begin{pmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{pmatrix} = \begin{pmatrix} c_{1,1} & c_{1,2} \\ c_{2,1} & c_{2,2} \end{pmatrix}$$

- This scheme needs 7 multiplications instead of 8.
- Recursive application allows to multiply $n \times n$ matrices with $O(n^{\log_2 7})$ operations in the ground ring.
- Let ω be the smallest number so that $n \times n$ matrices can be multiplied using $O(n^\omega)$ operations in the ground domain.

$$\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \begin{pmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{pmatrix} = \begin{pmatrix} c_{1,1} & c_{1,2} \\ c_{2,1} & c_{2,2} \end{pmatrix}$$

- This scheme needs 7 multiplications instead of 8.
- Recursive application allows to multiply $n \times n$ matrices with $O(n^{\log_2 7})$ operations in the ground ring.
- Let ω be the smallest number so that $n \times n$ matrices can be multiplied using $O(n^{\omega})$ operations in the ground domain.
- Then $2 \leq \omega < 3$. What is the exact value?

- Strassen 1969: $\omega \leq \log_2 7 \leq 2.807$

- Strassen 1969: $\omega \leq \log_2 7 \leq 2.807$
- Pan 1978: $\omega \leq 2.796$
- Bini et al. 1979: $\omega \leq 2.7799$
- Schönhage 1981: $\omega \leq 2.522$
- Romani 1982: $\omega \leq 2.517$
- Coppersmith/Winograd 1981: $\omega \leq 2.496$
- Strassen 1986: $\omega \leq 2.479$
- Coppersmith/Winograd 1990: $\omega \leq 2.376$

- Strassen 1969: $\omega \leq \log_2 7 \leq 2.807$
- Pan 1978: $\omega \leq 2.796$
- Bini et al. 1979: $\omega \leq 2.7799$
- Schönhage 1981: $\omega \leq 2.522$
- Romani 1982: $\omega \leq 2.517$
- Coppersmith/Winograd 1981: $\omega \leq 2.496$
- Strassen 1986: $\omega \leq 2.479$
- Coppersmith/Winograd 1990: $\omega \leq 2.376$
- Stothers 2010: $\omega \leq 2.374$
- Williams 2011: $\omega \leq 2.3728642$
- Le Gall 2014: $\omega \leq 2.3728639$

- Only Strassen's algorithm beats the classical algorithm for reasonable problem sizes.

- Only Strassen's algorithm beats the classical algorithm for reasonable problem sizes.
- Want: a matrix multiplication algorithm that beats Strassen's algorithm for matrices of moderate size.

- Only Strassen's algorithm beats the classical algorithm for reasonable problem sizes.
- Want: a matrix multiplication algorithm that beats Strassen's algorithm for matrices of moderate size.
- Idea: instead of dividing the matrices into 2×2-block matrices, divide them into 3×3-block matrices.

- Only Strassen's algorithm beats the classical algorithm for reasonable problem sizes.
- Want: a matrix multiplication algorithm that beats Strassen's algorithm for matrices of moderate size.
- Idea: instead of dividing the matrices into 2×2-block matrices, divide them into 3×3-block matrices.
- Question: What's the minimal number of multiplications needed to multiply two 3×3 matrices?

- Only Strassen's algorithm beats the classical algorithm for reasonable problem sizes.
- Want: a matrix multiplication algorithm that beats Strassen's algorithm for matrices of moderate size.
- Idea: instead of dividing the matrices into 2×2-block matrices, divide them into 3×3-block matrices.
- Question: What's the minimal number of multiplications needed to multiply two 3×3 matrices?
- Answer: Nobody knows.

Question: What's the minimal number of multiplications needed to multiply two 3×3 matrices?

Question: What's the minimal number of multiplications needed to multiply two 3×3 matrices?

- naive algorithm: 27

Question: What's the minimal number of multiplications needed to multiply two 3×3 matrices?

- naive algorithm: 27
- padd with zeros, use Strassen twice, cleanup: 25

Question: What's the minimal number of multiplications needed to multiply two 3×3 matrices?

- naive algorithm: 27
- padd with zeros, use Strassen twice, cleanup: 25
- best known upper bound: 23 (Laderman 1976)

Question: What's the minimal number of multiplications needed to multiply two 3×3 matrices?

- naive algorithm: 27
- padd with zeros, use Strassen twice, cleanup: 25
- best known upper bound: 23 (Laderman 1976)
- best known lower bound: 19 (Bläser 2003)

Question: What's the minimal number of multiplications needed to multiply two 3×3 matrices?

- naive algorithm: 27
- padd with zeros, use Strassen twice, cleanup: 25
- best known upper bound: 23 (Laderman 1976)
- best known lower bound: 19 (Bläser 2003)
- maximal number of multiplications allowed if we want to beat Strassen: 21 (because $\log_3 21 < \log_2 7 < \log_3 22$).

Laderman's scheme from 1976:

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix} \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix} = \begin{pmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,1} & c_{3,2} & c_{3,3} \end{pmatrix}$$

$$\begin{aligned}
c_{1,1} &= -M_6 + M_{14} + M_{19} \\
c_{2,1} &= M_2 + M_3 + M_4 + M_6 + M_{14} + M_{16} + M_{17} \\
c_{3,1} &= M_6 + M_7 - M_8 + M_{11} + M_{12} + M_{13} - M_{14} \\
c_{1,2} &= M_1 - M_4 + M_5 - M_6 - M_{12} + M_{14} + M_{15} \\
c_{2,2} &= M_2 + M_4 - M_5 + M_6 + M_{20} \\
c_{3,2} &= M_{12} + M_{13} - M_{14} - M_{15} + M_{22} \\
c_{1,3} &= -M_6 - M_7 + M_9 + M_{10} + M_{14} + M_{16} + M_{18} \\
c_{2,3} &= M_{14} + M_{16} + M_{17} + M_{18} + M_{21} \\
c_{3,3} &= M_6 + M_7 - M_8 - M_9 + M_{23}
\end{aligned}$$

Laderman's scheme from 1976:

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix} \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix} = \begin{pmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,1} & c_{3,2} & c_{3,3} \end{pmatrix}$$

where ...

$$\begin{aligned}
M_1 &= (-a_{1,1} + a_{1,2} + a_{1,3} - a_{2,1} + a_{2,2} + a_{3,2} + a_{3,3}) \cdot b_{2,2} \\
M_2 &= (a_{1,1} + a_{2,1}) \cdot (b_{1,2} + b_{2,2}) \\
M_3 &= a_{2,2} \cdot (b_{1,1} - b_{1,2} + b_{2,1} - b_{2,2} - b_{2,3} + b_{3,1} - b_{3,3}) \\
M_4 &= (-a_{1,1} - a_{2,1} + a_{2,2}) \cdot (-b_{1,1} + b_{1,2} + b_{2,2}) \\
M_5 &= (-a_{2,1} + a_{2,2}) \cdot (-b_{1,1} + b_{1,2}) \\
M_6 &= -a_{1,1} \cdot b_{1,1} \\
M_7 &= (a_{1,1} + a_{3,1} + a_{3,2}) \cdot (b_{1,1} - b_{1,3} + b_{2,3}) \\
M_8 &= (a_{1,1} + a_{3,1}) \cdot (-b_{1,3} + b_{2,3}) \\
M_9 &= (a_{3,1} + a_{3,2}) \cdot (b_{1,1} - b_{1,3})
\end{aligned}$$

Laderman's scheme from 1976:

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix} \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix} = \begin{pmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,1} & c_{3,2} & c_{3,3} \end{pmatrix}$$

where ...

$$M_{10} = (a_{1,1} + a_{1,2} - a_{1,3} - a_{2,2} + a_{2,3} + a_{3,1} + a_{3,2}) \cdot b_{2,3}$$
$$M_{11} = (a_{3,2}) \cdot (-b_{1,1} + b_{1,3} + b_{2,1} - b_{2,2} - b_{2,3} - b_{3,1} + b_{3,2})$$
$$M_{12} = (a_{1,3} + a_{3,2} + a_{3,3}) \cdot (b_{2,2} + b_{3,1} - b_{3,2})$$
$$M_{13} = (a_{1,3} + a_{3,3}) \cdot (-b_{2,2} + b_{3,2})$$
$$M_{14} = a_{1,3} \cdot b_{3,1}$$
$$M_{15} = (-a_{3,2} - a_{3,3}) \cdot (-b_{3,1} + b_{3,2})$$
$$M_{16} = (a_{1,3} + a_{2,2} - a_{2,3}) \cdot (b_{2,3} - b_{3,1} + b_{3,3})$$
$$M_{17} = (-a_{1,3} + a_{2,3}) \cdot (b_{2,3} + b_{3,3})$$
$$M_{18} = (a_{2,2} - a_{2,3}) \cdot (b_{3,1} - b_{3,3})$$

Laderman's scheme from 1976:

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix} \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix} = \begin{pmatrix} c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,1} & c_{3,2} & c_{3,3} \end{pmatrix}$$

where ...

$$M_{19} = a_{1,2} \cdot b_{2,1}$$
$$M_{20} = a_{2,3} \cdot b_{3,2}$$
$$M_{21} = a_{2,1} \cdot b_{1,3}$$
$$M_{22} = a_{3,1} \cdot b_{1,2}$$
$$M_{23} = a_{3,3} \cdot b_{3,3}$$

- While Strassen's scheme is essentially the only way to do the 2×2 case with 7 multiplications, there are several distinct schemes for 3×3 matrices using 23 multiplications.

- While Strassen's scheme is essentially the only way to do the 2×2 case with 7 multiplications, there are several distinct schemes for 3×3 matrices using 23 multiplications.
- In fact, there are infinitely many such schemes due to Johnson and McLoughlin, but they involve fractional coefficients and therefore do not work for every coefficient ring.

- While Strassen's scheme is essentially the only way to do the 2×2 case with 7 multiplications, there are several distinct schemes for 3×3 matrices using 23 multiplications.
- In fact, there are infinitely many such schemes due to Johnson and McLoughlin, but they involve fractional coefficients and therefore do not work for every coefficient ring.
- If we insist in integer coefficients, there have so far (and to our knowledge) been only three other schemes for 3×3 matrices and 23 multiplications.

- While Strassen's scheme is essentially the only way to do the 2×2 case with 7 multiplications, there are several distinct schemes for 3×3 matrices using 23 multiplications.
- In fact, there are infinitely many such schemes due to Johnson and McLoughlin, but they involve fractional coefficients and therefore do not work for every coefficient ring.
- If we insist in integer coefficients, there have so far (and to our knowledge) been only three other schemes for 3×3 matrices and 23 multiplications.
- Using altogether about 35 years of computation time, we found more than 13000 new schemes for 3×3 and 23, and we expect that there are many others.

- While Strassen's scheme is essentially the only way to do the 2×2 case with 7 multiplications, there are several distinct schemes for 3×3 matrices using 23 multiplications.
- In fact, there are infinitely many such schemes due to Johnson and McLoughlin, but they involve fractional coefficients and therefore do not work for every coefficient ring.
- If we insist in integer coefficients, there have so far (and to our knowledge) been only three other schemes for 3×3 matrices and 23 multiplications.
- Using altogether about 35 years of computation time, we found more than 13000 new schemes for 3×3 and 23, and we expect that there are many others.
- Unfortunately we found no scheme with only 22 multiplications

How to search for a matrix multiplication scheme?

How to search for a matrix multiplication scheme?

Make an ansatz

$$
\begin{aligned}
M_1 &= (\alpha_{1,1}^{(1)} a_{1,1} + \alpha_{1,2}^{(1)} a_{1,2} + \cdots)(\beta_{1,1}^{(1)} b_{1,1} + \cdots) \\
M_2 &= (\alpha_{1,1}^{(2)} a_{1,1} + \alpha_{1,2}^{(2)} a_{1,2} + \cdots)(\beta_{1,1}^{(2)} b_{1,1} + \cdots) \\
&\vdots \\
c_{1,1} &= \gamma_{1,1}^{(1)} M_1 + \gamma_{1,1}^{(2)} M_2 + \cdots \\
&\vdots
\end{aligned}
$$

How to search for a matrix multiplication scheme?

Make an ansatz

$$
\begin{aligned}
M_1 &= (\alpha_{1,1}^{(1)} a_{1,1} + \alpha_{1,2}^{(1)} a_{1,2} + \cdots)(\beta_{1,1}^{(1)} b_{1,1} + \cdots) \\
M_2 &= (\alpha_{1,1}^{(2)} a_{1,1} + \alpha_{1,2}^{(2)} a_{1,2} + \cdots)(\beta_{1,1}^{(2)} b_{1,1} + \cdots) \\
&\vdots \\
c_{1,1} &= \gamma_{1,1}^{(1)} M_1 + \gamma_{1,1}^{(2)} M_2 + \cdots \\
&\vdots
\end{aligned}
$$

Set $c_{i,j} = \sum_k a_{i,k} b_{k,j}$ for all i, j and compare coefficients.

How to search for a matrix multiplication scheme?

This gives the Brent equations (e.g., for 3×3 with 23 multiplications)

$$\forall\ i, j, k, l, m, n \in \{1, 2, 3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k} \delta_{i,m} \delta_{l,n}$$

How to search for a matrix multiplication scheme?

This gives the Brent equations (e.g., for 3×3 with 23 multiplications)

$$\forall\ i,j,k,l,m,n \in \{1,2,3\}: \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k}\delta_{i,m}\delta_{l,n}$$

- $3^6 = 729$ cubic equations
- $23 \cdot 9 \cdot 3 = 621$ variables

How to search for a matrix multiplication scheme?

This gives the Brent equations (e.g., for 3×3 with 23 multiplications)

$$\forall\ i, j, k, l, m, n \in \{1, 2, 3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k} \delta_{i,m} \delta_{l,n}$$

- $3^6 = 729$ cubic equations
- $23 \cdot 9 \cdot 3 = 621$ variables

Laderman claims that he solved this system by hand,
but he doesn't say exactly how.

How to search for a matrix multiplication scheme?

This gives the Brent equations (e.g., for 3×3 with 23 multiplications)

$$\forall\ i, j, k, l, m, n \in \{1, 2, 3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k} \delta_{i,m} \delta_{l,n}$$

Solve this system in $\mathbb{Z}_2$.

How to search for a matrix multiplication scheme?

This gives the Brent equations (e.g., for 3×3 with 23 multiplications)

$$\forall\ i,j,k,l,m,n \in \{1,2,3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k}\delta_{i,m}\delta_{l,n}$$

Solve this system in $\mathbb{Z}_2$.

Reading $\alpha_{i,j}^{(q)}$, $\beta_{k,l}^{(q)}$, $\gamma_{m,n}^{(q)}$ as boolean variables and $+$ as XOR, the problem becomes a SAT problem.

Problem: SAT solvers don't like XOR. They want CNF as input.

Problem: SAT solvers don't like XOR. They want CNF as input.

$$a + b = 1 \iff (\bar{a} \vee \bar{b}) \wedge (a \vee b)$$

Problem: SAT solvers don't like XOR. They want CNF as input.

$$a + b = 1 \iff (\bar{a} \vee \bar{b}) \wedge (a \vee b)$$

$$a + b + c = 1 \iff (\bar{a} \vee \bar{b} \vee c) \wedge (\bar{a} \vee \bar{c} \vee b) \wedge (\bar{b} \vee \bar{c} \vee a) \wedge (a \vee b \vee c)$$

Problem: SAT solvers don't like XOR. They want CNF as input.

$$
\begin{aligned}
a + b = 1 &\iff (\bar{a} \vee \bar{b}) \wedge (a \vee b) \\
a + b + c = 1 &\iff (\bar{a} \vee \bar{b} \vee c) \wedge (\bar{a} \vee \bar{c} \vee b) \\
&\quad \wedge (\bar{b} \vee \bar{c} \vee a) \wedge (a \vee b \vee c) \\
a + b + c + d = 1 &\iff (\bar{a} \vee \bar{b} \vee \bar{c} \vee \bar{d}) \wedge (\bar{a} \vee \bar{b} \vee c \vee d) \\
&\quad \wedge (\bar{a} \vee \bar{c} \vee b \vee d) \wedge (\bar{a} \vee \bar{d} \vee b \vee c) \\
&\quad \wedge (\bar{b} \vee \bar{c} \vee a \vee d) \wedge (\bar{b} \vee \bar{d} \vee a \vee c) \\
&\quad \wedge (\bar{c} \vee \bar{d} \vee a \vee b) \wedge (a \vee b \vee c \vee d).
\end{aligned}
$$

Problem: SAT solvers don't like XOR. They want CNF as input.

$$a + b = 1 \iff (\bar{a} \vee \bar{b}) \wedge (a \vee b)$$

$$\begin{aligned} a + b + c = 1 \iff & (\bar{a} \vee \bar{b} \vee c) \wedge (\bar{a} \vee \bar{c} \vee b) \\ & \wedge (\bar{b} \vee \bar{c} \vee a) \wedge (a \vee b \vee c) \end{aligned}$$

$$\begin{aligned} a + b + c + d = 1 \iff & (\bar{a} \vee \bar{b} \vee \bar{c} \vee \bar{d}) \wedge (\bar{a} \vee \bar{b} \vee c \vee d) \\ & \wedge (\bar{a} \vee \bar{c} \vee b \vee d) \wedge (\bar{a} \vee \bar{d} \vee b \vee c) \\ & \wedge (\bar{b} \vee \bar{c} \vee a \vee d) \wedge (\bar{b} \vee \bar{d} \vee a \vee c) \\ & \wedge (\bar{c} \vee \bar{d} \vee a \vee b) \wedge (a \vee b \vee c \vee d). \end{aligned}$$

Expanding a 23-term sum into CNF like this gives a million clauses.

SAT people avoid this explosion by assigning new variables ("Tseitin variables") to subexpressions before converting to CNF:

SAT people avoid this explosion by assigning new variables ("Tseitin variables") to subexpressions before converting to CNF:

$$a + b + c + d + e + f + g + h + i = 0$$

SAT people avoid this explosion by assigning new variables ("Tseitin variables") to subexpressions before converting to CNF:

$$a + b + c + d + e + f + g + h + i = 0$$

$$\downarrow$$

$$a + b + c = T_1$$

$$d + e + f = T_2$$

$$g + h + i = T_3$$

$$T_1 + T_2 + T_3 = 0$$

SAT people avoid this explosion by assigning new variables ("Tseitin variables") to subexpressions before converting to CNF:

$$a + b + c + d + e + f + g + h + i = 0$$

$$\downarrow$$

$$a + b + c = T_1 \quad \rightarrow \text{CNF}$$

$$d + e + f = T_2 \quad \rightarrow \text{CNF}$$

$$g + h + i = T_3 \quad \rightarrow \text{CNF}$$

$$T_1 + T_2 + T_3 = 0 \quad \rightarrow \text{CNF}$$

SAT people avoid this explosion by assigning new variables ("Tseitin variables") to subexpressions before converting to CNF:

$$a + b + c + d + e + f + g + h + i = 0$$

$$\downarrow$$

$$a + b + c = T_1 \quad \rightarrow \text{CNF}$$

$$d + e + f = T_2 \quad \rightarrow \text{CNF}$$

$$g + h + i = T_3 \quad \rightarrow \text{CNF}$$

$$T_1 + T_2 + T_3 = 0 \quad \rightarrow \text{CNF}$$

This decreases the number (and length) of clauses at the cost of increasing the number of variables.

- Even these simplified SAT instances are very difficult to solve.

- Even these simplified SAT instances are very difficult to solve.
- State of the art solvers are not able to solve them.

- Even these simplified SAT instances are very difficult to solve.
- State of the art solvers are not able to solve them.
- We help them by making the problem a bit harder, e.g., by

- Even these simplified SAT instances are very difficult to solve.
- State of the art solvers are not able to solve them.
- We help them by making the problem a bit harder, e.g., by
    - replacing the XOR-conditions $\sum_q x_q = 0$ by "zero or two of the x_q shall be true",

- Even these simplified SAT instances are very difficult to solve.
- State of the art solvers are not able to solve them.
- We help them by making the problem a bit harder, e.g., by
  - replacing the XOR-conditions $\sum_q x_q = 0$ by "zero or two of the x_q shall be true",
  - instantiating some of the variables $\alpha_{i,j}^{(q)}$, $\beta_{i,j}^{(q)}$, $\gamma_{i,j}^{(q)}$ by the values they have in known schemes,

- Even these simplified SAT instances are very difficult to solve.
- State of the art solvers are not able to solve them.
- We help them by making the problem a bit harder, e.g., by
  - replacing the XOR-conditions $\sum_q x_q = 0$ by "zero or two of the x_q shall be true",
  - instantiating some of the variables $\alpha_{i,j}^{(q)}$, $\beta_{i,j}^{(q)}$, $\gamma_{i,j}^{(q)}$ by the values they have in known schemes,
  - forcing some of the products $\alpha_{i,j}^{(q)}\beta_{k,l}^{(q)}\gamma_{m,n}^{(q)}$ to zero or one, in accordance with an educated guess.

- Even these simplified SAT instances are very difficult to solve.
- State of the art solvers are not able to solve them.
- We help them by making the problem a bit harder, e.g., by
  - replacing the XOR-conditions $\sum_q x_q = 0$ by "zero or two of the x_q shall be true",
  - instantiating some of the variables $\alpha_{i,j}^{(q)}$, $\beta_{i,j}^{(q)}$, $\gamma_{i,j}^{(q)}$ by the values they have in known schemes,
  - forcing some of the products $\alpha_{i,j}^{(q)}\beta_{k,l}^{(q)}\gamma_{m,n}^{(q)}$ to zero or one, in accordance with an educated guess.
- We use the SAT solver yalsat, which uses a different paradigm than the common state of the art but happens to perform better on our problems than the state of the art solvers.

- Even these simplified SAT instances are very difficult to solve.
- State of the art solvers are not able to solve them.
- We help them by making the problem a bit harder, e.g., by
  - replacing the XOR-conditions $\sum_q x_q = 0$ by "zero or two of the x_q shall be true",
  - instantiating some of the variables $\alpha_{i,j}^{(q)}$, $\beta_{i,j}^{(q)}$, $\gamma_{i,j}^{(q)}$ by the values they have in known schemes,
  - forcing some of the products $\alpha_{i,j}^{(q)}\beta_{k,l}^{(q)}\gamma_{m,n}^{(q)}$ to zero or one, in accordance with an educated guess.
- We use the SAT solver yalsat, which uses a different paradigm than the common state of the art but happens to perform better on our problems than the state of the art solvers.

Each index of a variable in a term

$$\alpha_{i,j}^{(q)}\beta_{k,l}^{(q)}\gamma_{m,n}^{(q)}$$

has a natural index mate in another variable.

Each index of a variable in a term

$$\alpha_{i,j}^{(q)}\beta_{k,l}^{(q)}\gamma_{m,n}^{(q)}$$

has a natural index mate in another variable.

Each index of a variable in a term

$$\alpha_{i,j}^{(q)}\beta_{k,l}^{(q)}\gamma_{m,n}^{(q)}$$

has a natural index mate in another variable.

Each index of a variable in a term

$$\alpha^{(q)}_{i,j}\beta^{(q)}_{k,l}\gamma^{(q)}_{m,n}$$

has a natural index mate in another variable.

Each index of a variable in a term

$$\alpha_{i,j}^{(q)}\beta_{k,l}^{(q)}\gamma_{m,n}^{(q)}$$

has a natural index mate in another variable.

Not all combinations are equally likely to appear in a solution.

Each index of a variable in a term

$$\alpha_{i,j}^{(q)}\beta_{k,l}^{(q)}\gamma_{m,n}^{(q)}$$

has a natural index mate in another variable.

Not all combinations are equally likely to appear in a solution.

- Almost all terms with three mismatches (i.e., $i \neq m$ and $j \neq k$ and $l \neq n$) are zero. We randomly select half of them and set them to zero.

Each index of a variable in a term

$$\alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)}$$

has a natural index mate in another variable.

Not all combinations are equally likely to appear in a solution.

- Almost all terms with three mismatches (i.e., $i \neq m$ and $j \neq k$ and $l \neq n$) are zero. We randomly select half of them and set them to zero.
- Every term with no mismatch (i.e., $i = m$ and $j = k$ and $l = n$) must be one for at least one q. Typically, each such term appears for exactly one q.

Each index of a variable in a term

$$\alpha_{i,j}^{(q)}\beta_{k,l}^{(q)}\gamma_{m,n}^{(q)}$$

has a natural index mate in another variable.

Not all combinations are equally likely to appear in a solution.

- Almost all terms with three mismatches (i.e., $i \neq m$ and $j \neq k$ and $l \neq n$) are zero. We randomly select half of them and set them to zero.
- Every term with no mismatch (i.e., $i = m$ and $j = k$ and $l = n$) must be one for at least one q. Typically, each such term appears for exactly one q.
- Since there are 27 such terms and 23 q's, there must be 19 q's with one term and 4 q's with two terms. We randomly enforce such an assignment.

- For many random choices of such additional constraints, we gave the SAT solver a few minutes to find a solution.

- For many random choices of such additional constraints, we gave the SAT solver a few minutes to find a solution.
- Usually it did not find any, but there were also many cases in which a solution was found.

- For many random choices of such additional constraints, we gave the SAT solver a few minutes to find a solution.
- Usually it did not find any, but there were also many cases in which a solution was found.
- Are all these solutions really new? What does it mean for a solution to be new?

Matrix multiplication $AB = C$ enjoys several symmetries:

Matrix multiplication $AB = C$ enjoys several symmetries:

- $AUU^{-1}B = C$ for every invertible U

Matrix multiplication $AB = C$ enjoys several symmetries:

- $AUU^{-1}B = C$ for every invertible U
- $VAB = VC$ for every invertible V

Matrix multiplication $AB = C$ enjoys several symmetries:

- $AUU^{-1}B = C$ for every invertible U
- $VAB = VC$ for every invertible V
- $ABW = CW$ for every invertible W

Matrix multiplication $AB = C$ enjoys several symmetries:

- $AUU^{-1}B = C$ for every invertible U
- $VAB = VC$ for every invertible V
- $ABW = CW$ for every invertible W
- $B^\top A^\top = C^\top$

Matrix multiplication $AB = C$ enjoys several symmetries:

- $AUU^{-1}B = C$ for every invertible U
- $VAB = VC$ for every invertible V
- $ABW = CW$ for every invertible W
- $B^\top A^\top = C^\top$
- and one more that is a little more subtle

Matrix multiplication $AB = C$ enjoys several symmetries:

- $AUU^{-1}B = C$ for every invertible U
- $VAB = VC$ for every invertible V
- $ABW = CW$ for every invertible W
- $B^\top A^\top = C^\top$
- and one more that is a little more subtle

The symmetry group turns out to be $S_3 \times \mathrm{GL}(n)^3$.

Matrix multiplication $AB = C$ enjoys several symmetries:

- $AUU^{-1}B = C$ for every invertible U
- $VAB = VC$ for every invertible V
- $ABW = CW$ for every invertible W
- $B^\top A^\top = C^\top$
- and one more that is a little more subtle

Taking also into account that we can reorder the sums in the Brent equations, the symmetry group is in fact $S_{23} \times S_3 \times \mathrm{GL}(n)^3$.

Matrix multiplication $AB = C$ enjoys several symmetries:

- $AUU^{-1}B = C$ for every invertible U
- $VAB = VC$ for every invertible V
- $ABW = CW$ for every invertible W
- $B^\top A^\top = C^\top$
- and one more that is a little more subtle

Taking also into account that we can reorder the sums in the Brent equations, the symmetry group is in fact $S_{23} \times S_3 \times \mathrm{GL}(n)^3$.

For $\mathbb{Z}_2$, this group has almost 10^{30} elements. For comparison: the whole search space has size $2^{621} \approx 10^{187}$.

Matrix multiplication $AB = C$ enjoys several symmetries:

- $AUU^{-1}B = C$ for every invertible U
- $VAB = VC$ for every invertible V
- $ABW = CW$ for every invertible W
- $B^\top A^\top = C^\top$
- and one more that is a little more subtle

Taking also into account that we can reorder the sums in the Brent equations, the symmetry group is in fact $S_{23} \times S_3 \times \mathrm{GL}(n)^3$.

Group elements can map solutions into other solutions. A solution is new when it cannot be mapped to one we have seen before.

Matrix multiplication $AB = C$ enjoys several symmetries:

- $AUU^{-1}B = C$ for every invertible U
- $VAB = VC$ for every invertible V
- $ABW = CW$ for every invertible W
- $B^\top A^\top = C^\top$
- and one more that is a little more subtle

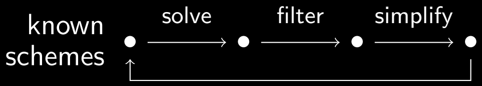
Taking also into account that we can reorder the sums in the Brent equations, the symmetry group is in fact $S_{23} \times S_3 \times \mathrm{GL}(n)^3$.

Group elements can map solutions into other solutions. A solution is new when it cannot be mapped to one we have seen before.

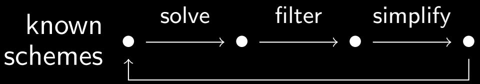
The 13000 new schemes announced earlier are new in this sense.

Symmetries can also be used to simplify the solutions, which was useful for producing new solutions from known ones.

Symmetries can also be used to simplify the solutions, which was useful for producing new solutions from known ones.

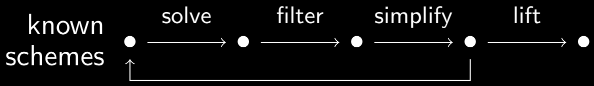


Symmetries can also be used to simplify the solutions, which was useful for producing new solutions from known ones.



There are two post processing steps:

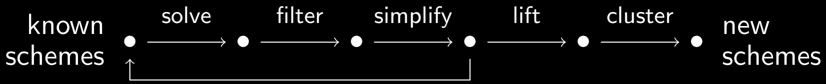
Symmetries can also be used to simplify the solutions, which was useful for producing new solutions from known ones.



There are two post processing steps:

- lifting: introduce signs so that the schemes work not only for $\mathbb{Z}_2$ but also for $\mathbb{Z}$ (and thus for any coefficient ring)

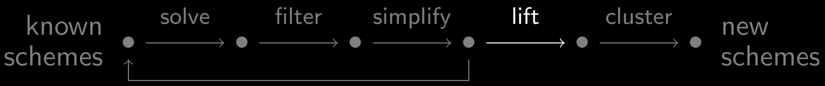
Symmetries can also be used to simplify the solutions, which was useful for producing new solutions from known ones.



There are two post processing steps:

- lifting: introduce signs so that the schemes work not only for $\mathbb{Z}_2$ but also for $\mathbb{Z}$ (and thus for any coefficient ring)
- clustering: extract parameterized families from the schemes.

Symmetries can also be used to simplify the solutions, which was useful for producing new solutions from known ones.



There are two post processing steps:

- lifting: introduce signs so that the schemes work not only for $\mathbb{Z}_2$ but also for $\mathbb{Z}$ (and thus for any coefficient ring)
- clustering: extract parameterized families from the schemes.

Lifting.

Remember the Brent equations:

$$\forall\ i, j, k, l, m, n \in \{1, 2, 3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k} \delta_{i,m} \delta_{l,n}$$

Lifting.

Remember the Brent equations:

$$\forall\ i,j,k,l,m,n \in \{1,2,3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k}\delta_{i,m}\delta_{l,n}$$

Suppose we know a solution in $\mathbb{Z}_2$.

Lifting.

Remember the Brent equations:

$$\forall\ i, j, k, l, m, n \in \{1, 2, 3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k}\delta_{i,m}\delta_{l,n}$$

Suppose we know a solution in $\mathbb{Z}_2$.

Assume it came from a solution in $\mathbb{Z}$ with coefficients in $\{-1, 0, +1\}$.

Lifting.

Remember the Brent equations:

$$\forall\ i, j, k, l, m, n \in \{1, 2, 3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k} \delta_{i,m} \delta_{l,n}$$

Suppose we know a solution in $\mathbb{Z}_2$.

Assume it came from a solution in $\mathbb{Z}$ with coefficients in $\{-1, 0, +1\}$.

Then each $0 \in \mathbb{Z}_2$ was $0 \in \mathbb{Z}$ and each $1 \in \mathbb{Z}_2$ was $-1 \in \mathbb{Z}$ or $+1 \in \mathbb{Z}$.

Lifting.

Remember the Brent equations:

$$\forall\ i,j,k,l,m,n \in \{1,2,3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k}\delta_{i,m}\delta_{l,n}$$

Suppose we know a solution in $\mathbb{Z}_2$.

Assume it came from a solution in $\mathbb{Z}$ with coefficients in $\{-1, 0, +1\}$.

Then each $0 \in \mathbb{Z}_2$ was $0 \in \mathbb{Z}$ and each $1 \in \mathbb{Z}_2$ was $-1 \in \mathbb{Z}$ or $+1 \in \mathbb{Z}$.

Plug the 0s of the $\mathbb{Z}_2$-solution into the Brent equations.

Lifting.

Remember the Brent equations:

$$\forall\ i,j,k,l,m,n \in \{1,2,3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k}\delta_{i,m}\delta_{l,n}$$

Suppose we know a solution in $\mathbb{Z}_2$.

Assume it came from a solution in $\mathbb{Z}$ with coefficients in $\{-1, 0, +1\}$.

Then each $0 \in \mathbb{Z}_2$ was $0 \in \mathbb{Z}$ and each $1 \in \mathbb{Z}_2$ was $-1 \in \mathbb{Z}$ or $+1 \in \mathbb{Z}$.

Plug the 0s of the $\mathbb{Z}_2$-solution into the Brent equations.

Find variables that can be set to 1 w.l.o.g. and set them to 1.

Lifting.

Remember the Brent equations:

$$\forall\, i,j,k,l,m,n \in \{1,2,3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k}\delta_{i,m}\delta_{l,n}$$

Suppose we know a solution in $\mathbb{Z}_2$.

Assume it came from a solution in $\mathbb{Z}$ with coefficients in $\{-1, 0, +1\}$.

Then each $0 \in \mathbb{Z}_2$ was $0 \in \mathbb{Z}$ and each $1 \in \mathbb{Z}_2$ was $-1 \in \mathbb{Z}$ or $+1 \in \mathbb{Z}$.

Plug the 0s of the $\mathbb{Z}_2$-solution into the Brent equations.

Find variables that can be set to 1 w.l.o.g. and set them to 1.

For each of the remaining variables, add a new equation $x^2 - 1$.

Lifting.

Remember the Brent equations:

$$\forall\ i, j, k, l, m, n \in \{1, 2, 3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k} \delta_{i,m} \delta_{l,n}$$

Suppose we know a solution in $\mathbb{Z}_2$.

Assume it came from a solution in $\mathbb{Z}$ with coefficients in $\{-1, 0, +1\}$.

Then each $0 \in \mathbb{Z}_2$ was $0 \in \mathbb{Z}$ and each $1 \in \mathbb{Z}_2$ was $-1 \in \mathbb{Z}$ or $+1 \in \mathbb{Z}$.

Plug the 0s of the $\mathbb{Z}_2$-solution into the Brent equations.

Find variables that can be set to 1 w.l.o.g. and set them to 1.

For each of the remaining variables, add a new equation $x^2 - 1$.

Solve the resulting nonlinear system over $\mathbb{Q}$.

Lifting.

Toy example: consider the system

$$
\begin{aligned}
a_1b_1c_1+a_1b_3c_2+a_3b_2c_1+a_3b_1c_3 &= 0 & a_1b_1c_1+a_2b_1c_1+a_3b_2c_1+a_3b_3c_2 &= 0 \\
a_1b_1c_1+a_2b_2c_1+a_3b_2c_1+a_2b_3c_2 &= 0 & a_1b_1c_2+a_2b_1c_1+a_3b_1c_2+a_3b_3c_2 &= 0 \\
a_1b_2c_2+a_2b_1c_2+a_3b_2c_2+a_2b_1c_3 &= 0 & a_1b_3c_2+a_2b_2c_1+a_3b_3c_1+a_3b_1c_3 &= 0 \\
a_2b_1c_1+a_3b_3c_1+a_3b_1c_2+a_3b_1c_3 &= 0 & a_2b_2c_1+a_2b_1c_3+a_2b_2c_3+a_3b_2c_2 &= 0
\end{aligned}
$$

Lifting.

Toy example: consider the system

$$
\begin{aligned}
a_1b_1c_1+a_1b_3c_2+a_3b_2c_1+a_3b_1c_3 &= 0 & \quad a_1b_1c_1+a_2b_1c_1+a_3b_2c_1+a_3b_3c_2 &= 0\\
a_1b_1c_1+a_2b_2c_1+a_3b_2c_1+a_2b_3c_2 &= 0 & \quad a_1b_1c_2+a_2b_1c_1+a_3b_1c_2+a_3b_3c_2 &= 0\\
a_1b_2c_2+a_2b_1c_2+a_3b_2c_2+a_2b_1c_3 &= 0 & \quad a_1b_3c_2+a_2b_2c_1+a_3b_3c_1+a_3b_1c_3 &= 0\\
a_2b_1c_1+a_3b_3c_1+a_3b_1c_2+a_3b_1c_3 &= 0 & \quad a_2b_2c_1+a_2b_1c_3+a_2b_2c_3+a_3b_2c_2 &= 0
\end{aligned}
$$

A solution is

$$(a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3) = (1, 0, 1, 1, 1, 0, 1, 0, 0) \in \mathbb{Z}_2^9.$$

Lifting.

Toy example: consider the system

$$\begin{aligned}
a_1b_1c_1+a_1b_3c_2+a_3b_2c_1+a_3b_1c_3 &= 0 & \qquad a_1b_1c_1+a_2b_1c_1+a_3b_2c_1+a_3b_3c_2 &= 0\\
a_1b_1c_1+a_2b_2c_1+a_3b_2c_1+a_2b_3c_2 &= 0 & \qquad a_1b_1c_2+a_2b_1c_1+a_3b_1c_2+a_3b_3c_2 &= 0\\
a_1b_2c_2+a_2b_1c_2+a_3b_2c_2+a_2b_1c_3 &= 0 & \qquad a_1b_3c_2+a_2b_2c_1+a_3b_3c_1+a_3b_1c_3 &= 0\\
a_2b_1c_1+a_3b_3c_1+a_3b_1c_2+a_3b_1c_3 &= 0 & \qquad a_2b_2c_1+a_2b_1c_3+a_2b_2c_3+a_3b_2c_2 &= 0
\end{aligned}$$

A solution is

$$(a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3) = (1, 0, 1, 1, 1, 0, 1, 0, 0) \in \mathbb{Z}_2^9.$$

Lifting.

Toy example: consider the system

$$
\begin{array}{rrrr}
a_1b_1c_1+ & +a_3b_2c_1+ & & =0 \\
a_1b_1c_1+ & +a_3b_2c_1+ & & =0 \\
+ & + & + & =0 \\
+ & + & + & =0
\end{array}
\qquad
\begin{array}{rrrr}
a_1b_1c_1+ & +a_3b_2c_1+ & & =0 \\
+ & + & + & =0 \\
+ & + & + & =0 \\
+ & + & + & =0
\end{array}
$$

A solution is

$$(a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3) = (1, 0, 1, 1, 1, 0, 1, 0, 0) \in \mathbb{Z}_2^9.$$

Lifting.

Toy example: consider the system

$$
\begin{array}{llll}
a_1 b_1 c_1 + & + a_3 b_2 c_1 + & & = 0 \\
a_1 b_1 c_1 + & + a_3 b_2 c_1 + & & = 0 \\
+ & + & + & = 0 \\
+ & + & + & = 0
\end{array}
\qquad
\begin{array}{llll}
a_1 b_1 c_1 + & + a_3 b_2 c_1 + & & = 0 \\
+ & + & + & = 0 \\
+ & + & + & = 0 \\
+ & + & + & = 0
\end{array}
$$

A solution is

$$(a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3) = (1, 0, 1, 1, 1, 0, 1, 0, 0) \in \mathbb{Z}_2^9.$$

Because of $(-1)xy = 1(-x)y$, we may set $a_1 = c_1 = 1$ w.l.o.g.

Lifting.

Toy example: consider the system

$$\begin{array}{llllllll} a_1 b_1 c_1 + & +a_3 b_2 c_1 + & & = 0 & a_1 b_1 c_1 + & +a_3 b_2 c_1 + & & = 0 \\ a_1 b_1 c_1 + & +a_3 b_2 c_1 + & & = 0 & + & + & + & = 0 \\ + & + & + & = 0 & + & + & + & = 0 \\ + & + & + & = 0 & + & + & + & = 0 \end{array}$$

A solution is

$$(a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3) = (1, 0, 1, 1, 1, 0, 1, 0, 0) \in \mathbb{Z}_2^9.$$

Because of $(-1)xy = 1(-x)y$, we may set $a_1 = c_1 = 1$ w.l.o.g.

Lifting.

Toy example: consider the system

$$\begin{array}{llllll} b_1 & + & +a_3 b_2 & + & & = 0 \\ b_1 & + & +a_3 b_2 & + & & = 0 \\ & + & + & + & & = 0 \\ & + & + & + & & = 0 \end{array} \qquad \begin{array}{llllll} b_1 & + & +a_3 b_2 & + & & = 0 \\ & + & + & + & & = 0 \\ & + & + & + & & = 0 \\ & + & + & + & & = 0 \end{array}$$

A solution is

$$(a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3) = (1, 0, 1, 1, 1, 0, 1, 0, 0) \in \mathbb{Z}_2^9.$$

Because of $(-1)xy = 1(-x)y$, we may set $a_1 = c_1 = 1$ w.l.o.g.

Lifting.

Toy example: consider the system

$$
\begin{array}{ccccc}
b_1 & + & +a_3 b_2 & + & = 0 \\
b_1 & + & +a_3 b_2 & + & = 0 \\
 & + & + & + & = 0 \\
 & + & + & + & = 0
\end{array}
\qquad
\begin{array}{ccccc}
b_1 & + & +a_3 b_2 & + & = 0 \\
 & + & + & + & = 0 \\
 & + & + & + & = 0 \\
 & + & + & + & = 0
\end{array}
$$

A solution is

$$(a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3) = (1, 0, 1, 1, 1, 0, 1, 0, 0) \in \mathbb{Z}_2^9.$$

Because of $(-1)xy = 1(-x)y$, we may set $a_1 = c_1 = 1$ w.l.o.g.

Adding $a_3^2 - 1 = b_1^2 - 1 = b_2^2 - 1 = 0$ and solving gives the solution

$$(a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3) = (1, 0, 1, -1, 1, 0, 1, 0, 0) \in \mathbb{Z}^9.$$

Lifting.

Can every $\mathbb{Z}_2$-solution be lifted to a $\mathbb{Z}$-solution in this way?

Lifting.

Can every $\mathbb{Z}_2$-solution be lifted to a $\mathbb{Z}$-solution in this way?

No, and we found some which don't admit a lifting.

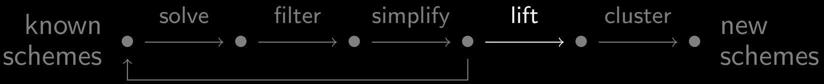
Lifting.

Can every $\mathbb{Z}_2$-solution be lifted to a $\mathbb{Z}$-solution in this way?

No, and we found some which don't admit a lifting.

But they are very rare. In almost all cases, the lifting succeeds.

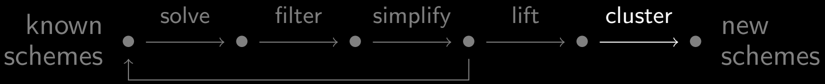
Symmetries can also be used to simplify the solutions, which was useful for producing new solutions from known ones.



There are two post processing steps:

- lifting: introduce signs so that the schemes work not only for $\mathbb{Z}_2$ but also for $\mathbb{Z}$ (and thus for any coefficient ring)
- clustering: extract parameterized families from the schemes.

Symmetries can also be used to simplify the solutions, which was useful for producing new solutions from known ones.



There are two post processing steps:

- lifting: introduce signs so that the schemes work not only for $\mathbb{Z}_2$ but also for $\mathbb{Z}$ (and thus for any coefficient ring)
- clustering: extract parameterized families from the schemes.

Clustering.

Suppose we have a solution to the Brent equations:

$$\forall\ i, j, k, l, m, n \in \{1, 2, 3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k}\delta_{i,m}\delta_{l,n}$$

Clustering.

Suppose we have a solution to the Brent equations:

$$\forall\, i,j,k,l,m,n \in \{1,2,3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k}\delta_{i,m}\delta_{l,n}$$

- If we forget the values of $\alpha_{i,j}^{(q)}$, we can recover them by solving a linear system.

Clustering.

Suppose we have a solution to the Brent equations:

$$\forall\ i, j, k, l, m, n \in \{1, 2, 3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k}\delta_{i,m}\delta_{l,n}$$

- If we forget the values of $\beta_{k,l}^{(q)}$, we can recover them by solving a linear system.

Clustering.

Suppose we have a solution to the Brent equations:

$$\forall\, i, j, k, l, m, n \in \{1, 2, 3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k}\delta_{i,m}\delta_{l,n}$$

- If we forget the values of $\gamma_{m,n}^{(q)}$, we can recover them by solving a linear system.

Clustering.

Suppose we have a solution to the Brent equations:

$$\forall\ i, j, k, l, m, n \in \{1, 2, 3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k}\delta_{i,m}\delta_{l,n}$$

- If we forget the values of $\gamma_{m,n}^{(q)}$, we can recover them by solving a linear system.

Clustering.

Suppose we have a solution to the Brent equations:

$$\forall\, i, j, k, l, m, n \in \{1, 2, 3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k}\delta_{i,m}\delta_{l,n}$$

- If we forget the values of $\gamma_{m,n}^{(q)}$, we can recover them by solving a linear system.
- This computation often gives nontrivial affine spaces of solutions, i.e., more general schemes involving free parameters.

Clustering.

Suppose we have a solution to the Brent equations:

$$\forall\ i,j,k,l,m,n \in \{1,2,3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k}\delta_{i,m}\delta_{l,n}$$

- If we forget the values of $\gamma_{m,n}^{(q)}$, we can recover them by solving a linear system.
- This computation often gives nontrivial affine spaces of solutions, i.e., more general schemes involving free parameters.
- In fact, for every $q \in \{1,\dots,23\}$ we can independently set replace all $\alpha_{i,j}^{(q)}$ or all $\beta_{k,l}^{(q)}$ or all $\gamma_{m,n}^{(q)}$ by unknowns.

Clustering.

Suppose we have a solution to the Brent equations:

$$\forall\, i, j, k, l, m, n \in \{1, 2, 3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k}\delta_{i,m}\delta_{l,n}$$

- If we forget the values of $\gamma_{m,n}^{(q)}$, we can recover them by solving a linear system.
- This computation often gives nontrivial affine spaces of solutions, i.e., more general schemes involving free parameters.
- In fact, for every $q \in \{1, \dots, 23\}$ we can independently set replace all $\alpha_{i,j}^{(q)}$ or all $\beta_{k,l}^{(q)}$ or all $\gamma_{m,n}^{(q)}$ by unknowns.
- Playing the game repeatedly with various choices, we introduce more and more free parameters into the schemes.

Clustering.

Suppose we have a solution to the Brent equations:

$$\forall\ i,j,k,l,m,n \in \{1,2,3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k}\delta_{i,m}\delta_{l,n}$$

- We found several families with up to 17 parameters and with coefficients in $\mathbb{Z}$.

Clustering.

Suppose we have a solution to the Brent equations:

$$\forall\ i,j,k,l,m,n \in \{1,2,3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k}\delta_{i,m}\delta_{l,n}$$

- We found several families with up to 17 parameters and with coefficients in $\mathbb{Z}$.
- Gröbner bases computations can be used to check that these parameters are really independent.

Clustering.

Suppose we have a solution to the Brent equations:

$$\forall\, i,j,k,l,m,n \in \{1,2,3\} : \sum_{q=1}^{23} \alpha_{i,j}^{(q)} \beta_{k,l}^{(q)} \gamma_{m,n}^{(q)} = \delta_{j,k}\delta_{i,m}\delta_{l,n}$$

- We found several families with up to 17 parameters and with coefficients in $\mathbb{Z}$.
- Gröbner bases computations can be used to check that these parameters are really independent.
- For comparison: The schemes of Johnson and McLoughlin had only 3 parameters and coefficients in $\mathbb{Q}$.

So what?

So what?

- Okay, so there are many more matrix multiplication methods for 3×3 matrices with 23 coefficient multiplications than previously known.

So what?

- Okay, so there are many more matrix multiplication methods for 3×3 matrices with 23 coefficient multiplications than previously known.
- In fact, we have shown that the dimension of the algebraic set defined by the Brent equation is much larger than was previously known.

So what?

- Okay, so there are many more matrix multiplication methods for 3×3 matrices with 23 coefficient multiplications than previously known.
- In fact, we have shown that the dimension of the algebraic set defined by the Brent equation is much larger than was previously known.
- But none of this has any immediate implications on the complexity of matrix multiplication, neither theoretically nor practically.

So what?

- Okay, so there are many more matrix multiplication methods for 3×3 matrices with 23 coefficient multiplications than previously known.
- In fact, we have shown that the dimension of the algebraic set defined by the Brent equation is much larger than was previously known.
- But none of this has any immediate implications on the complexity of matrix multiplication, neither theoretically nor practically.
- In particular, it remains open whether there is a multiplication method for 3×3 matrices with 22 coefficient multiplications. If you find one, let us know.

Check out our website for browsing through the schemes and families we found:

http://www.algebra.uni-linz.ac.at/research/matrix-multiplication/